بسم الله الرحمن الرحيم

دورة مطوري الويب

تقديم: أنيس حكمت أبوحميد

الموقع الالكتروني: 2nees.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript

3

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

دورة مجانية

مقدمة

الحمد لله الذي بنعمته تتم الصالحات ، الحمد لله الذي خلق الأرض والسموات ، الحمد لله الذي علم العثرات ، فسترها على اهلها وانزل الرحمات ، ثم غفرها لهم ومحا السيئات ، فله الحمد ملئ خزائن البركات ، وله الحمد ما تتابعت بالقلب النبضات ، وله الحمد ماتعاقبت الخطوات ، وله الحمد عدد حبات الرمال في الفلوات ، وعدد ذرات الهواء في الأرض والسماوات ، وعدد الحركات والسكنات ، سبحانه سبحانه سبحانه سبحانه
الطير سبحه والوحش مجده والموج كبره والحوت ناجاه والنمل تحت الصخور الصم قدسه والنحل يهتف حمداً في خلاياه

سبحان الله وبحمده ..سبحان الله العظيم ... الحمد لله رب العالمين..

الآن لننطلق معا على بركة الله تعالى في الجزء الثالث من الدورة وهو ال JavaScript

قبل أن تبدء

عليك أن تعلم صديقي أن جميع ما سيتم ذكره من أمثلة ومواضيع لن تغطي عالم الويب بشكل كامل، ولن تغطي جميع الخصائص لكل عنصر، وتأكد أن دور أي شرح/دورة/كتاب هو أن يضعك على الطريق وأنت عليك أن تكمل، واجعل من Google هو صديقك الأول ..ولا تبتعد عنه نهائيا، واجعل صديقك الثاني التطبيق العملي لكل أمر تتعلمه، إياك أن تقول سهل، وتتجاوز ..إذا فعلت ذلك، تأكد أنك لن تستطيع كتابة شيفرة برمجية صغيرة ... !!

طبيعة هذه الدورة.. طبيعة برمجية قد تختلف عن سابقاتها باسلوب أو طريقة التفكير ... ^_^.. الأمر سهل أكثر مما تتصور .. لكن مارس ..تعلم .. واكتب.. وابحث.. واسعى للأفضل

أنيس حكمت أبوحميد aneeshikmat@gmail.com

قبل أن تبدء

**) أي عمل يخالف الشريعة الاسلامية بأي وسيلة كانت فأنت مسؤول عن ذلك أمام الله تعالى، وهذه الدورة مجانية لكل من لن يخالف الشريعة الاسلامية في عمله ...

**) متطلبات الدورة:

- HTML
- CSS

تفكر ...

بسم الله الرحمن الرحيم

{فقُلْتُ اسْتغْفِرُوا ربّكُمْ إِنّهُ كان غفّارا (١٠) يُرْسِلِ السّماء عليْكُمْ مِدْرارا (١١) ويُمْدِدْكُمْ بِأمْوالٍ وبنِين ويجْعلْ لكُمْ جنّاتٍ ويجْعلْ لكُمْ أنْهارا (١٢)

سورة نوح

أنيس حكمت أبوحميد aneeshikmat@gmail.com

ما هي ال JavaScript

ال جافا سكربت هي لغة برمجة صممت خصيصا للتعامل مع صفحات الويب، والجافا سكربت هي ليست JAVA .. وتعد الجافا سكربت لغة برمجة سهلة التعلم، ولا يوجد موقع الآن الى ويستخدم الجافا سكربت في صفحاته أو المكتبات المبنية عليها ..

لغة البرمجة هذه تعد إحدى ٣ لغات **يجب** تعلمها لأي مطور ومصمم مواقع انترنت ..وهي أول لغة برمجة فعلية يجب تعلمها باعتبار أن ال html هي markup language ..وال css هي styling sheet...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

ما هي ال JavaScript

والآن لماذا تعد الجافا سكربت، من أكثر اللغات انتشارا؟، لأنها سهلة ويتم استخدامها في جميع الأجهزة تقريبا، ويمكن استخدامها داخل الويب، أو الحواسيب، أو الأجهزة المحمولة والهواتف الذكية ..

والسؤال الآن .. ما هو الشيء الذي يمكننا من خلال تعلمنا للجافا سكربت من فعله ؟

الإجابة بكل بساطة ^_^ ..أمووووووور كثيرة ^_^

أنيس حكمت أبوحميد aneeshikmat@gmail.com

ما هي ال JavaScript

أريدك أولا أن تتعرف الى مصطلح مهم وهو **Document Object Model** ويختصر ب DOM...

ماذا يعني DOM: هذا هو المعيار الرسمي والمعتمد من ال W3C للتعامل أو الوصول الى html ...وهذا يعني التحكم بالعمليات المرتبطة بأي عنصر html من اضافة وحذف وتعديل.. وأي أمر آخر يمكن فعله ^_^.

الجافا سكربت رائعة جدا ..فهي بوصولها الى ال html ..تصنع نوع من التفاعلية بين المستخدمين والصفحات .. ولذلك فالجافا سكربت تقوم بالتعديل والحذف والإضافة بالإضافة الى امكانية التعامل مع ال css وتغيير الخصائص، والتعامل مع السمات ..الكثير من الأمور والتي سنتحدث عنها بإذن الله تعالى ^_^

وقبل البدء JavaScript VS JAVA

يجب دائما قبل البدء بشرح الجافا سكربت ..توضيح أمر مهم جدا ..

JAVASCRIPT NOT JAVA ... أي أن الجافا سكربت ليست جافا، وكل منهما لغة منفصلة عن الأخرى مستقلة بمفاهيمها وبنيتها التركيبية...

ال JAVA هي لغة برمجة كائنية التوجه OOP صممت من قبل James من شركة Sun .

أما ال جافا سكربت: فهو سكربت تم انتاجه من قبل Netscape..

هناك ملاحظة مهمة: وهي مع انهما لغتان منفصلتان، الا أنه قد يوجد هناك بعض التشابه وهناك بعض الاختلاف ..وذلك حسب زاوية النظر الى وظيفة أو طبيعة المهمة المراد انجازها .. لذلك ستجد عبارة في الانترنت ..They are both similar and quite

هذه المعلومات التي تم ذكرها..هي معلومات لك ..لكي تكون ملما بما هو أمامك ..وبما ستعمل عليه بإذن الله تعالى ^_*

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أين يمكنني كتابة الجافا سكربت

أنيس حكمت أبوحميد aneeshikmat@gmail.com

يمكنني كتابة الجافا سكربت ب ٣ أماكن مختلفة ^_^...

الأول داخل ال head tag وذلك عن طريق استخدام ال <script></script>...

الثاني داخل ال body tag وذلك عن طريق استخدام ال <script></script>..

الثالث داخل ملف بامتداد js خارج صفحة ال html يستدعى بالطريقة التالي:

..<script src=”ScriptFileName.js"></script>

والآن لنبدء بشرح كل مكان من هذه الأماكن ^_^*

ملاحظات:

١) كود الجافا سكربت يقع داخل ال <script></script> (بالنسبة للمكان الأول والثاني).

٢) “type="text/javascript كانت تستخدم قديما أو في المتصفحات القديمة لتعريف أن نوع السكربت هو جافا سكربت، لكن الآن لا داعي لذلك

٣) يمكنك كتابة أي عدد من السكربت..^_^، وفي أي مكان كان..داخل الbody or head

٤) يتم وضع السكربت في نهاية ال body لتسريع الصفحة لأنه ينفذ بعد تحميل ال body، ويتم وضع السكربت في ال head لكي ينفذ أو يستدعى قبل ال body ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أين يمكنني كتابة الجافا سكربت

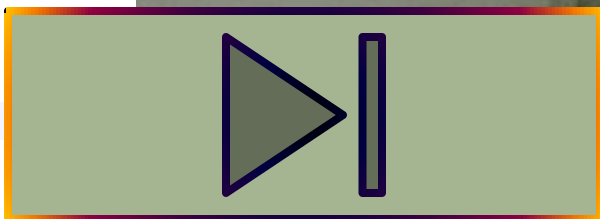
١) كتابة ال script داخل ال head:

```
<head>
    <title>ANEES HIKMAT (JS PAGE)</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
    <!-- Head Script-->
    <script>
        alert("anees");
    </script>
</head>
```

جافا سكربت كود لاظهار رسالة

٢) كتابة ال script داخل ال body:

```
<body>
    <div id="page-1">
        Hi Anees Hi Anees Hi Anees Hi Anees Hi
        Hi Anees Hi Anees Hi Anees Hi Anees Hi
        Hi Anees Hi Anees Hi Anees Hi Anees Hi
    </div>
    <!-- Body Script-->
    <script>
        alert("Hello man");
    </script>
</body>
```



aneeshikmat@gmail.com أنيس حكمت أبوحميد

أين يمكنني كتابة الجافا سكربت

٣) كتابة ال Script داخل ملف خارجي : الآن ..كما رأينا فإنه يمكننا كتابة ال script داخل ملف خارجي يستدعى عن طريق ال <script src=“”></script> ..وهذا الاستدعاء أيضا يمكن أن يكون داخل ال head أو داخل ال body .^_^

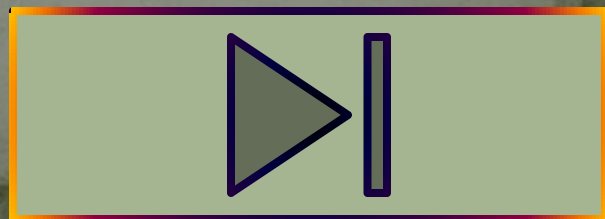
شاهد مثالا:

```
<head>
    <title>ANEES HIKMAT (JS PAGE)</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
    <!-- Call external Script-->
    <script src="script.js"></script>
</head>
```

2.html | script.js

```
1   alert("Anees Will be call external script");
```

ملف الجافا سكربت.. لكن لاحظ أنني لم أستخدم <script> tag </script> داخل هذا الملف ^_*



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript output

بما أننا نتعامل مع لغة برمجة، فإنه من المهم أن نعرف كيف يمكننا طباعة النتائج، وفحص القيم أثناء وقبل وبعد التنفيذ .. ولذلك يجب أن نتطرق الى طرق طباعة أو اظهار البيانات أو القيم عن طريق الجافا سكربت..

أولا: يمكننا اظهار النتائج من خلال تعديل احدى محتويات ال html الموجودة وذلك يكون باتباع خطوتين ..أ) document.getElementById(*id*) و
ب) استخدام innerHTML

ال document.getElementId(id) وهنا يتم تحديد عنصر ال html بنائا على ال id الذي تم وضعه ... أما ال innerHtml فهذه تقوم بالتأشير على محتوى هذا العنصر ^_*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript output

ملاحظة: سيتم وضع الشيفرة البرمجية الخاصة بالأمثلة بنفس صفحة ال html.. الا اذا كان السكربت ..طويل أو به فكرة معينة، علما أنني أستخدم هذا الأسلوب الآن لغايات تنظيف وسهولة قرائة وتنفيذ الأمثلة ..والعمل الحقيقي يكون غالبا باستخدام ال js external file .. الا في بعض الأكواد السريعة أو الخفيفة أو الاستدعاء أو الحالات الخاصة .. وجميع هذه الأمور ستتعلمها لوحدك ..ويكفيك أن تعرف الآن كيف تستخدم الجافا سكربت ..^_*

الآن لنعود ونشاهد مثالا على استخدام الطريقة الأولى لطباعة القيم وهي عن طريق تعديل محتوى أحد عناصر ال html ..

```
<!-- Call body Script-->
<script>
    document.getElementById("span-2").innerHTML = "This is JavaScript outPut, span-1 was changed";
</script>
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript output

اذا انتبهت الى التعليق الموجود في الصورة السابقة ..فستجد أنني كتبت body script ..لماذا .. لأنه اذا قمت باستدعاء هذا ال script قبل انشاء عناصر ال html.. فلن أستفيد شيئا ...

```
    <!-- Call head Script-->
    <script>
        document.getElementById("span-1").innerHTML = "This is JavaScript outPut, span-2 was changed";
    </script>
</head>

<body>
    <div id="page-3">
        <span id="span-1">External Script call ^_*</span><br />
        <span id="span-2">External Script call ^_*</span>
    </div>

    <!-- Call body Script-->
    <script>
        document.getElementById("span-2").innerHTML = "This is JavaScript outPut, span-1 was changed";
    </script>
```

لن يتم تعديل المحتوى

سيتم تعديل المحتوى

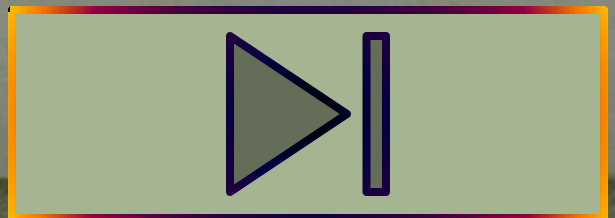
الآن أريدك أن تنتبه للشريحة التالية^^

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript output

بما أننا قمنا باستدعاء أو طلب تغيير لعنصر مش موجود أو غير معرف بعد ..فإن النتيجو هو ظهور خطأ ... لكن كيف يمكنني رؤية الأخطاء الناتجة من السكربت ..لاحظ الصورة ..

الآن شاهد المثال ... وأرجو أن تنظر الى الخطأ... ^_^



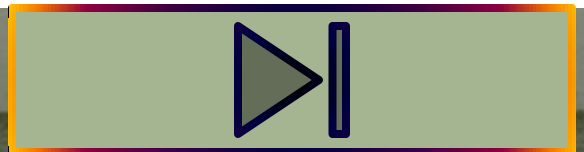
aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript output

ثانيا: استخدام ال document.write().. هذه الدالة تقوم على اضافة أو كتابة محتوى الى صفحة ال html.. (اختصار ال document يرمز الى ما يتعلق بملف أو صفحة الويب) ، ولكن لا تستخدم هذه الدالة الى اذا بدك تعمل فحص لعنصر أو خاصية معينة ...والسبب في ذلك أنها تقوم باعادة تحميل أو كتابة عناصر ال html الموجودة داخل الصفحة ...طبعا يمكننا استخدام ال html tag داخل ال innerHtml وال write.. *_^

شاهد المثال:

```
<!-- Call body Script-->                    لاحظ كيف أنني استخدمت ال html tag داخل ال write ^_*
<script>
    document.write("This is JavaScript outPut by write method, this is <b>Last line</b><br />");
</script>
```



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript output

ثالثا: باستخدام ال console.log: هذه الطريقة أهم طريقة من بين الطرق.. وأروعها وأفضلها ^_^..على الأقل بالنسبة لي :Pp ...

تظهر النتائج باستخدام ال debug الموجود داخل أي متصفح (عن طريق الضغط ب F12) ومن ثم نختار ال console ^_^

Look to console by press F12 then select console tab

تفكر ...

سقوط الإنسان ليس فشلا
ولكن الفشل
أن يبقى حيث سقط !!

توماس آديسون

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Syntax

لكل لغة برمجة بالعالم، قواعد يجب الالتزام بها لكتابة أي شيفرة برمجية ..ومن هذه اللغات الجافا سكربت بكل تأكيد ..ويقصد بال syntax هو المبادئ أو القواعد التي من خلالها يمكننا كتابة الأوامر البرمجية، وهذه المبادئ أو القواعد تكون محددة مسبقا، ويجب معرفتها قبل البدء ببرمجة أي لغة ...

*) ملاحظة: الجافا سكربت لغة برمجية خفيفة، مقارنة مع لغات البرمجة الأخرى، لكنها قوية ^_^ *

*) ملاحظة٢: إن الشيفرة البرمجية الناتجة من ال syntax هي عبارة عن جملة برمجية، هذه الجملة تسمى statements ومجموعة الجمل تسمى code

*) ملاحظة ٣: الجافا سكربت حساسة لحالة الأحرف (case sensitive) مثلا Var لا تساوي var...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Syntax

قواعد أو هجائيات اللغة للأنواع المختلفة:

١) الأرقام: تكتب الأرقام بالجافا سكربت بثلاثة صيغ رئيسية وهي:
أ) أرقام صحيحة مثل ٥٧٨٢
ب) أرقام عشرية مثل ٥,٣ وانتبه أن الفاصلة العشرية هي (. نقطة) ^_^ *
ج) باستخدام الرمز e مثل 155e5 هذه الكتابة هي الأسلوب العلمي لتمثيل الأرقام بالحاسوب' خصوصا للأرقام الكبيرة والمرفوعة للقوة العاشرة وأكثر.. لذلك تجدها كثيرا في الآلات الحاسبة والحواسيب ...

٢) النصوص: يقصد بالنصوص هو أي حرف أو رقم أو نص داخل " أو ' (double quote, quote) مثل: "^_^ 91 anees"

٣) التعابيير الحسابية : يقصد بالتعابير الحسابية هي العمليات الحسابية مثل الطرح والجمع والضرب والقسمة ..مثال: 5*5 ^_____^

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Syntax

٤) المصفوفات (array): هي **مجموعة** من الأرقام أو النصوص التي تخزن داخل متغير واحد. مثل ["anees","taher","saed"] ...

٥) Object(الكائنات): أريدك أن تعرف أن المصفوفات أو ال function أو الأرقام حتى داخل الجافا سكربت هي object ...ويمكن تعريفها على أنها البيانات فقط، مضافا اليها ال method والخصائص (properties) مثل {Age:"24", Name:"Anees^_*"}

٦) ال function: هو نطاق أو حيز تم حجزه لتنفيذ وظيفة محددة أو معينة، ويتم تنفيذ هذه الوظيفة من خلال استدعاء هذا ال function...

٧) المتغيرات variables: وهي حرف أو مجموعة من الحروف التي يتم حجزها كإسم برمجي يدل على أمر ما، ليتم استخدامه لحفظ قيمة معينة متغيرة غالبا للقيام بوظيفة معينة مثل var age = 24; السنة القادمة يكون age = age + 1 وهو ٢٥

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Syntax

٨) يمكن استخدام العمليات الحسابية والعمليات المنطقية مثل الرياضيات، بالإضافة الى ذلك يمكن اسناد أو ارجاع القيمة مثل استخدام ال = *_^
أمثلة: sq = 5 * 5 أيضا flag = 5 < 3 أو x = 4 ...

٩) ال statements: وهي مجموعة الجمل البرمجية (هل تذكر ذلك) مثل
x =5 * 5;
y= 6 * 6; .... هذه المعادلتين أو الجملتين هما JavaScript statments

ملاحظة: يرمز للجافا سكربت ب JS ^_^

١٠) ال keywords: هي الكلمات الخاصة باللغة، والتي لا يمكن استخدامها الى لهدف معين داخل اللغة .. مثل var هذا keyword يستخدم لتعريف متغير ..
مثل var age = 24

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Syntax

١) Identifiers(المعرفات): وهي أي اسم اطلق على function, variable , object ويجب أن يكون فريد(unique) أي غير مكرر ، وهذا يعني أن اي Identifiers يجب أن يكون unique ..

من قواعد التعريف للمتغيرات أو ال function أو ال object (أو بصيغة أخرى قواعد أي Identifiers) هي أن يسمى بحروف، ويمكن أن يحتوي أرقام ولكن يجب أن لا يبدأ برقم، ويمكن استخدام ال (_) underscore أو ($) dollar

٢) المتغيرات داخل الجافا سكربت يمكن أن تحتوي أكثر من نوع من البيانات مثل var x= 5 أو var y = “anees” ...الخ

٣) أي شيفرة برمجية يتم كتابتها داخل function يمكن استخدماها أكثر من مرة وبدون تقييد بعدد محدد من المرات ^_^

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Syntax

٤١) الفراغات أو المسافات (white space): الجافا سكربت يتجاهل الفراغات الاضافية بالشيفرة البرمجية، وهذا الأمر جميل جدا.. بحيث يفيدك هذا الأمر بتنسيق الشيفرة البرمجية بطريقة أكثر سهولة بالقرائة مثل
var x=5 و var x = 5 ... الطريقة الثانية أسهل أو أريح بالنظر.. وخصوصا عند وجود شفيرة برمجية كبيرة أو متعددة ...

٥١) عند كتابتك أي سطر برمجي في الجافا سكربت أو بغيرها .. لا تزد حجم السطر الواحد عن ٨٠ خانة محجوزة(٨٠ حرف أو رقم أو مسافات ..الخ)

٦١) في حال وجود أسطر برمجية طويلة تفوق بحجمها ال ٨٠ حرف نقوم بفصل السطر البرمجي الى جزئين، هذا الفصل إما أن يكون بعد اشارة ال (=) أو الفصل في حالة وجود جملة نصية (أي جملة تقع بين ” أو ‘)..شاهد المثال بالصفحة التالية ^_* —

aneeshikmat@gmail.com أنيس حكمت أبوحميد

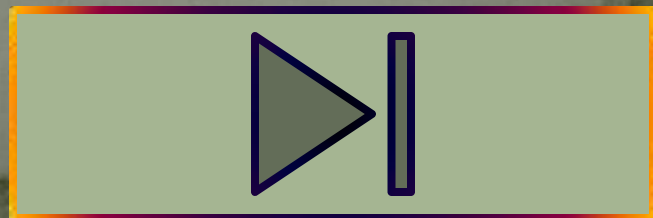
JavaScript Syntax

```
<script>
    document.getElementById("id-h1").innerHTML = "This is JavaScript outPut, \
            page-6 was changed";

    document.getElementById("id-h2").innerHTML =
          "This is JavaScript outPut, page-6 was changed h2";

</script>
```

١)هل لاحظت كيف يتم فصل الأسطر... الأولى لقد استتخدمنا ال backslash (\ ) بينما في الثانية قمنا مباشرة بعد المساواة بفصل الجملة ^_^



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Syntax

الآن أنت قد تقول ... لماذا كل هذا الكلام.. ولماذا ذكرته هنا على شكل نقاط .. ولماذا لم أتجاوزه وأبدء بالبرمجة مباشرة ...

والجواب بكل تأكيد ...أنني لن أستطيع أن أقوم ببرمجة أو قرائة أو تعلم أي لغة برمجة بدون تعلم قواعد وهجائيات التفكير لكل لغة..والبنية التركيبية لها .. هل يمكنك أن تبدأ بصناعة الطائرة الورقية دون معرفتك بما تحتاجه من خشب ولاصق وورق ؟! ..**طبعا لا** ..(كان نفسي من زمان أعمل طائرة ورقية ..بس ولا مرة ساويت وحدة وزبطت :P)..وهكذا هي البرمجة ...

الكلام الذي سبق مهم جدا الآن ..اذا لم تركز بما قلت ولم تركز على الأمثلة المختصرة التي وضعتها ..أرجوا أن تعود وتعيد القرائة .. أريدك أن تبدء بقوة ^_^..

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Statements

هل تذكر الأمثلة الاولى التي طرخناها بهذه الدورة ..هل تذكر ال innerHtml ؟! هذه جميعها JS statements ^_^

والآن السؤال القوي.. ما هو الأسلوب الذي يمكن أن يفصل بين كل statements و statements أخرى ؟

الجواب هو (الفاصلة المقوطة ;)... والسؤال الآن هل هذه الفائدة الوحيدة للفاصلة المنقوطة ؟! ...

أيضا الجواب لا ^_^* ... فهي تمكننا أيضا من كتابة أكثر من statements على سطر واحد مثل age = 24; name="anees";year="1991"

ملاحظة: الأسلوب الصحيح أو الدارج لكتابة الجافا سكربت هو بوضع الفاصلة المنقوطة بنهاية كل statements لكنها بالحقيقة ..ليست الزامية ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Statements

الآن ماذا نسمي مجموعة ال Statements ؟ مجموعة الجمل البرمجية تسمى code

أيضا تسمى مجموعة الشيفرة البرمجية الموجودة داخل ال block مثل ال function ب Code Block.

هناك الكثير من ال identifier والتي تعتبر statements مثل ال for loop أو while أو switch ... الخ ..(سيتم ذكرهم في الشرح بإذن الله تعالى ..)

والآن لنتابع ^_^

تفكر ...

يجب أن تكون عندنا مقبرة جاهزة لندفن
فيها أخطاء الأصدقاء

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Comments

هذا الموضوع المهم والرائع والبسيط ^_^... تكلمنا عنه في ال html وفي ال css وحان دوره الآن ..لنتحدث عنه داخل الجافا سكربت..

بنفس الوظائف وبنفس الهدف من استخدام ال comment داخل ال css أو ال html نستخدمه داخل الجافا سكربت.. فهو يقوم بشرح وتوضيح الكثير من الجمل والأسطر البرمجية، بالإضافة الى الوظائف المتغيرات أو functions ..أول أي code أو code block...

في الجافا سكربت ...هناك اسلوبين لكتابة ال comment ..وهما:

١)ال Single Line comment: ويكت إما على شكل inline-comment أو oneline-comment ويكون هذا الأسلوب باستخدام ال (//) مثل --->

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Comments

```
<script>
    // -this is variable :P - ((( OneLine comment )))
    var setX = 5;
    var setY = 10; // This is variable too :P ((( Inline comment )))
</script>
```

Single Line Comment Example

أما الطريقة الثانية فهي multi-line comment: ونستخدم هذا الأسلوب لكتابة أكثر من تعليق، وهذا الأسلوب غالبا ما يستخدم لشرح لوظيفة Code block أو شرح document ..ويستخدم بكثرة مع ال API .. والآن لنرى كيف يمكننا استخدامها ..

```
/*
* this function will be do more of thing
* this function have a comment
* this function developed by Anees :P
*/
function myFunctionName(){
    // Any Code you needed
}
```

الزامي

لتوضيح الأسطر فقط
أسلوب تنسيقي فقط
وغير الزامي

الزامي

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Comments

أثناء تطوير المواقع أو البرامج فإن التوثيق يلعب دورا مهما جدا، وهو دور آخر نستخدمه كمطورين ..وهو تعطيل الشفيرة البرمجية وتفعيلها ... ^_^ وذلك لغايات فحص النتائج ... هذا الأسلوب يتضح أثره وأهميته أثناء كتابتك للشيفرة البرمجية، وخصوصا في المراحل القادمة ..مثل تعاملك مع ال loop أو جمل الشرط ...الخ

```
var setX = 5;
setX = setX + 20;
//setX = setX * 20;
console.log(setX);

// (or)
var setX = 5;
/*
setX = setX + 20;
setX = setX * 20;
*/
console.log(setX);
```

في هذا المثال قمت بتعطيل سطر برمجي ..لأرى ما هو الناتج قبل عملية الضرب ..

في هذا المثال قمت بتعطيل الأسطر البرمجية الاخاصة بالعمليات الحسابة لأرى القيمة الأولية (الافتراضية) للمتغير

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Variables

المتغيرات .. الآن لنبدء فعلا وبقوة ..بالدخول الى لغة البرمجة بشكل أكبر، وخصوصا أننا الآن أصبحت لدينا معرفة حول التفاصيل أو القواعد الخاصة بهذه اللغة بشكل عام.. والآن ننتقل للجزء الثاني والعملي في الدورة ... هيا بنا أيها المبرمج العظيم ^_*

المتغيرات هو اسم رمزي يحتوي على عدد معلوم أو غير معلوم من المعلومات، هذه المعلومات يطلق عليها اسم ال "قيمة value"، ويتم حجز مساحة داخل الذاكرة مخصصة لهذا المتغير.

يقص بالإسم الرمزي الاسم الذي يمكن أن يتكون من حرف أو مجموعة حروف بالإضافة الى الأرقام أو _ أو $ ضمن قواعد محددة، وعادة يدل اسم المتغير على وظيفته ..مثال var age = 24; ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Variables

ويقصد بالقيم أو المعلومات هو ما سيتم حفظه داخل المتغير مثل

... var age = 24;

فكرة المتغيرات في فكرة رياضية (رياضيات) ..فكما نعلم وكما تعلمنا جميعا في المراحل الدراسية وخصوصا في حل المعادات ..أن نستخدم س و ص كمتغيرات لحل أي معادلة ..

مثل س + ص = ٢٥ .. في الجافا script بنفس الطريقة x + y = 25 ^_^

تذكير: انتبه للقواعد التي تم ذكرها سابقا حول شروط كتابة المتغير .

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Variables

والآن كيف يمكننا تعريف أو انشاء متغير؟
أولا ..عملية انشاء المتغير تسمى declaring
ثانيا.. لانشاء متغير نستخدم كلمة محجوزة من كلمات اللغة وهي var
ثالثا..

```
var varName;
varNmae = "ANEES";
```

نقوم بتعريف المتغير, ويمكن أن يكتب بالشكل التالي

أو

```
var varName = "ANEES";
```

أن يعرف المتغير وتسند له قيمة مباشرة مثل

عند تعريف المتغير ومن ثم اسناد القيمه له (الطريقة الأولى) فإن القيمة تكون undefined.

ملاحظة: undefined يعني متغير لا يوجد له قيمة ..
Null: يعني مغير لا يوجد له قيمة ...(هناك فرق بينهم طبعا اذا أحببت .. ابحث عن الموضوع تحت عنوان difference between null and undefined in JavaScript)

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Variables

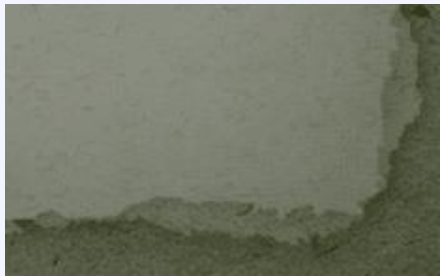
*) يفضل أن تقوم بانشاء المتغيرات جميعها في أعلى السكربت...

يمكن يتم تعريف أو كتابة المتغيرات كل متغير على حدى كالمثال السابق،
ويمكن كتابتها على شكل جمل ..شاهد هذا المثال:

```
/*
* this example will be display a variable declare in one statement
* also, its will be display a variable with an initialize value
*/
var varName, varName1, varName2, varName3;// using comma
var varName4 = "hi", varName5 = 10, varName6;// using comma with initialize
var varName7; var varName8;var varName9;// using semicolon
//var varName10;varName99; // this hide since its error
```

```
varName = "anees";
varName1 = 5;
varName2 = 2.5;
varName3 = "Hikmat";
```

اسناد قيم للمتغيرات



aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Variables

```
varName6 = varName1 + varName2;
varName7 = varName + varName3;
varName8 = varName6 * varName5;
varName9 = 'Javascript Variable <span style="color: red;"><br />var varName, varName1, varName2, varName3;<br />\
            var varName4 = "hi", varName5 = 10, varName6;<br /> \
            var varName7; var varName8, varName9;</span><br />'
//var varName10;varName99;

document.getElementById("all-var").innerHTML = varName9 + " <br /> " + varName6 + " <br /> " + varName7 + " <br /> " + varName8;
/script>
```

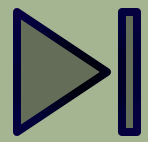
مجموعة من العمليات الحسابية على المتغيرات ...
ومن ثم طباعة النتائج داخل صفحة الويب...
لاحظ .. أن جملة الطباعة ..قمنا بشرحها قبل ذلك .. وبالنسبة للمتغير رقم 9 .. فهذه تم شرحها أيضا ..وهي طريقة كتابة جمل على أكثر من سطر ^_*... أما بخصوص 6 و 7 و 8 فهي عمليات جمع وضرب مع ملاحظة أن النصوص يمكننا استخدام عملية الجمع بينهم ..وتكون عملية الجمع هنا بمعنى ضم النصين بجانب بعضهما ..

يرجى الانتباه جيدا للمثال، والنظر الى الشيفرة البرمجية بعناية وتمهل...
ويرجى القيام بكتابة متغيرات بجميع الطرق التي ذكرت ..

Javascript Variable
var varName, varName1, varName2, varName3;
var varName4 = "hi", varName5 = 10, varName6;
var varName7; var varName8, varName9;

7.5
aneesHikmat
75

النتائج ... لا تنسى أننا تعلمنا ال
HTML & CSS
والآن جافا سكربت ...



aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Variables

والآن بعد مشاهدتنا للمثال السابق... هل خطر ببالك سؤال؟ماذا لو قمنا بإنشاء المتغير مجددا ؟

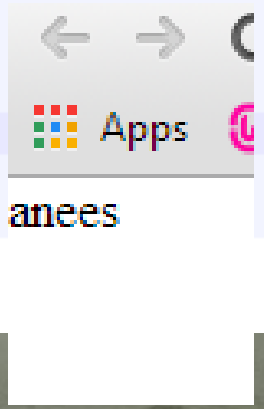
الآن لو قمت باعادة انشاء المتغير.. فستكتشف ..أنه لن يعطيك أي خطأ.. بالإضافة الى ذلك ..فإن القيمة المخزنة داخل المتغير لن تفقد ..

```
<body id="page-8">
    <div id="all-var"></div>

    <script>
        var varName = "anees";
        var varName;
        var varName;
        var varName;

        document.getElementById("all-var").innerHTML = varName;
    </script>
</body>
```

لاحظ أنني هنا قمت باتشاء المتغير أكثر من مرة ... القيمة المخزنة داخل المتغير هي anees
اذن سيكون الناتج هو anees



قم بتطبيق المثال ^_^

تفكر ...

الوقت كالسيف إن لم تقطعه قطعك

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Data Types

يوجد أنواع مختلفة من البيانات لكل لغة برمجة ... أنواع البيانات هذه مهمة جدا في البرمجة، وخصوصا في تحديد النوع المناسب لكل متغير، وهذا يتضح في اللغات التي يجب أن تحدد بها النوع ..

الجافا سكربت لا تشترط منك تحديد نوع البيانات (Dynamic Types).. لذلك فالأمر قد يكون أسهل نوعا ما، ولكن تبقى ملزما بمعرفة الأنواع والفروق أو الاستخدامات لكل من هذه الأنوع ..

أنواع البيانات الموجودة داخل الجافا سكربت هي(ذكرت اسم التصنيف للمعرفة):

١) ال primary (primitive) data types: وهي ال
String, Boolean, Number

٢) االل composite (reference) data types: وهي ال array, object

٣) ال special data type: وهي ال Null, undefined..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Data Types

والآن سنتلكم بإذن الله تعالى عن هذه الأنواع بالترتيب التالي:

String, Number, Boolean, Array, Object, Null, Undefined

١) ال String: هذا النوع يقوم بحفظ مجموعة أو سلسلة من الحروف أو الأرقام أو الرموز الخاصة، ويتم تضمين هذه السلسلة داخل " or ' ...

مثال:

```
var varString = "anees";
```

الآن ماذا يمكننا أن نستفيد من استخدام ال ' أو " ..ولماذا وجد خياران وليس خيار واحد ... الجواب ..أن هذا الأمر الجميل يسمح لك باستخدام ال " أو ' داخل نفس النص، وفي اللغات الأخرى مثل ال php ممكن أن تفيد بأخذ القيم من المتغيرات من عدمه ... أمر بسيط لكن مهم .. لنشاهد أمثلة على ذلك

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Data Types

```
varString = "anees hikmat anees abu-hmiad";  " "
varString = 'anees hikmat anees abu-hmiad';  ' '
varString = "anees 'hikmat' anees abu-hmiad"; " ' ' "
varString = 'anees "hikmat" anees abu-hmiad'; ' " " '
varString = 'anees \'hikmat\' anees abu-hmiad'; ' \' \' '
varString = "anees \"hikmat\" anees abu-hmiad"; " \"\" "
```

قم بتطبيق الأمثلة ..^_^..(ملاحظة ..فيما يتعلق بالمعلومات حول انواع البيانات هذه، فإننا نقوم الآن بذكر فكرة بسيطة عن كل نوع، لكن عندما نتقدم بالمستوى قليلا ..سنتلكم عن هذه الأنواع ..بمستوى أعلى نوعا ما ...)

٢)ال Number: الأرقام اذا كنت تذكر ..فهي يمكن أن تكون صحيحة، أو يمكن أن تقبل على شكل عشري باستخدام ال (.) بدلا من الفاصلة.. أو يمكننا استخدام ال (e) كرمز حاسوبي للقوى ...شاهد المثال:

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Data Types

```
// Number Example
varNumber = 5;
varNumber = 5.5;
varNumber = 2e10;
varNumber = 2e-10;
```

صحيح
عشري
للأرقام الكبيرة بالحاسوب
للأرقام الصغيرة بالحاسوب

لا تنسى تطبيق المثال ^_^... ملاحظة: جرب القيام بعملية طرح لل varNumber = 2 – 2e10; ^_^..وانظر الناتج *_

٣) Boolean: هذا النوع رائع ..وسهل جدا ..فهو يحتوي قيمتان فقط ..إما ture (صح) أو false (خطأ)، ويستخدم هذا النوع من البيانات غالبا للتحقق من شرط معين ^_^

```
// Boolean Number
varBoolean = true;
varBoolean = false;
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Data Types

٤) array: المصفوفات هي نوع من أنواع البيانات التي يمكن استخدامها داخل الجافا سكربت، وهي مهمة جدا.. وتعرف على أنها طريقة أو وسيلة لتنظيم مجموعة من البيانات مثل الأرقام والأحروف..الخ، وترتبط دائما بمفهوم موقع وقيمة key/value ... ويتم استخدام ال [square brackets] كأداة احتواء للمصفوفة ..

شاهد المثال:

```
// Array Example
varArray = ["a","n","e","e","s"]; // Good Use
varArray = new Array("a","n","e","e","s"); // Bad Use
```

لانشاء مصفوفة فإننا نعتمد الطريقة الأولى، فهي أفضل وسلبياتها أقل...

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Data Types

٥) object: ويمكن تعريفها على أنها البيانات فقط، مضافا اليها ال method والخصائص (properties).. هل تذكرت أين قمنا بذكر هذا التعريف.؟! نعم ..ذكرناها في js syntax ..لاحظ كيف أن المعلومات التي ذكرت وقتها مهمة للتقدم ..وسنحتاج هذه المفاهيم أكثر فأكثر.. فأبق نفسك متابعا أول بأول.. لأننا كلما سرنا الى الأمام، احتجنا أن يدفعنا ما تعلمنا من الخلف...

مثال:

```
// Object Example
varObject = {firstName:"Anees", age:24, carType: "Still Without Car -_-"};
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Data Types

٦) Undefined & Null: هل تذكر عندما تكلمنا عن ال Null وال Undefined ؟..
لقد قلنا أن المعنى للكلمتين هو لا يوجد قيمة ...وهذا لا يعني صفر.. لأن الصفر هو قيمة
لا يوجد قيمة ..يعني لم يسند لها قيمة ... (فارغة)

إن هذان يمثلان نوعان من أنواع البيانات، نوعين مختلفين ولكن مع وجود بعض الفروقات
... هل بحثت عن الفروقات ؟!...

حقيقة احترت هل أذكر الفروقات أو ما هي النقاط لذلك قررت أن أضع لك رابطا لموقع
..هذا الموقع ابقه في قلبك فهو من أروع المواقع التي يمكن أن تتعلم منها كمبرمج ...

من نقاط الاختلاف هو أن ال undefined تعطى مباشرة الى المتغيرات التي لم يتم
تعريفها ... بينما ال null لأي قيمة اعتمدت وتم تصفيرها .. والتمييز يكون باستخدام ال
== أو ال === ... لأن = = = تعني مقارنة للقيمة والنوع معا .. بينما ال = = مقارنة
للقيمة بدون النوع.

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript typeof Operator

الآن ... هل الطريقة الوحيدة لمعرفة نوع البيانات هو عن طريق الخبرة البرمجية أو معرفة الأنواع من خلال النظر ؟! ..الجواب طبعا لا ..

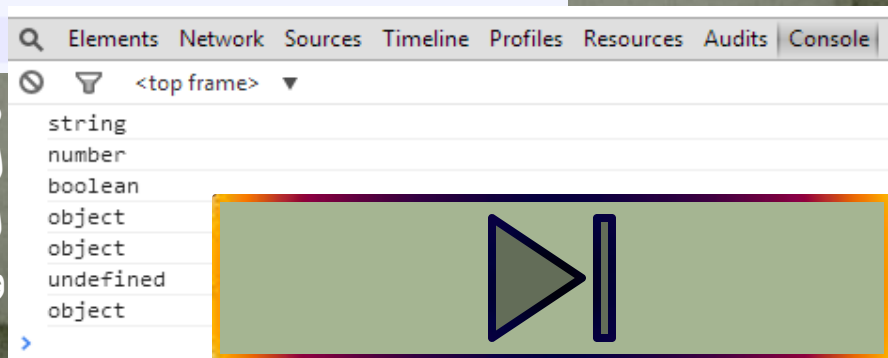
هناك وسيلة يمكن استخدامها لمعرفة أنواع البيانات للمتغيرات ..

مثال:

```
// Using typeof operator
console.log(typeof varString);
console.log(typeof varNumber);
console.log(typeof varBoolean);
console.log(typeof varArray);
console.log(typeof varObject);
console.log(typeof varUndefind);
console.log(typeof varNull);
```

اسم المتغير

هنا قمت بطباعة أنواع البيانات داخل ال debug
ويمكنك استخدام أكثر من طريقة لعرض النتائج _*



هذا المثال مهم جدا جدا جدا ... انظر الى نتائج تطبيق هذا المثال وانظر الى console عن طريق F12... بعد ذلك افتح على الشيفرة البرمجية لهذه الصفحة واقرأ الأوامر وطرق كتابة الكود..وانتبه للتعليقات ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Object

الآن ..بعد انتهائنا من الحديث عن المتغيرات وأنواع البيانات.. لنأتي بنوع من التفصيل حول كل نوع .. خصائصه ..استخدامته .الخ

وأول ما نبدء به هو ال Object.. لقد قمنا بتعريف ال Object ..وذكرنا كيف يمكن أن نستخدم هذا ال object ..

والآن لماذا يمكن أن نستخدم ال Object .. ؟؟؟

اذا جئنا للحياة الواقعية .. فإن هذه الحياة هي class فيه جميع الخصائص والدوال التي يحتاجها من يعيش ومخلوق في هذه الحياة .. ال object يمثل أي كائن أو مخلوق يرث أو يحتاج الى هذه الدوال أو الخصائص، وقد يشترك في هذه الخصائص أكثر من شخص ولكن بقيم مختلفة وهذا يعني خصائص واحدة بقيم مختلفة ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Object

ومن الأمثلة المشهورة والتي تضرب بكثرة هو الإنسان ..

الإنسان يمثل object، له خصائص ودوال، والانسان ليس شخص واحد، بل هم كثر -يستثنى من ذلك بني صهيون وأبناء اليهودية ومن معهم ووالاهم-، ومع ذلك تجد أن لكل منهم اسم مختلف، وعمر مختلف، وهناك لون للبشرة، وهناك خصائص الشعر...الخ ..-هذه الخصائص- أما الدوال فالإنسان ليس ثابت فيتحرك.. ويمكن أن يقف ..ويمكن أن يجلس.. ويمكن أن يسرع ...الخ هذه النقاط وهذا المثال الذي ذكرته...

الآن بنفس الأسلوب الذي رأينا به هذا الانسان .. فإنه يجب أن نرى غيره .. ما رأيك أن تقوم بصنع مثال يتحدث عن السيارة ..(السيارة تمثل object)

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Object

يمكننا تعريف ال Object بأكثر من طريقة .. ولقد قمنا بذكر الطريقة التي سنستخدمها .. والأفضل وهي عن طريق استخدام ال .{curly braces.}

وهناك طريقة أخرى عن طريق استخدام ال new...

أيضا يوجد هناك أكثر من طريقة لاسترجاع القيم من ال Object ..الطريقة التي قمنا باستخدامها هي عن طريق اسم ال object ومن ثم نقطة ومن ثم ال key .. مثال: varObject.keyName (هذه الطريقة التى أفضل استخدامها...أما الطريقة الثانية فهي:

```
// Object Example
varObject = {firstNa                es", age:24, carType: "Still Without Car -_-"};
varObject['carType']; (or) varObject.carType;
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Object

جميع الذي تم ذكره .. كان يخص الخصائص.. لكن ماذا عن الدوال؟ ..هل يمكن الوصول اليها من خلال ال object ؟

الجواب هو نعم ^_^.. وذلك من خلال اسم ال obj ثم نقطة ثم اسم الدالة ثم ()

شاهد هذا المثال:

```
// Object Example
varObject = {firstName:"Anees", carType: "Still Without Car -_-",
  age: function(){return 23 + 1;} };
var CurrentAge = varObject.age();
```

النتائج 24

تعريف دالة

استدعاء الدالة

بخصوص ال function سيتم شرحها في الدروس القادمة بإذن الله تعالى لاكن الآن ..انظر الى طريقة الاستدعاء عن طريق ال object ..وقم بنسخ نفس المثال وتطبيقه عندك ..وانظر النتائج ^_^

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Object

أتوقع أنه لديك سؤال مهم..

لماذا كنت أقول دالة (method) ولم أقل function؟ .. والجواب هو أن ال function يطلق عليه function خارج اطار ال object .. أما اذا كان ال object هو بنفسه يحتوي على function. فيصبح ال function هذا كأنه خاصية وهذا يسمى دالة (method)

وبهذا نستطيع بالبرمجة اذا قيل لك أن هذه دالة (method) مباشرة أنك لن تستطيع استخدام هذا ال method(ال function داخل ال obj) الا عن طريق object .. أما الFunction لوحده ..فيمكن استدعائه عن طريق اسم ال function ^_*

تفكر ...

قد أخرج البخاري، والترمذي وغيرهما عن جابر بن عبد الله رضي الله عنهما قال: كان رسول الله صلى الله عليه وسلم يعلمنا الاستخارة في الأمور كلها، كما يعلمنا السورة من القرآن يقول: إذا هم أحدكم بالأمر فليركع ركعتين من غير الفريضة، ثم ليقل: اللهم إني استخيرك بعلمك، وأستقدرك بقدرتك، وأسالك من فضلك العظيم، فإنك تقدر ولا أقدر، وتعلم ولا أعلم، وأنت علام الغيوب، اللهم إن كنت تعلم أن هذا الأمر خير لي في ديني ومعاشي وعاقبة أمري، أو قال: عاجل أمري وأجله، فاقدره لي ويسره لي ثم بارك لي فيه، وإن كنت تعلم أن هذا الأمر شر لي في ديني ومعاشي وعاقبة أمري، (أو قال: عاجل أمري وآجله) فاصرفه عني، واصرفني عنه، واقدر لي الخير حيث كان ثم رضني به. ، قال: ويسمى حاجته، أي يذكر حاجته عند قوله: اللهم إن كنت تعلم أن هذا الأمر، فيقول مثلا: اللهم إن كنت تعلم أن سفري أو زواجي من فلانة.... إلخ خير لي في ديني... وإن كنت تعلم أن سفري.... إلخ شر لي في ديني....

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Functions

ال Code Block ^_* ... هل تذكر عندما تكلمنا عن ال syntax وال statements وال code ومن ثم وصلنا الى code block .. نحن الآن بصدد الحديث عن ال code block...

ال function هو code block وهو وسيلة صممت لإداء وظيفة محددة، ويتم تنفيذ هذه الوظيفة لحظت الاستدعاء ...لاحظ المثال ومن ثم قم بتطبيقه ^_*

```
function CodeBlockFunction(){
    var x = 5, y = 6;
    return x + y;
}
alert(CodeBlockFunction());
```

اسم ال function

ما سيتم ارجاعه من نتائج

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Functions

اذا لاحظت فإن الصيغة العامة لكتابة ال Function هي:

function (keyword) ثم اسم ال function ثم ()

أما قواعد كتابة اسم ال function فهي: نفس القواعد الخاصة بكتابة أسماء المتغيرات *_*.... راجع القواعد ..اذا كنت قد نسيتها ^_*(ارجع للنقطة رقم ١١ في الجافا سكربت syntax أو اذهب الى درس المتغيرات)

الآن سأتحدث عن مصطلحين مهمين جدا وهما ال

parameter & arguments

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Functions

ال parameter هي المتغيرات التي يتم وضعها داخل ال (parentheses) الخاصة بال function ..

أما ال Arguments فهي المتغيرات أو القيم التي يتم ارسالها الى function أثناء الاستدعاء.

مثال:

```
// declare function with parameters
function functionWithParam(a, b, c) {
    return a + b + c;
}

var SomeData = 55;
document.getElementById("all-var").innerHTML =
        " functionWithParam:   " + functionWithParam(5, 6, SomeData);
```

parameters

Arguments

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Functions

والآن ..كيف يمكننا استخدام ال function ؟ أو بمعنى آخر كيف يمكننا استدعاء ال function؟

الجواب، هناك ٣ طرق وهي:

١) عن طريق ال (call) مثل الأمثلة التي ذكرناها ... فكان يوجد هناك شيفرة برمجية تقوم على استدعاء ال function ..

٢) عن طريق ال user event ، فيكون ال function مرتبط بحدث أو فعل يقوم به المستخدم مثل ال click ..

٣) أن يستدعى ال function بشكل تلقائي ^_^

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Functions

بالنسبة للطريقة الأولى ..فقد رأينا عليها أمثلة كثيرة ..(هناك أكثر من أسلوب داخل هذا الشكل، لكننا نكتفي بالشكل الرئيسي global ..وبخصوص ما تبقى يمكنك أن تتطلع عليها بعد انهائك للمستويات الأولى ^_*)..

أما بالنسبة للطريقة الثانية، فسأذكر مثالا بسيطا عن طريق button click لأنه أسهل event ..ثم سنعاود الحديث عن الموضوع بإذن الله تعالى في موضوع ال javascritp event.... شاهد هذا المثال:

```
<input type="button" id="btnFunction" value="Call Function By User Event"
       onclick="functionWithUserEvent('anees','hikmat');"/>
<script>
    // declare function by event Click
    function functionWithUserEvent(firstName, middleName){
        var ConcateName = firstName + "   " + middleName;
        document.getElementById("fun-click").innerHTML =
                " functionWithUserEvent:   " + ConcateName;
    }
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Functions

أما الطريقة الثالثة وهي Self-Invoking تقوم على استدعاء نفسها بشكل تلقائي ..شاهد المثال:

لاحظ أنه لا يوجد اسم لل Function وأن الاستدعاء تم عن طريق ال ()

```
// SELF-INVOKE
(function(){
    alert("ANEES HIKMAT SELF-INVOKE FUNCTION");
})();
</script>
```

الآن .. ما رأيك ؟...هل أصبحت قادرا على كتابة أي function واستدعائه؟

إننا كثيرا ما نستخدم ال function وهذا الأمر سهل جدا.. والآن بقي أمر آخير لم نتحدث عنه هو استخدام ال return داخل ال function ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Functions

في العديد من الأمثلة استخدمت return وفي بعض الأمثلة الأخرى لم أستخدمها...

هذه الكلمة هي من الكلمات المحجوزة باللغة، وتستخدم لإرجاع قيمة معينة ..وتوضع دائما في نهاية ال function لأنها سوف تقوم بانهاء عمل هذا ال function بارجاع قيمة معينة للشيفرة البرمجية التي استدعت هذا ال function مثال:

لاحظ أنه بوجود ال return ..فإننا نرجع القيمة الناتجة لمستقبل بغض النظر ما هو ..

** المستقبل أقصد به ال variable أو أي أمر من أوامر الطباعة ... الخ

```
// Return Example
function returnEx1(a, b){
    return a * b * 5;
}

var ex1 = returnEx1(5, 2);
document.getElementById("return-ex1").innerHTML =
        " Return Example 1:   " + ex1;
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Functions

وفي حال لم نستخدم ال return .. فإننا لا نقوم بارجاع قيمة ..لذلك فإننا نضع الاستدعاء لل function بدون أن نرجعه لقيمة أخرى ..مثل:

```
// Return Example 2
function returnEx2(a, b){
    document.getElementById("return-ex2").innerHTML =
        " Return Example 2:  " + a + "  " + b;
}

returnEx2("anees", "hikmat");
```

لا يوجد return

لاحظ .. لم يتم ارجاع القيمة ... فقط استدعاء مباشرة

من بعض استخدامات ال return أيضا ليس ارجاع القيمة فقط.. وكن ايقاف تنفيذ function معين .. مثلا عندما يوضع بال function كلمة return; ..فإنه سيخرج من هذا ال function وغالبا ما نستخدمها مع جمل الشرط في حال وقوع خطأ ولم يتم ارجاع قيمة ..والكثير..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Functions

```
return;
return true;
return false;
return x;
return x + y / 3;
```

شاهد هذه الأمثلة:

قم بتجربة ارجاع هذه القيم .. وانظر النتائج ^_* ...

الآن سأقوم بإذن الله تعالى بعرض بعض الأمثلة أيضا ..حتى نخرج من هذا الموضوع ... متقنين له .. لكن أريدك أن تقوم بتطبيق الأمثلة.. وكل ما ذكرناه أرجو أن تكون قد طبقته .. لإنك إن تقم بالتطبيق بشكل عملي... فأنت لم تتعلم شيئا ...

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Functions

```
// Return Example 3
function returnEx3() {
    console.log("A");
    console.log("B");
    return;
    console.log("C");
    console.log("D");
}

returnEx3();
```

ما هي النتائج لهذا ال function

```
// Return Example 4
function returnEx4(){
    return 4;
}

function returnEx4_1(x){
    return x * 10;
}

var result_Ex4 = returnEx4_1(returnEx4());
// plus ( + ) in java script, thats mean concat string like this a + b = ab
console.log("result_Ex4: " + result_Ex4);
```

لاحظ كيف أن ال function الأول كان هو ال arguments لل function الثاني ..

ما هي النتيجة هنا ؟

aneeshikmat@gmail.com أنيس حكمت أبوحميد

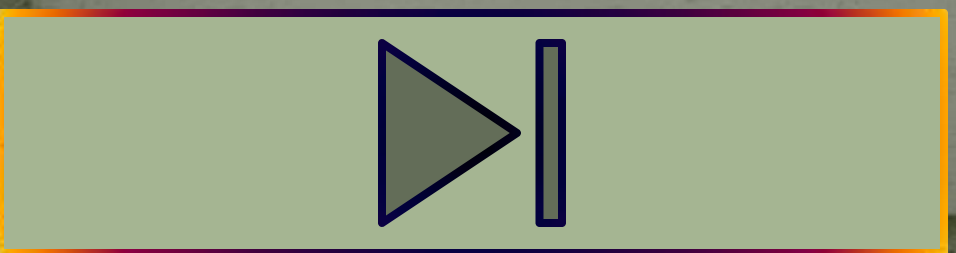
JavaScript Functions

الآن مع المثال الآخير.. والذي يقوم بتنفيذ المثال السابق بطريقة ال Self Invoke

```
// Return Example 5
(function (){
    console.log("Self-INVOKE-Function: " + returnEx4_1(returnEx4()));
})();
```

هنا استخدمت ال

والآن حان الوقت لرؤية ناتج تطبيق الأمثلة، ولرؤية الشيفرة البرمجية الخاصة بهذه الأمثلة ... (يجب أن تكون قد قمت بتطبيق الأمثلة .. والآن ستجدها هنا..اذا حصل أي خطأ لديك .. فلديك الاجابة هنا ثم عد واكتب ما أخطأت به) ^_^*



تفكر ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Scope

ال Scope ..لو قمت بترجمة هذه الكلمة لوجدت أن معناها هو نطاق.. وهنا في البرمجة تعني مجموعة المتغيرات وال function وال object التي يمكننا الوصول اليها من نقطة برمجية معينة ..

ولتبسيط الموضوع سأذكر مثالا وهو لو افترضنا أن هنالك غرفة محاطة ب ٤ واجهات من الطوب..ويوجد شخص بداخلها يشاهد التلفاز وشخص بالخارج ولا يوجد أي نافذة لرؤة الداخل أو لرؤية الخارج، فهل يستطيع الذي بالخارج مشاهدة التلفاز مع الذي بالداخل الاجابة طبعا لا، لإن نطاق الوصول لهذا الرجل لا يستطيع اختراق الجدران، وكذلك الأمر للشخص الذي بالداخل، فلا يستطيع معرفة ما يحدث بالخارج...الآن ماذا لو قمنا بوضع نافذة مظللة، بحيث يستطيع الذي بالخارج رؤية ما في الداخل؟، حينها سيستطيع الذي بالخارج مشاهدة التلفاز وبذات الوقت الذي بالداخل لن يستطيع مشاهدة الخارج ... وهكذا

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Scope

الآن يجب أن تعلم أن الجافا سكربت تملك scope ..هذا ال scope هو function scope ..وبهذا يكون المتغير أو ال scope إما

١) Local variable
٢) Global variable
٣) Auto Global

والآن لنبدء معا بهذا الموضوع السهل والمهم ^_^*

١) ال local variable : في هذا النوع يتم انشاء المتغير داخل نطاق ال function وبهذا يكون ال local variable يملك local scope وهذا يترتب عليه ..

أ) لا يستطيع أحد الوصول الى هذا المتغير واستخدامه الى ال function الذي يحتوي هذا المتغير

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Scope

ب) يمكن انشاء هذا المتغير بنفس الإسم في أكثر من function دون أن يأثر كل منهم على الآخر..لأنه لا يستطيع أحد بالخارج رؤية التلفاز الذي بالداحل ^_*

ج) مجرد استدعاء ال function فإنه يتم انشاء هذه المتغيرات..وبمجرد انتهاء هذا ال function يتم حذف المغير ^_*...شاهد المثال ١)

```
<script>

    var superName = "Hikmat";
    // Function (thats mean is scope)
    function CodeBlockFunction(){
        var userName = "anees"; // local variable => thats mean its have a local Scope
    }
    console.log("This is not local scope its a " + typeof superName);
    console.log("This is local scope " + typeof userName);

</script>
```

local variable

بما أنه يستطيع أن يصل للمتغير فهو سيطبع نوعه (string)

Undefined هنا

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Scope

المثال ٢)

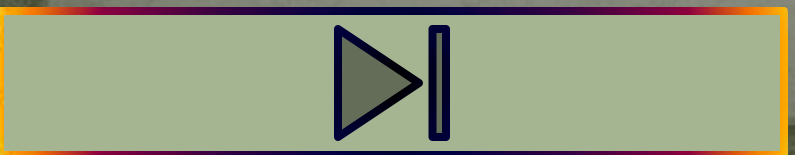
```
function funAddNum(){
    var num1 = 5;
    var num2 = 6;
    document.getElementById("fun2-ex-num").innerHTML =
            " Local Scope Result (in function):  " + (num1 + num2);
}
funAddNum();
document.getElementById("fun2-no-num").innerHTML =
        " Try to use local variable from out the function:  " + (num1 + num2);
</script>
```

هنا سيقوم بطباعة المطلوب بشكل صحيح

هنا لن يقوم بتنفيذ الجملة..
وسيطبع خطأ داخل ال
console

Uncaught ReferenceError: num1 is not defined

ملاحظة: في حالة حصول مثل هذا الخطأ ..فإن أي سكربت سيأتي بعده لن يعمل .. شاهد المثال: (مهم جدا أن تقوم بكتابة الأمثلة وتطبيقها) ..هذه الأمثلة مهم أن ترى الشيفرة البرمجية وال console ..مهم جدا ..ومهم أن تقرأ التعليقات..



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Scope

٢) ال Global variable: يتم تعريف هذا النوع من المتغيرات خارج ال function ، وبهذه الحالة يمكن أن يستخدم في جميع الأماكن ^_^ داخل السكربت، ويمكن أن يستخدم داخل ال function ...وهذا يعني أن المتغير يمتلك Global Scope...

شاهد مثالا:

```
var num1 = 10; // Global variable
var num2 = 20; // Global variable

function AddNum(){
    document.getElementById("fun2-ex-num").innerHTML =
            " Global Variable in the function: <b>" + (num1 + num2) + "</b>";
}
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Scope

مثال ٢:

```
var num1 = 10; // Global variable
var num2 = 20; // Global variable

function AddNum(){
    document.getElementById("fun2-ex-num").innerHTML =
        " Global Variable in the function: <b>" + (num1 + num2) + "</b>";
}
AddNum();

document.getElementById("fun2-no-num").innerHTML =
    " Global Variable out the function: <b>" + (num1 + num2) + "</b>";
// will be execute .. no error
alert("This execute, since no error");
```

استخدام من داخل ال function

استخدام من خارج ال function

سيتم تنفيذه

لاحظ هنا باستخدام ال Global ..استطعنا أن نستخدم المتغيرات داخل وخارج ال function بدون أي خطأ... ^_^ *

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Scope

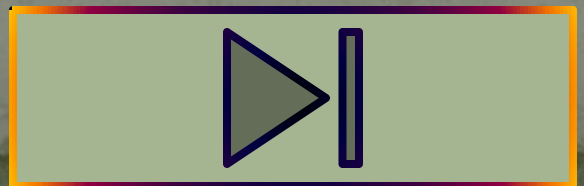
٣) Auto Global: في هذا النوع ..نكون قد أسندنا قيمة الى المتغير، دون تعريفه باستخدام ال var...وهنا بطريقة تلقائية..يقوم باعتبار هذا المتغير هو Global ..شاهد المثال:

```
var num1 = 10; // Global variable
var num2 = 20; // Global variable

function AddNum(){
    num3 = 30; // auto global, no (var) was used
}

AddNum();// function call
document.getElementById("fun2-ex-num").innerHTML =
    "Global & auto Global <b>" + (num1 + num2 + num3) + "</b>";
```

لاحظ..مع أنه داخل ال function الا أنني قمت باستخدامه خارجا ..لاحظ أنني لم أسبق كتابته ب var



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Scope

ملاحظات على ال Scope:

١) يمجرد انشاء متغير الجافا سكربت فإنه يبدأ دورته في العمل...(يقصد بالانشاء لحظة الوصول الى المتغير وحجز مساحه له في الذاكرة)

٢) ال local variable ينتهي ويحذف من الذاكرة بمجرد انتهاء عمل ال function..واذا قمنا بعمل استدعاء مجددا لل function ..فإنه يقوم بانشاء المتغير مجددا ..ويحجز له المساحة مجددا...

٣) ال Global variable ينتهي اذا قمت باغلاق الصفحة ^_^

٤) النقاط من ١ الى ٣ يطلق عليها دورة حياة المتغيرات بالجافا سكربت وهي (Java Script Variable Life Time)

٥) اذا قمت بعريف متغير على شكل parameters .. فإن هذه المتغيرات تعتبر local..(راجع درس ال function اذا كنت لا تعرف ال parameters)

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Scope

والآن لنشاهد بعض الأمثلة

حالة 1

```
var num1 = 10; // Global variable
var num2 = 20; // Global variable

function AddNum(){
    num1 = 30;
}

AddNum();// function call
document.getElementById("fun2-ex-num").innerHTML =
    "Global & auto Global <b>" + (num1 + num2) + "</b>";
```

ماذا تتوقع أن يكون الناتج هنا ؟
هل هو 30 أم 50؟
تتبع الكود سطرا بسطر..ونفذ
المثال حتى تتأكد

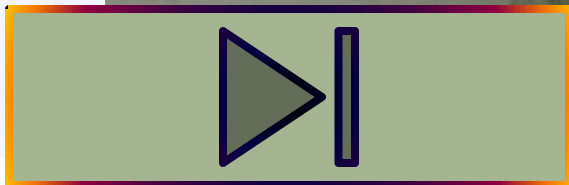
الحالة 2

```
var num1 = 10; // Global variable
var num2 = 20; // Global variable

function AddNumTow(num1, num3){
    return num1 + num2 + num3;
}

var tot2 = AddNumTow(12, 88);
document.getElementById("fun2-no-num").innerHTML =
    "Global & auto Global <b>" + (tot2) + "</b>";
</script>
```

ماذا تتوقع أن يكون الناتج هنا؟
هل هو خطأ أم 120 أم 118 أم
غير ذلك



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Console Note

ملاحظة: أثناء تطويرك للمواقع الالكترونية، وخصوصا أثناء تعاملك مع الجافا سكربت، أو أيا من المكاتب الخاصة بها ... أبقي ال console ظاهرا أمامك لتبقى على اطلاع ان حصل هناك أي خطا.. هذا أمر يسهل عليك العمل كثير..

ايضا ..لمعلوماتك ..فإنه يمكنك أثناء العمل..تطبيق السكربت مباشرة داخل ال console ...

شاهد هذا الفيديو البسيط ^_^... إنه لأمر رائع ^_^..مهم جدا للمشاهدة، وكثير من المطورين لا يعرفون عن هذه الخاصية ..وحتى بعد تجاوزهم لمستوى معين...وأنا منهم طبعا..فعرفت هذه المعلومات بعد أكثر من عام على دراستي للجافا سكربت .. ..^_^*

تفكر ...

فإن عبت قومـــــاً بالذي فيك مثلـــه
فكيف يعيب الناس من هو أعور
وان عبت قوماً بالذي ليس فيهم
فذلك عند الله والنـــــاس أكبـــر

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Events

ال events .. ^_^...يقصد بال events أي حدث يمكن أن يقوم به المستخدم أو المتصفح على أي عنصر من عناصر ال html.. وبوجود الجافا سكربت، فإنه يمكن أن تحدث استجابة معينة من خلال السكربت لهذا ال event ^_^، مثال: ال hover ^_^..هل تذكرها ..ال click.... هل تذكرها ...الخ

**) مثال على event للمتصفح: عند تحميل الصفحة ..قم بتنفيذ أمر معين.

الصيغة العامة لاستخدام ال event:

```
<input type="button" value="Click Event" onclick="alert('Event Click');" />
```

جميع الخصائص التي في هذا العنصر تم شرحها سابقا.. لكن الآن نأتي الى ال eventوهو هنا onclick ..أي عند قيامي بالنقر على الزر ..ماذا سيحدث..في هذه الصورة ..الناتج سيكون مربع نص مكتوب فيه ev click

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Events

الآن بالصيغة العامة ..يمكن استبدال ال onclick ب أي event آخر..
أما بخصوص ما سيحصل عند حدوث الحدث ..فإنه يعود اليك ..

ويمكنك استخدام أكثر من أسلوب لتنفيذ كود معين عند حدوث حدث ..

١) عن طريق تنفيذ كود الجافا سكربت مباشرة
٢) عن طريق عمل call ل function مكتوب داخل السكربت
٣) ويمكنك أيضا تنفيذ الجافا سكربت وجعل التأثير يعمل مباشرة على العنصر باستخدام ال this

والآن ..بالنسبة للنقطة واحد ..فهذا المثال الموجود بالصيغة العامة ..والآن ٢ و ٣ لنرى أمثلة ^_*

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Events

بخصوص الطريقة الثانية:

```
<div>
    <input type="button" value="Mouse Out Event" onmouseout="addmouseOutNumber();" />
    <div id="mouseOutNumber">0</div>
</div>
<script>
    var mouseOutNumber = 0;

    function addmouseOutNumber (){
        mouseOutNumber = mouseOutNumber + 1;
        document.getElementById("mouseOutNumber").innerHTML =
            " Total Leave Button is <b>" + (mouseOutNumber) + "</b>";
    }
</script>
```

هنا عند خروج مؤشر الفأرة من العنصر سينطلق الحدث, والذي سيقوم به الحدث هو استدعاء function

```
<input type="button" value="Mouse Over Event" onmouseover="this.value=addmouseOverNumber();" />
<script>
    var mouseOutNumber = 0, mouseOverNumber = 0;;

    function addmouseOverNumber(){
        mouseOverNumber = mouseOverNumber + 1;
        return mouseOverNumber;
    }
</script>
```

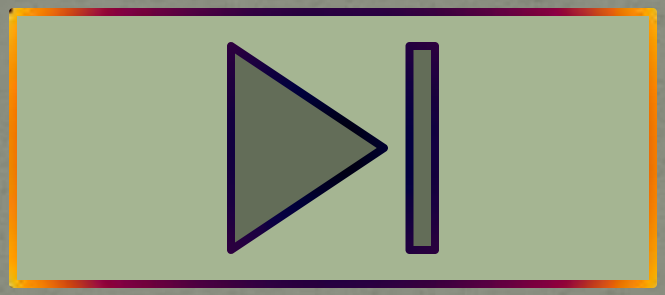
بما أن هذا العنصر له value..فإن هذه ال value هي التي أرغب بتغييرها ..ال this تعني هذا العنصر الذي أنا فيه ثم . ثم value وهذه اشارة لتغير قيمة ال value لهذا العنصر..ثم = ثم ناتج ارجاع ال function

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Events

هل قمت بتنفيذ الأمثلة السابقة ؟.. قم بتجربة ذلك ..

ثم شاهد هذا المثال:



رجائا ركز بهذه الأمثلة..أعلم أنها سهلة وبسيطة .. لكن فكر في فكرة كل مثال منهم ^_^

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Events

أهمية ال events:

لل events استخدامات كثيرة جدا وأهمها:

١) التحكم بمدخلات المستخدمين handle user input

٢) التفاعل مع المستخدمين بحيث يتم التعامل بشكل سريع مع أي اجراء يقوم به المستخدم user action

٣) التفاعل مع المتصفح والقيام بالإجراء المطلوب بكل سهولة وسرعة .. browser action

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Events

الآن لنتحدث عن ال eventsالتي يمكنني استخدامها:

١) click: وهذا الحدث ينطلق عند النقر

٢) onchange: هذا الحدث ينطلق بمجرد خروجك من العنصر ال html الخاص به، لكن بشرط أن يكون قد حصل تغيير معين ...على هذا العنصر

٣) onmouseover: وهذا الحدث ينطلق عند دخولك الى العنصر

٤) onmouseout: هذا الحدث ينطلق بمجرد خروجك من نطاق العنصر

٥) onmousedown: ينطلق هذا الحدث بمجرد النقر على زر الفأرة

٦) onmouseup: وهذا الحدث ينطلق بمجرد رقع أصبعي عن زر الفأرة

٧) onload: هذا الحدث ينطلق بمجرد أن العنصر تم تحميله.

٨) onfocus: هذا الحدث بمجرد ما تم تحديد العنصر.. وهذا الحدث يعمل مع select, <input>, <a>

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Events

٩) onblur: هذا الحدث ينطلق بمجرد خروج التحديد عن العنصر، وهو بنفس مواصفات ال onfocus ولكن هذا خروج للتحديد، وذاك دخول التحديد

١٠) ondblclick: هذا الحدث ينطلق عند قيامنا بالنقر نقرا مزدوجا (الضغط على زر الفأرة مرتين متتابعتين)

١١) onmousemove: ينطلق هذا الحدث طالما موشر الفأرة يتحرك فوق العنصر.

٢١) onkeydown: ينطلق هذا الحدث بمجرد الضغط على أي زر من أزرار لوحة المفاتيح.

٣١) onkeypress: بنفس فكرة ال onkeydown لكن هذا الحدث لا يعتبر ال shift وال ctrl وال alt..الخ من الأزرار..وينطلق فقط للأرقام والحروف والرموز الخاصة ..

٤١) onkeyup: بمجرد رفع اصبعي عن أي من أزرار لوحة المفاتيح ..ينطلق.

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Events

والسؤال ..هل يوجد المزيد..نعم يوجد ..لكن هذه أهم ال event الموجودة والمدعومة من جميع المتصفحات الرئيسية ..والآن ..

عليك أن تقوم بتطبيق مثال على كل خاصبة ..وهذا أمر مهم .. وقبل أن تنطلق للشريحة التالية.. لأنه لا يمكنك تجاهل هذا الموضوع المهم...

ستقول ..كيف سأطبق ولم أشاهد أمثلة عليها ؟!. . سأقول لك أنت مبرمج محترف الآن.. لديك الصيغة العامة ..وتستطيع كتابة أي سكربت بسيط .. ولديل ال events ..كل ما تبقى عليك هو التجربية ^_^

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Events

هل قمت بتنفيذ الأسئلة السابقة ؟!

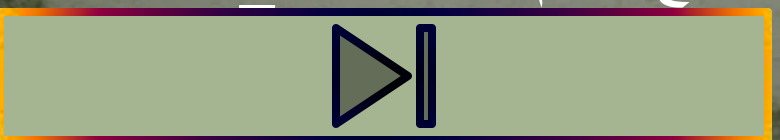
اذا قمت بذلك ..وحدثت معك أخطاء ..أو تريد ان ترى جميع هذه ال events تعال معي لنرى هذا المثال ^_^

ملاحظة: بعض الخصائص الجديدة في المثال:

Placeholder: خاصية يمكن استخدامها لعمل watermark ..(شاهدها بالمثال)

لتغيير الخصائص ل css عن طريق الجافا سكربت يمكننا استخدام العنصر الي بدنا نغيرلو الخصائص ثم style ثم background ثم = ثم القيمة

..^_^والآن لنفرح معا بالمثال ..يجب أن تستطيع القيام به ..سهل ^_^



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Events

والآن ..أخر نقطة حول هذا الموضوع وهي ..هل كتابة ال event تكون دائما داخل ال html ..ونستدعي من خلاله الجافا سكربت ؟...

الجواب بكل تأكيد ..لا ^_^..يمكنك كتابة السكربت وال event داخل ال script tag ...

شاهد الصيغة العامة:

```
object.EventName=function(){myScript};
```

مثال على الصيغة العامة:

```
document.getElementById("btnMouseMove").onmouseover = function() {myFunctionOver()};
```

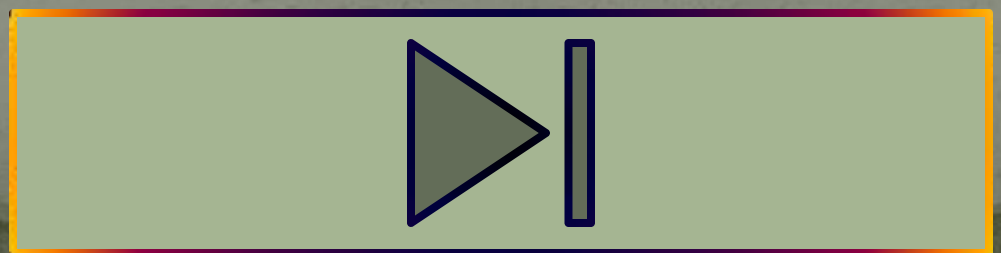
aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Events

شاهد المثال التالي:

```
<input type="button" id="btnMouseMove" title="Click Event" value="Click Event" />
<script>
    // Declare Event
    document.getElementById("btnMouseMove").onmouseover = function() {myFunctionOver()};
    document.getElementById("btnMouseMove").onmouseout = function() {myFunctionOut()};

    function myFunctionOver() {
        document.getElementById("btnMouseMove").style.backgroundColor = "red";
        document.getElementById("btnMouseMove").style.color = "blue";
    }
    function myFunctionOut() {
        document.getElementById("btnMouseMove").style.backgroundColor = "green";
        document.getElementById("btnMouseMove").style.color = "red";
    }
</script>
```



تفكر ...

أعوذ بالله من الشيطان الرجيم

"اللَّهُ لَا إِلَٰهَ إِلَّا هُوَ الْحَيُّ الْقَيُّومُ لَا تَأْخُذُهُ سِنَةٌ وَلَا نَوْمٌ لَهُ مَا فِي السَّمَاوَاتِ وَمَا فِي الْأَرْضِ مَنْ ذَا الَّذِي يَشْفَعُ عِنْدَهُ إِلَّا بِإِذْنِهِ يَعْلَمُ مَا بَيْنَ أَيْدِيهِمْ وَمَا خَلْفَهُمْ وَلَا يُحِيطُونَ بِشَيْءٍ مِنْ عِلْمِهِ إِلَّا بِمَا شَاءَ وَسِعَ كُرْسِيُّهُ السَّمَاوَاتِ وَالْأَرْضَ وَلَا يَئُودُهُ حِفْظُهُمَا وَهُوَ الْعَلِيُّ الْعَظِيمُ"

آية الكرسى (البقرة ٢٥٥)

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

الآن ..سنعود مجددا للحديث عن الأنواع التي ذكرناها من قبل ..لكن بشيء فيه تفصيل نوعا ما ...بالإضافة ذكر بعض الدوال التي يمكن استخدامها مع كل نوع من أنواع البيانات ..

أول هذه الأنواع هو ال String (النصي) ويستخدم هذا النوع لحفظ مجموعة من الحروف أو الأرقام أو الرموز الخاصة والتعديل عليها ...وقد شرحنا ذلك سابقا في أنواع البيانات ..

والآن سننطلق معا بإذن الله تعالى ... الى عالم جديد ..يملأه المغامرة والأفكار

*_^

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

١) يمكن كتابة النصوص(String) داخل (‘) quote أو double quote(“) مثل “anees” أو ‘anees’ ...

٢) يمكن كتابة ال ‘ داخل ال ” ويمكن كتابة ” داخل ال ‘ ... شاهد المثال:
“‘anees ‘test’” أو ‘ ”anees “test”‘....

٣) يمكننا استخدام ال backslash (\)..لكتابة ‘ داخل “ أو ” داخل ‘ ...شاهد المثال: “ ’\hikmat\’ anees” أو ‘ ”\hikmat\” anees’ أو حتى يمكنك كتابة “ \” hikmat \” anees”

قد تتعجب ..لماذا ذكرت هذه النقاط؟..الجواب بكل بساطة ..عشان ما يفقع معاك الكود ^_* ..مصطلح ينحب استخدامه ...*_*...أريدك أن تشاهد المثال بالصفحة التالية لتعلم لماذا تطرقنا الى مثل هذه المواضيع

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

لاحظ شكل الشيفرة البرمجية بالصورة..

```
alert('is'nt true');
alert("Anees Like "javascript"");
</script>
```

لاحظ الخطأ .. هذه الشيفرة البرمجية لن تعمل..

حسب النقاط التي ذكرناها ..ما رأيك أن تقوم بحل هذا الشكل ليصبح صحيحا ب ٣ طرق ^_^..(الطرق تم ذكرها بالشريحة السابقة)

*) الرموز الخاصة وال backslash:

لقد لاحظت في الشريحة السابقة كيف أن ال (\) backslash كان لها دور كبير في حل مشكلة ال ' أو " ... ال ' وال " هذه تسمى رموز خاصة ..الآن يوجد أيضا العديد من الأمور التي يمكن استخدام ال backslash لها ..شاهد الجدول بالشريحة التالية ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript String

Code	Outputs
\'	single quote
\"	double quote
\\	backslash
\n	new line
\r	carriage return
\t	tab
\b	backspace
\f	form feed

كما تلاحظ .. فإنك تشاهد مجموعة من الخصائص التي يمكن صنعها باستخدام ال backslash
..شاهد مثلا كيف يمكننا طباعة سطر جديد لنص ..\n... هذه مهمة..جربها داخل alert وانظر النتائج..وجرب ال
 وشوف أيتهما التي ستعمل ...^_^

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript String

والسؤال القوي ...هل ينتهي ال string هنا .. الجواب طبعا لا ..فنحن لم نبدأ بعد ..ما ذكرناه أمور تخص نفس النص ..والآن نريد أن نصل الى خصائص أو معلومات حول هذا النص مثل معرفة عدد الحروف..تغيير أو استبدال كلمة معينة ...الخ ..

*) length: هذه الخاصية تعد من خصائص ال string وهي تسمح لنا بمعرفة طول نص معين ..من بدايته حتى نهايته ... مثلا anees اذا استخدمت هذه الخاصية فالناتج هو ٥ ..شاهد المثال:

```
<script>
    var x = "anees hikmat";
    alert(x.length);
</script>
```

الآن في هذا المثال الناتج سيكون هو ١٢ ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

الآن سأبدء من هنا بالتحدث عن الدوال (method) التي سنقوم باستخدامها مع ال string بإذن الله تعالى..

١) charAt()
٢) charCodeAt()
٣) concat()
٤) fromCharCode()
٥) indexOf()
٦) lastIndexOf()
٧) replace()
٨) search()

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

٩) slice()
١٠) split()
١١) substr()
١٢) substring()
١٣) toLowerCase()
١٤) toUpperCase()
١٥) trim()

هذه الدوال ..حسب اعتقادي أهم الدوال الموجودة ..وهناك غيرها بكل تأكيد ويمكنك البحث عنها .. هذه الدوال قد تكون هي أكثر الدوال المستخدمة أثناء المشاريع ..ولذلك فهي ذي الأهمية بمكان ... والآن لنبدأ على بركة الله ..

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

١) charAt(): تستخدم هذه الدالة لارجاع حرف معين ..أو char معين موجود داخل هذا النص بنائا على موقعه داخل النص مثل:

```
var x = "anees hikmat";
var charAtVar = x.charAt(4);// string.charAt(Integer Number);
console.log("x.charAt(4) ex1: " + charAtVar); // The result is ( s )
```

لاحظ ..أن ال charAt لا يقبل الى أعداد صحيحة ..مثل ٠ ..١..٢ الخ وهو اجباري.. والملاحظة الثانية أن العد يبدأ من صفر ..وهذا يعني ..لو أردت طباعة الحرف الأخير باستخدام ال length ..فسيكون الموقع هو length – ١ ..

شاهد هذا المثال: بعد ذلك قم بتطبيق المثال الأول والثاني (ضروري) ^_^ *

```
charAtVar = x.charAt(x.length - 1);
console.log("x.charAt(4) ex2: " + charAtVar); // The result is ( t )
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

٢)charCodeAt() : في دورة ال html ..إن كنت تذكر فقد تطرقنا الى موضوع ال Charest وتكلمنا عن ال Unicode ..وغيرها ..وقلنا أن لكل char حسب نظام ال Unicode الذي يستخدمه رقم يمثله ..الآن بإمكاننا معرفة رقم ال Unicode لهذا ال char... فمثلا ال A رقمها هو ٦٥ بالجدول ورقم ال a هو ٩٧ .. ^_^ *

شاهد هذا المثال:

```
charAtVar = x.charCodeAt(0);
console.log("x.charCodeAt(0) ex3: " + charAtVar); // The result is ( 97 )
```

أيضا هنا ..تستقبل هذه الدالة رقم صحيح فقط، وهو الزامي، ويبدأ العد من صفر وصولا الى ال length – 1 .. ^_^ *.. الآن قم بتطبيق المثال هذا ..

أيضا أريدك أن تقوم بطباعة ال Unicode للحرف قبل الأخير دون كتابة رقم ثابت لهذا العنصر..يعني اذا جائت الكلمة ٥ أو ١٠ أو ٢٠ حرف..بطبع ال Unicode للحرف القبل الأخير مباشرة ^_^ *

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript String

٣) ()concat: تستخدم هذه الدالة للدمج بين النصوص ..بحيث يمكن الدمج بين نصين أو أكثر باستخدام هذه الدالة ...

الصيغة العامة هي:

```
(stringNameVariable.concat (str1, str2, str3, str4)
```

ال str يرمز الى متغير من نوع string وال ..StringNameVar ترمز الى اسم المتغير الذي نريد ضم النصوص الموجودة داخل ال concat اليه..لنشاهد المثال معا ثم قم بتنفيذ المثال ..(عدد المتغيرات من ١ الى x)

```
var str1 = "anees ", str2 = "hikmat ", str3 = "anees ", str4 = "abu-hmaid"
    , result = "str1.concat(str2, str3, str4): ";

console.log(result.concat(str1, str2, str3, str4));
cript>
```

النتج

```
result.concat(str1, str2, str3, str4): anees hikmat anees abu-hmaid
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

٤)fromCharCode() : هذه الدالة تقوم بعملية عكسية لل charCodeAt
بحيث تقوم بأخذ رقم ال Unicode وتحويله الى ال char..^_^*
الصيغة العامة (طبعا عدد الأرقام ألمراد تحويلها من ١ الى n):

```
String.fromCharCode(unicodeVar1, unicodeVar2, unicodeVar3, unicodeVar4)
```

شاهد الأمثلة: (وقم بتطبيقها).

```
var unicodeVar1 = 65, unicodeVar2 = 67, unicodeVar3 = 78, unicodeVar4 = 50;
console.log(String.fromCharCode(unicodeVar1, unicodeVar2, unicodeVar3, unicodeVar4));
```

اذا استعطت معرفة الناتج ... فأنت في قمة الروعة ...الآن ^_^*

```
x = "anees hikmat";
result = "Concat & charCodeAt & fromCharCode & length in one example EX 7: ";
console.log(result.concat(String.fromCharCode(x.charCodeAt(x.length - 2),
        x.charCodeAt(1),x.charCodeAt(2), x.charCodeAt(3),x.charCodeAt(4))));
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

٥) indexOf(): هذه الدالة الرائعة ..تخبرنا بموقع أول نص تم العثور عليه ضمن سلسلة الحروف..بمعنى آخر لوقمت باستخدام هذه الدالة مع حرف ال e في كلمة anees ...فإنه سيرجع موقع أول حرف e بلاقيه ..وهنا ٢

الآن ..لنرى الصيغة العامة:

```
// indexOf Example
console.log(x.indexOf("a", 3));
</script>
```

يبدأ البحث من الموقع الثالث..وهذا اختياري والحالة الافتراضية هي 0

اللي بدنا نبحث عنه

النص الي بنبحث فيه

اذا أرجعت الدالة ـ١ .فهذا يعني أن العنصر المراد البحث عنه ..غير موجود

..أيضا هناك ملاحظة مهمة..أن هذه الدالة حساسة لحالة الأحرف بخصوص البحث ..فالبحث عن A لا يساوي البحث عن a ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

٦)lastIndexOf(): هذه الدالة عكس السابقة، فهي **ترجع آخر عنصر** مراد البحث عنه ..بدلا من أول عنصر...ويبدأ البحث في هذه الدالة من آخر حرف موجود رجوعا الى أول حرف..ويرجع الموقع الفعلي لهذا الحرف في حال وجد ..مثل أريد أن أبحث عن حرف e في كلمة anees ..باستخدام هذه الدالة ..فيكون الناتج هو ٣..

```
// lastIndexOf Example
x = "anees hikmat anees abu-hmaid";
console.log("x.lastIndexOf('a') Ex 9: " + x.lastIndexOf("a")); // result is 25
console.log("x.lastIndexOf('a', 24) Ex 10: " + x.lastIndexOf("a", 24)); // result is 19
```

يبدأ العد من آخر موقع

هنا يبدأ العد من الموقع 24 رجوعا الى الصفر...

ما ينطبق على indexOf ينطبق على هذه الدالة ..فهي حساسة لحالة الأحرف..والقيمة صفر تدل على عدم وجود قيمة ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript String

٧) ()replace: هذه الدالة الجميلة ..تقوم على البحث عن نص معين موجود داخل string...ومن ثم استبداله بنص جديد، النص المراد البحث عنه وتبديله إما أن يكتب مباشرة، وإما أن يكتب بصيغة RegExp|.. هذا الموضوع سنتطرق اليه الحقا ان شاء الله ..والآن سنكتفي بالنص المباشر..

الصيغة العامة:

```
x.replace("anees", "taher")
```

X: هو ال string الذي سنبحث فيه عن كلمة anees ونقوم باستبدالها بكلمة taher ... مكان كلمة anees يمكنك وضع regExp ..لكن لن أتطرق لهذا الموضوع... شاهد المثال التالي:

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript String

```
x = "anees hikmat anees abu-hmaid";
console.log("x.replace(\"anees\", \"taher\") ex11: " + x.replace("anees", "taher"));
```

```
x.replace("anees", "taher") ex11: taher hikmat anees abu-hmaid
```

ناتج تطبيق المثال

هل لاحظت الناتج ..نعم ..فقط أول كلمة من ال x هي التي تغيرت فقط.. ولكي نستطيع تغيير كامل الكلمات داخل النص يمكننا اضاف / قبل وبعد النص وبدون وضع ” مع اضافة حرف ال g ..شاهد المثال الثاني:

```
console.log("x.replace(/anees/g, \"taher\") ex11: " + x.replace(/anees/g, "taher"));
```

```
x.replace(/anees/g, "taher") ex11: taher hikmat taher abu-hmaid
```

هل لاحظت كيف هو شكل البحث الآن ..اللون الأصفر الآن يمثل regExp بسيط وال g وهناك ال i هي flag ..وتعني ال g كل العناصر..وال i تجاهل حالة الأحرف... (طبق الأمثلة ^_*)

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

٨) search(): هذه الدالة تستخدم للبحث عن نص معين داخل ال string ويقوم على ارجاع موقع النص هذا ... ويرجع -١ في حال لم يكن النص موجودا .. يكون هنا البحث عن طريق string أو regExp ..وفعليا في هذه الدالة ..يتم تحويل النص الى regExp...

الصيغة العامة:

```
x.search("Anees")
```

ال x ترمز الى النص المراد البحث فيه عن كلمة Anees... في هذه الحالة يجب أن يكون متطابق، ويمكننا استخدام ال i لتجاهل حالة الأحرف.. لكن عملية جلب الموقع ..تكون لأول قيمة تم ايجادها ...

```
console.log("x.search(\"Anees\") ex14: " + x.search("Anees"));
console.log("x.search(/Anees/gi) ex15: " + x.search(/Anees/gi));
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

٩) slice(): هذه الدالة الجميلة تقوم على أخذ جزء من النص ..وارجاعه ..

الصيغة العامة:

في الصيغة العامة ..البداية الزامية والنهاية اختيارية ..في حال أنك اكتفيت بالدباية فإنه سيقوم بأخذ الجزء من البداية التي قمت بتحديدها ..وصولا لنهاية النص ...لنشاهد أمثلة على ذلك ..

```
// Slice Example
x = "anees hikmat anees Anees abu-hmaid";
console.log("x.slice(0, 5) ex16: "    + x.slice(0 , 5));
console.log("x.slice(5) ex17: "       + x.slice(5));
console.log("x.slice(-1) ex18: "      + x.slice(-1));
console.log("x.slice(19, -10) ex19: " + x.slice(19, -10));
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

```
x.slice(0, 5) ex16: anees
x.slice(5) ex17:  hikmat anees Anees abu-hmaid
x.slice(-1) ex18: d
x.slice(19, -10) ex19: Anees
```

شرح المثال مع النتائج:

لاحظ المثال رقم ١٦: الناتج كان anees والسبب أنني قلت له ..قم بأخذ من الموقع ٠ بطول ٥ فكان هذا الناتج

المثال ١٧: حددت البداية من الموقع الخامس فكان الناتج من الموقع الخامس وحتى نهاية النص (لإنه لم تحدد النهاية)

المثال ١٨: قلت له -١ وهذا يعني قم بالبدأ من العنصر الآخير.. ولم أحدد نهاية فطبع الحرف d وهو أصلا آخر حرف...

المثال ١٩: قلت له قم بالقطع من ١٩ الى -١٠ وهذا يعني خذ النص من الموقع ال ١٩ والنهاية تكون قبل نهاية النص بعشرة أحرف..بمعتى آخر قمنا باستثناء آخر عشرة حروق وأول ١٩ حرف ...وأخذنا الي بيناتهم ^_*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript String

١٠) ()split: تستخدم هذه الدالة على تقسيم النص الى أجزاء بحيث تصبح مصفوفة من الأجزاء بدلا من أن تكون مصفوفة من الحروف..

ليتضح الفرق بشكل بسيط ...شاهد هذا المثال:

```
x = "anees hikmat anees Anees abu-hmaid";
console.log("x[1] ex21: "    + x[1]);
console.log("x.split(' ')->[1] ex22: "    +  x.split(" "));
```

ناتج المثال ٢١: هو حرف ال n.. أما ناتج المثال ٢٢ سيكون hikmat..

الآن نعود ..الصيغة العامة لهذه الدالة:

```
stringName.split(separator,limit);
cript>
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

يقصد بال separator الجزء الذي سنعتبره فاصلا لكي يقسم هذا النص الى مصفوفة بنائا عليه .. فمثلا يمكن أن يكون حرف معين..أو رقم معين ..أو regExp أو مسافة أو حتى "" ..بدون أي شيء وبهذه الحالة يتم فصل الحروف ..

أما ال limit فهو يحدد كم عنصر على الأكثر يمكن أن يتحول الى مصفوفة فيمكنني أن أحدد ٥ عناصر فقط ..الخ..

هذان الخياران اختياريان...شاهد هذا المثال(أرجو أن تقوم بالتطبيق مهم جدا):

```
x = "anees hikmat anees Anees abu-hmaid";
console.log("x.split(' ') ex20: "    + x.split(" "));
console.log("x[1] ex21: "    + x[1]);
console.log("x.split(' ')[1] ex22: "    + x.split(" ")[1]);
console.log("x.split('') ex23: "    + x.split(""));
console.log("x.split('', 3) ex23: "    + x.split("", 3));
console.log("x.split('a') ex24: "    + x.split("a"));
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript String

نتائج تطبيق المثال السابق.. اذا لم تقم بحل أو تتبع نتائج المثال السابق ..عد قبل أن تقرء هذه النتائج ..

```
x.split(' ') ex20: anees,hikmat,anees,Anees,abu-hmaid
x[1] ex21: n
x.split(' ')[1] ex22: hikmat
x.split('') ex23: a,n,e,e,s, ,h,i,k,m,a,t, ,a,n,e,e,s, ,A,n,e,e,s, ,a,b,u,-,h,m,a,i,d
x.split('', 3) ex23: a,n,e
x.split('a') ex24: ,nees hikm,t ,nees Anees ,bu-hm,id
```

هذه هي نتائج الأمثلة السابقة.. تأكد منها بدقة ..

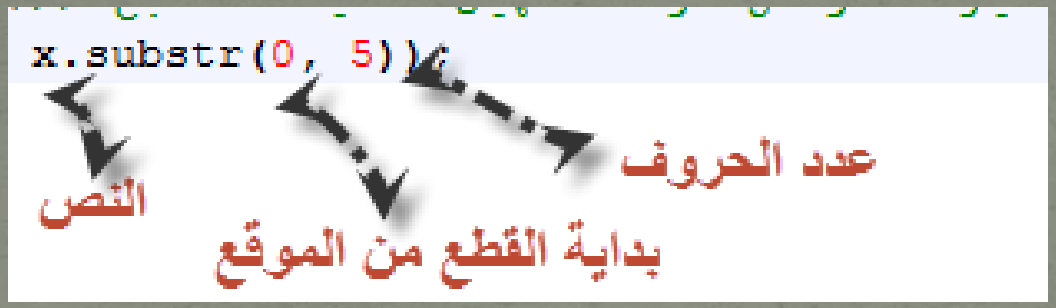
*_^ أووووووووووووه .....هانت يا شباب ..وصلنا الدالة رقم ١١ ^_^

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

١١) substr(): هذه الدالة مهمة جدا .. تقوم هذه الدالة على اقتطاع جزء من النص ..وهي تشبه ال slice ..لكن الاختلاف في هذه الدالة ..بأنك تقوم بتحديد عدد الحروف المراد اقتطاعها ..ونقطة البداية ..ونقطة البداية ممكن أن تكون موجبة.. ويمكن أن تكون سالبة ... لاكن القيمة الثانية وهي عدد الحروف المراد اقتطاعها ..فهي موجبة اجباريا..

الصيغة العامة:



في حال كانت بداية القطع ..أكبر من عدد الحروف .. فإن الناتج هو Null ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript String

شاهد المثال:

```
x = "anees hikmat anees Anees abu-hmaid";
console.log("x.substr(0, 5) ex25: "    + x.substr(0, 5));
console.log("x.substr(6, 6) ex26: "    + x.substr(6, 6));
console.log("x.substr(-5, 3) ex27: "   + x.substr(-5, 3));
console.log("x.substr(5, 0) ex28: "    + x.substr(5, 0));
console.log("x.substr(180, 10) ex29: "   + x.substr(180, 10));
```

Null

Null

النتائج(تتبع المثال..وحاول توقع النتائج):

```
x.substr(0, 5) ex25: anees
x.substr(6, 6) ex26: hikmat
x.substr(-5, 3) ex27: hma
x.substr(5, 0) ex28:
x.substr(180, 10) ex29:
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

١٢)substring(): هذه الدالة صورة مصغرة عن ال slice ..بحيث تقوم باقتطاع جزء معين من النص بنائا على البداية والنهاية التي تم تحديدها ..والنهاية التي توضع لا تكون ضمن القطع ..والفرق بين هذه الدالة وال slice هو أنه لا يمكنك استخدام قيم سالبة هنا .. بالإضافة الى أنه يوجد خاصية في هذه الدالة swapping بحيث لو قمت بوضع الرقم الأول أكبر من الثاني ..فإنه سيتم تبديل الرقمين ... والقيمة السالبة في هذه الدالة يتم تحويلها الى صفر...

الصيغة العامة:

```
x.substring(0, 5));
```

المتغير x هو string ... وال o هي البداية وال 5 هي آخر موقع ..هذا الموقع لن يتم اقتطاعه ...وسيتم قطع ٠ و ١ و ٢ و ٣و ٤ ...بدون الموقع ٥

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript String

شاهد المثال:

```
// substring Example
x = "anees hikmat anees Anees abu-hmaid";
console.log("x.substring(1, 2) ex30: "    + x.substring(1, 2));
console.log("x.substring(5, 0) ex31: "    + x.substring(5, 0));
console.log("x.substring(0, 5) ex32: "    + x.substring(0, 5));
console.log("x.substring(3, 8) ex33: "    + x.substring(3, 8));
console.log("x.substring(-5, 5) ex34: "    + x.substring(-5 , 5));
```

لاحظ النتائج بدقة: (وكل مثال موجود أمامك ..قم باستبداله بالدالة slice وقارن بين النتائج ...الاختلافات التي ذكرتها بالشريحة السابقة مهمة ^__*):

```
x.substring(1, 2) ex30: n
x.substring(5, 0) ex31: anees
x.substring(0, 5) ex32: anees
x.substring(3, 8) ex33: es hi
x.substring(-5, 5) ex34: anees
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

١٣)() toLowerCase : هذه الدالة البسيطة والرائعة ..مهمتها بكل بساطة هو تحويل حالة الأحرف الى lower case ... ^_*

الاستخدام:

```
x.toLowerCase()
```

يرمز ال x الى النص ...

شاهد المثال:

```
// toLowerCase Example
x = "ANeeS HIKMAT";
console.log("x{'ANeeS HIKMAT'}.toLowerCase() ex35: "    + x.toLowerCase());
```

```
x{'ANeeS HIKMAT'}.toLowerCase() ex35: anees hikmat
```

ناتج

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript String

١٤)() toUpperCase : هذه الدالة عكس الدالة السابقة ..فهي تقوم على تحويل حالة الأحرف الى Upper Case.. ^____^*

الاستخدام:

```
// toUpperCase Example
x = "anees Hikmat";
console.log("x{'anees Hikmat'}.toUpperCase() ex36: "    + x.toUpperCase());
```

شاهد النتائج:

```
x{'anees Hikmat'}.toUpperCase() ex36: ANEES HIKMAT
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript String

١٥)trim() : والآن ..آخر خاصية والحمد لله ^_* في موضوعنا عن ال string ..وهي خاصية ال trim ^_*

هذه الدالة تقوم بحذف الفراغات أو المسافات قبل وبعد النص ...

الاستخدام: `x.trim()`

مثال:

```
// trim Example
x = "                    anees Hikmat                    ";
console.log("x without trim ex37: "    + x);
console.log("x.trim() ex38: "    + x.trim());
```

نتائج

```
x without trim ex37:                    anees Hikmat                    
x.trim() ex37: anees Hikmat
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

والآن.. بحمد الله تعالى ..نكون انتهينا من الحديث عن أهم الدوال الخاصة بالنصوص داخل الجافا سكربت .. والآن ما رأيك أن نرى ..هل درست جيدا المواضيع السابقة أم لا ..

انظر الشكل:

Anees Hikmat Anees Abu-hmaid

A,n,e,e,s, ,H,i,k,m,a,t, ,A,n,e,e,s, ,A,b,u,-,h,m,a,i,d

هل يمكنك باستخدام الجافا سكربت ..عمل مثل هذا الشكل للنص التالي

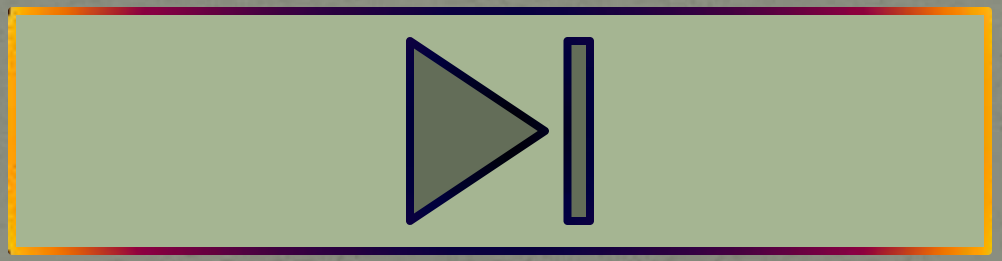
Anees Hikmat Anees Abu-hmaid ..بحيث تستخدم الدوال التي تحدثنا عنها ..مساعدة للحل: الأولى استخدمت replace والثانية split

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript String

هل قمت بحل السؤال ..^_*.. فكرة الحل بسيطة جدا ..وبسطر واحد جافا سكربت ..^_* لكل شكل... هل خطر ببالك استخدام ال html مثل ال mark tag ^_*..

شاهد الحل ..مع أمثلة على جميع الدوال السابقة ..تتبع الحل والنتائج والشيفرة البرمجية ... ^_*



تفكر ...

قال صل الله عليه وسلم : الشهداء الذين يقاتلون في سبيل الله في الصف الأول ولا يلتفتون بوجوههم حتى يقتلوا فأولئك يلقون في الغرف العلا من الجنة يضحك إليهم ربك، إن الله تعالى إذا ضحك إلى عبده المؤمن فلا حساب عليه.

صححه الشيخ الألباني فى صحيح الجامع.

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Number

تحدثنا عن الأرقام في موضوع أنواع البيانات وال syntax ^_*..والآن لنستزيد من الشعر بيت ...ولنتابع التقدم ^_*

الآن أول معلومة على الريق ^_*.. الجافا سكربت بتعتمد نظام ال 64bit للبيانات من نوع رقم .. وهذا الأمر معتمد من ال IEEE ..وبهذا في تختلف عن لغات البرمجة الأخرى من حيث الحاجة لتعريف متغيرات مثل integer (عدد صحيح) أو float (عدد عشري) أو long (عدد صحيح كبير) ...الخ

الجافا سكربت ..تعطيك نوعا من الراحة في التعامل..بحيث يمكنك فقط تعريف متغير ووضع الرقم بداخله ..

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Number

١) أول ما سنتحدث عنه ..هو القسمة على صفر... في الرياضيات هل يمكنك القسمة على صفر؟ ...الجواب بالطبع لا ..اذا اذا واجهت هذه المشكلة البرنامج الذي أنت به ..ماذا سيكون الحل بوجهة نظرك ؟

لكل لغة برمجة أسلوب خاص للتعامل مع هذا الخطأ، بالإضافة الى ذلك هناك أساليب يتبعها المبرمجون لتلافي هذا الخطأ ..

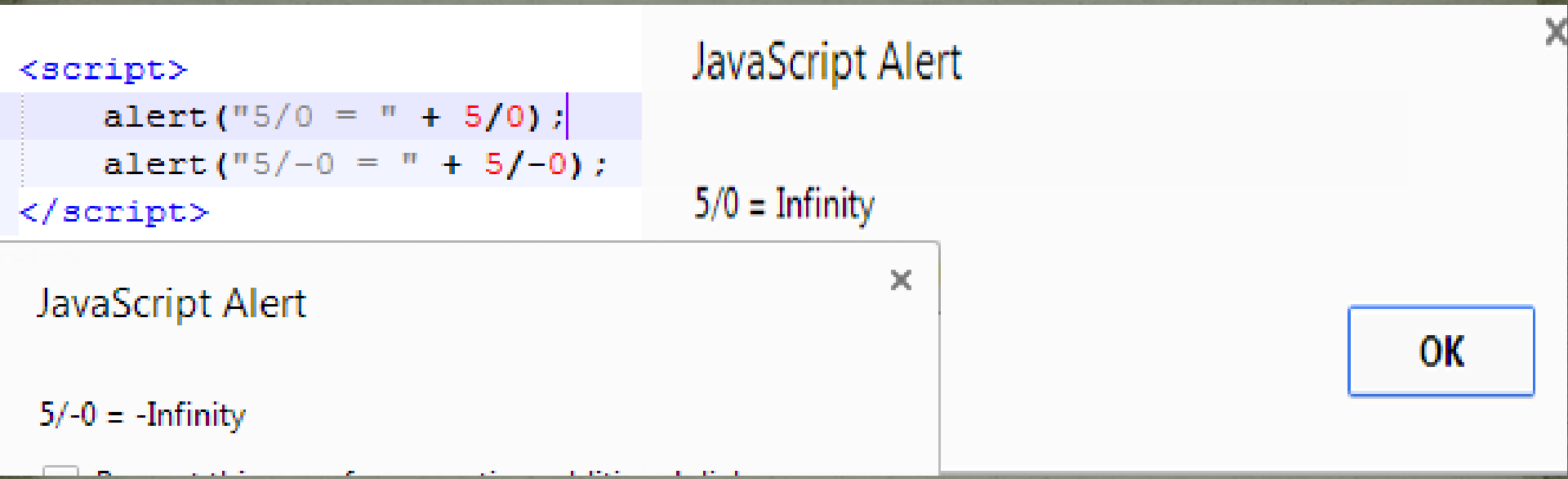
والآن وما يهمنا هنا هو تعامل الجافا سكربت..^_^.، هناك قيمة مخزنة داخل الجافا سكربت تسمى Infinity هذه القيمة هي التي ستنتج في حال القسمة على صفر... واذا قمت بالقسمة على سالب صفر فإن الناتج سالب Infinity....

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Number

والآن لنشاهد المثال معا ^_^:

```
<script>
    alert("5/0 = " + 5/0);
    alert("5/-0 = " + 5/-0);
</script>
```

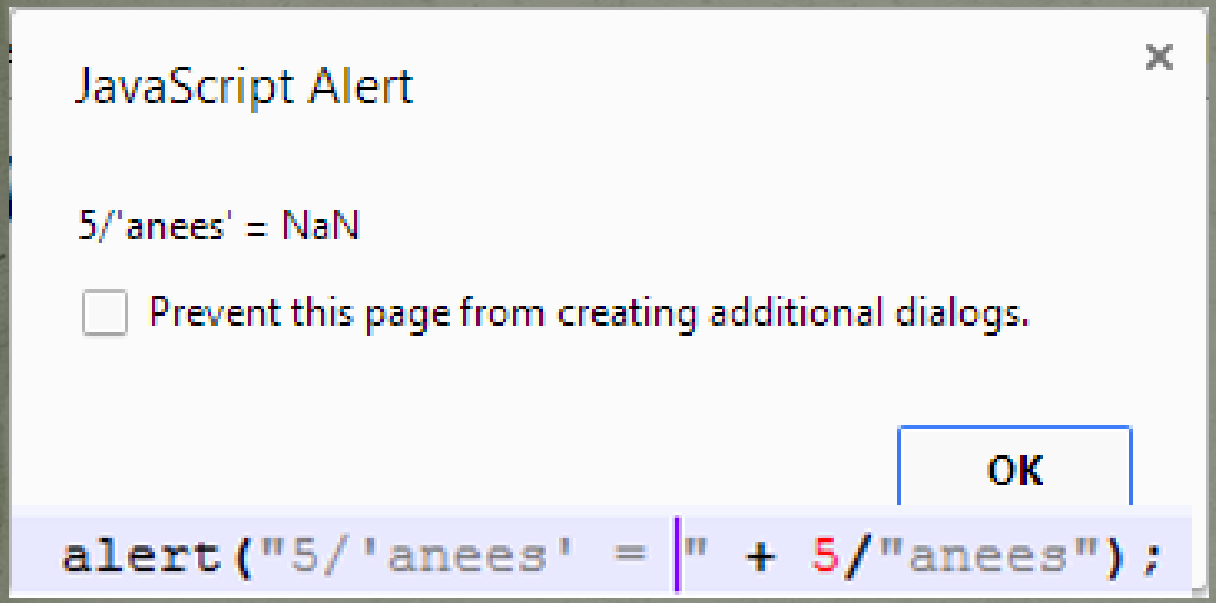


٢) NaN: هذه كلمة يتم استرجاعها من قبل الجافا سكربت لتدل على القيم / القيمة الموجودة ليست برقم.. Not a Number.. مثل القسمة على حرف .. "anees"/54 ... الناتج هو NaN...

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Number

شاهد مثالا على NaN:



ملاحظة: لو قمت بتجربة استخدام typeof مع ال NaN وال infinity..
سيكون الناتج ...(---------أنت ضع الناتج----------)..ملاحظة النوع لن يكون نص ...لذلك ..طبق وانظر النوع ..*_^

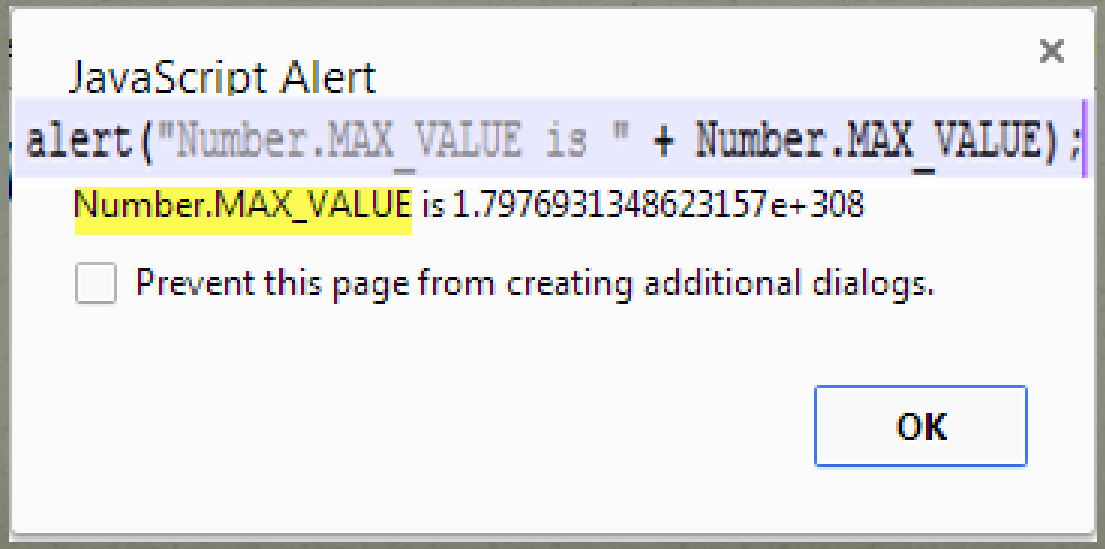
aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Number

الآن..هل هناك حد أعلى للرقم الذي يمكن ادخاله ؟!..وهل هناك حد أدنى ؟

الجواب ببساطة ..نعم ..هناك حد أعلى وحد أدنى لكل رقم ..ولمعرفة ذلك يمكنك استخدام بعض من خصائص ال Number كنوع مثل ال MAX_VALUE ...

شاهد المثال:



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Number

أيضا هناك ال Number.MIN_VALUE.. جربها بنفسك ^_*

ملاحظة: هناك ما يسمى بال global method وال method ..عندما نتكلم عن دالة على أنها global method فهذا يعني أنه يمكن تطبيقها على جميع أنواع البيانات ...والآن بحسب طبيعة عملنا مع ال Number فإنني سأذكر ٣ دوال مهمة وهي

١) parseInt: هذه الدالة تقوم على تحويل النص الى رقم صحيح، والذي يسترجع هو أول رقم والمسافات هي المسموحة _*
٢) parseFloat: هذه الدالة تقوم على تحويل النصر الى رقم عشري
٣) Number: تستخدم هذه الدالة لتحويل متغير الجافا سكربت الى رقم

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Number

```
Number(true)1
Number(false)0
Number( 10)10
Number(10 )10
Number(10 5)NaN
parseInt(10)10
parseInt(10.33)10
parseInt(10 6)10
parseInt(10 years)10
parseInt(years 10)NaN
parseFloat(10)10
parseFloat(10.33)10.33
parseFloat(10 6)10
parseFloat(10 years)10
parseFloat(years 10)NaN
```

لاحظ مجموعة النتائج لكل دالة
من الدوال الثلاث..

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Number

والآن سنتكلم بإذن الله تعالى عن بعض الدوال التي يمكن استخدامها مع ال Number type..

١) toString: تقوم هذه الدالة على ارجاع الرقم على شكل نص..

٢) toExponential: تستخدم هذه الدالة لتحويل الرقم الى شكل أسي بعد تقريبه، يتم ارجاع هذا الرقم على شكل نصي(string)...

٣) toFixed(): هذه الدالة تقوم على تحديد عدد الخانات العشرية التي أريدها أن تبقى.. وهذه مفيدة جدا في التعامل مع الأمور النقدية والتي بها أعشار... ^_*..الناتج يعود على شكل نصي أيضا..

٤) toPrecision(): هذه الدالة تقوم على تحديد طول الرقم الذي سيظهر مع تقريب..وترجع الناتج على شكل نصي....الآن لنشاهد أمثلة على الموضوع ^_*

*) ملاحظة التغيير بالستخدام الدوال التي ذكرنها والخاصة بالأرقام تكون على مستوى الناتج(قيمة مرجعة جديدة) ولا يغير هذا على القيمة المتغير الأصلي...

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Number

```
var num = 57.9457;
document.getElementById("data-method").innerHTML =
    "<br />num.toString() " + num.toString() + " <b st
    "num.toExponential(2) " + num.toExponential(2) + "
    "num.toFixed(2) " + num.toFixed(2) + " <b style='c
    "num.toFixed(1) " + num.toFixed(1) + " <b style='c
    "num.toPrecision(2) " + num.toPrecision(2) + " <b
    "num.toPrecision() " + num.toPrecision() + " <b st
cript>
```

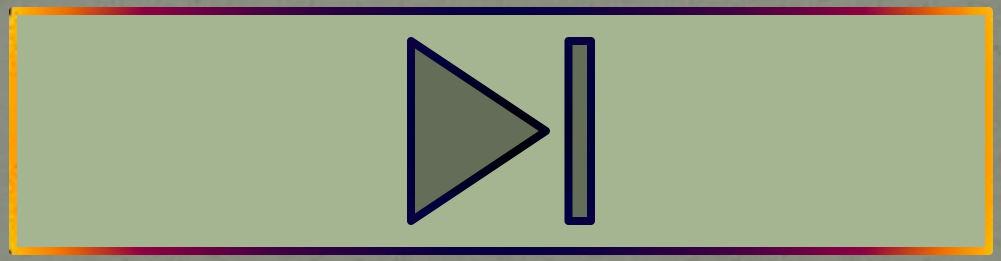
num.toString() 57.9457 **typeof: string**
num.toExponential(2) 5.79e+1 **typeof: string**
num.toFixed(2) 57.95 **typeof: string**
num.toFixed(1) 57.9 **typeof: string**
num.toPrecision(2) 58 **typeof: string**
num.toPrecision() 57.9457 **typeof: string**

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Number

والآن ..لنشاهد مثالا عن جميع المواضيع التي تكلمنا عنها بإذن الله تعالى ...

(أرجو أن تكون قد طبقت الأمثلة السابقة..وحاول الآن تغيير الأرقام والقيم ..وتوقع الناتج..)



تفكر ...

العلم يبني بيوتا لا عماد لها ..
والجهل يهدم بيت العز والكرم

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Operators

ال Operators هي مجموعة العمليات التي يمكن القيام بها بين المتغيرات أو القيم المختلفة ولإداء وظائف مختلفة، هذه العمليات تختلف بطبيعتها بحسب طبيعة المتغيرات أو القيم المراد التعامل معها ..أو الهدف منها ..

أنواع ال Operator:

١) Arithmetic Operator
٢) Assignment Operator
٣) String Operator
٤) Bitwise Operator
٥) Unary Operator
٦) Comparison Operator
٧) Logical Operator
٨) Conditional Operator

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Operators

والآن لنبدء على بركة الله تعالى ..بشرح أنواع ال Operator ^_*

١) Arithmetic Operator: وهي العمليات الرياضية والتي يمكن اجرائها بوساطة الجافا سكربت..وتكون هذه العمليات بين متغيرات أو قيم..

العمليات التي يمكن استخدامها في هذا النوع هي:

+ و – و * و / و % و ++ و –

والآن لنرى معا أمثلة وشرح لكل من هذه العمليات ^_*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Operators

```
var x = 10, y = 6, c;
document.getElementById("arthOperator").innerHTML =
"x + y = <b>" + (x + y) + "</b><br />" +
"x - y = <b>" + (x - y) + "</b><br />" +
"x * y = <b>" + (x * y) + "</b><br />" +
"x / y = <b>" + (x / y) + "</b><br />" +
"x % y = <b>" + (x % y) + "</b><br />" +
"x++ = <b>" + (x++) + "</b><br />" +
"++x = <b>" + (++x) + "</b><br />" +
"x-- = <b>" + (x--) + "</b><br />" +
"--x = <b>" + (--x) + "</b><br />";
</script>
```

جمع
طرح
قسمة
ضرب
باقي القسمة

هذه ++ أو ال --
تستخدم لإضافة 1 أو طرح 1
لكن ما الفرق ...

Variable is x = 10; y = 6;

x + y = 16
x - y = 4
x * y = 60
x / y = 1.6666666666666667
x % y = 4
x++ = 10
++x = 12
x-- = 12
--x = 10

النتائج

التفاصيل بالشريحة التالية
لكن انتبه جيدا للنتائج ؟

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Operators

عمليات الجمع والطرح والضرب والقسمة ..لن أقوم بشرحها ^_* ...

اشارة ال %تعني **باقي** القسمة ..فلذالك باقي قسمة ١٠ على ٦ هو ٤ ..

والآن نذهب الى x++ و ++x ..؟؟!!
الجملتان تعني اضافة رقم واحد للمتغير x .. بشكل آخر x = x + 1; ..^_*
لكن الفرق أن الأولى تعني x ثم زيادة واحد .. أما الثانية فتعني زيادة ١ ثم x..
ولهذا اذا لاحظت النتائج ..فإنه عندما قمنا بطباعة x++ ..قام بطباعة ١٠ .. لكن بعد هذه الحركة قام بزيادة ال x فأصبح ١١ .. وفي الحالة الثانية ..فإنه قام بزيادة ١ أولا ..فأصبحت ١١ + ١ ومن ثم طبعها فكان الناتج ١٢ ...
وهذا الكلام ينطبق على (--)..شاهد الأمثلة بالشرحة التالية ^_*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Operators

```
var x = 10, y = 6, c;
document.getElementById("arthOperator2").innerHTML =
"x++ = <b>" + (x++) + "</b> And x value is: " + x +"<br />" +
"++x = <b>" + (++x) + "</b> And x value is: " + x +"<br />" ;

var c = 5, d = 3;
console.log("c : " + c);
c++;
console.log("c++ : " + c);
++c;
console.log("++c : " + c);
console.log("d : "+ d );
c = c + (d++);
console.log("c = c + (d++) : " + c);
console.log("d : "+ d );
c = c + (++d);
console.log("c = c + (++d) : " + c);
console.log("d : "+ d );
```

تتبع المثال جيد ..وانتبه لقيمة المتغيرات بكل لحظة..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

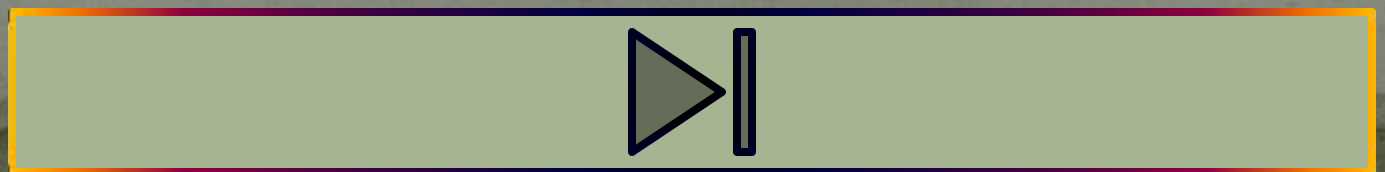
JavaScript Operators

```
c : 5
c++ : 6
++c : 7
d : 3
c = c + (d++) : 10
d : 4
c = c + (++d) : 15
d : 5
```

هذه هي النتائج الخاصة بالConsole

بهذا نكون انتهينا بإذن الله تعالى من موضوع ال Arithmetic Operator

شاهد الأمثلة من هنا ..ولا تنسى تتبع النتائج والنظر الى ال console...



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Operators

٢)Assignment Operator : وهذه العمليات تستخدم لاسناد القيم / حفظ القيم داخل متغيرات.. وقد استخدمنا اشارة ال (=) للاسناد في أحد الأمثلة السابقة ^_*..

العمليات التي يمكن استخدامها في هذا النوع هي:

= و =+ و =- و =* و =/ و =%

والآن ..لنرى مثال على هذه العمليات ^_____*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Operators

```
var x = 5, y = 2, c;
console.log("x : " + x + " && y : " + y + " && c : " + c);
x += 5;
console.log("x += 5 : " + x + " This mean x = x + 5; ");
x -= 5;
console.log("x -= 5 : " + x + " This mean x = x - 5; ");
x *= 5;
console.log("x *= 5 : " + x + " This mean x = x * 5; ");
x /= 5;
console.log("x /= 5 : " + x + " This mean x = x / 5; ");
y = x;
console.log("y = x : " + " This mean y = " + y + " && x = " + x + " && x and y = " + x);
y %= x;
console.log("y %= x : " + y + " This mean y = y % x");
```

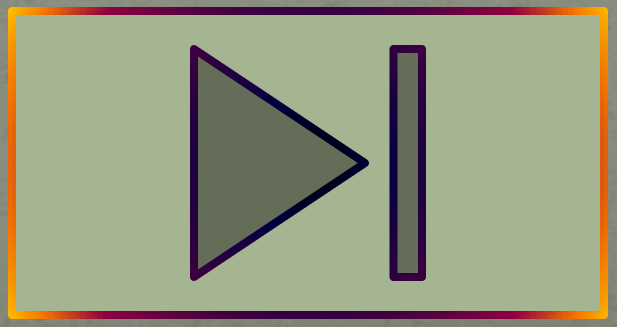
تتبع المثال جيدا ...وبنانا على ما
هو أمامك ..اكتب النتائج ^_*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Operators

```
x : 5 && y : 2 && c : undefined
x += 5 : 10 This mean x = x + 5;
x -= 5 : 5 This mean x = x - 5;
x *= 5 : 25 This mean x = x * 5;
x /= 5 : 5 This mean x = x / 5;
y = x :  This mean y = 5 && x = 5 && x and y = 5
y %= x : 0 This mean y = y % x
```

النتائج ^_^*



انظر الى المثال ..عدل عليه ..قارن النتائج ..هل كانت كما توقعت ؟

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Operators

٣) String Operator: العمليات على النصوص.. بكل بساطة يمكنك استخدام عملية واحدة فقط على النصوص..وهي عملية ال الجمع (+) ..وقد استخدمناها بكثرة في الأمثلة السابقة ...وهذه العملية تعني concatenation ... بالإضافة الى ذلك يمكنك استخدام ال (=+) ...

ملاحظة: اذا تمت عملية الجميع بين رقم ونص .. فإن الناتج هو نص..ولا يكون الناتج رقما في الجمع الى بين رقمين ...

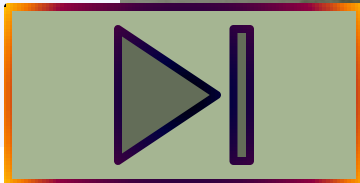
ملاحظة: في حال أردنا أن نجعل رقمين يتعاملان مع عملية الجمع مثل النص ..يمكننا أن نستخدم ال "" بينهما ..^_* ..أو بوضع أحد الأرقام أو جميعهما داخل ""... شاهد المثال في الصفحة التالية:

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Operators

```
// String Operator
console.log("Anees" + "Hikmat");
console.log("Anees " + "Hikmat");
console.log("Anees" + " Hikmat");
console.log("Anees" + " " +"Hikmat");
console.log(5 + " " + 6);
console.log(5 + "" + 6);
console.log("5" + 6);
console.log(5 + "6");
console.log("5" + "6");
var x = "anees", y = "hikamt";
x += " ";// هنا سيتم اذافة فراغ
console.log(x);
x += y;
console.log(x);
y += " " + x + " ^_^";
console.log(y);
</script>
```

تتبع النتائج..وطبق المثال ..ثم انظر
الى النتائج ..انتبه للتفاصيل
الموجودة ...



تفكر ...

رب اغفر لى وتب على إنك أنت التواب الرحيم .
رب اغفر لى وتب على إنك أنت التواب الغفور .
اللهم إنك عفو تحب العفو فاعف عنى .

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Operators

٤) Bitwise Operator : العمليات التي تتعلق بهذا الأسلوب لن أطيل بها..لإنك يجب أن تعرف موعا ما في ال logic ..لكن ما سأقوله أن هذا النوع يقوم التعامل مع الأرقام من خلال ال bit.. كل تمثيل ثنائي بجهاز الحاسوب.. هذا التمثيل له قواعد معينة ..مثل تحديد السالب والموجب..

الآن ما يهمني في الموضوع هو أن تعرف أن الجافا سكربت ..تحجز ٣٢ خانة للتعامل بال bitwise ..

العمليات الي يمكن استخدامها بال bitwise هي & و | و ~ و ^ و << و >>

الآن ..كيف مبدأ العمل... ؟؟ ال & هذه تعني bitwise and ..وهي تختلف عن ال &&.. بحيث تقوم على تحويل الرقم الى bitwise ومن ثم عمل & بين ال bit الخاصة بالرقم الأول وال bit الخاصة بالرقم الثاني ..والناتج الذي يعود يكون bit يحول الى رقم عادي ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Operators

والآن شاهد المثال التالي:

```
// Bitwise Operator
console.log("5 & 4 :" + (5 & 4));
```

```
0101  5
0100  4
----
0100  4
```

And Bitwise Operator

قاعدة
1 = 1 && 1
0 = 0 && 1
0 = 0 && 0

```
// Bitwise Operator
console.log('5 | 4 :" + (5 | 4));
```

```
0101  5
0100  4
----
0101  5
```

or bitwise operator

قاعدة:
1 = 1 | 1
1 = 0 | 1
0 = 0 | 0

```
// Bitwise Operator
console.log('5 ^ 4 :" + (5 ^ 4));
```

```
0101  5
0100  4
----
0001  1
```

xor bitwise operator

قاعدة
0 = 1 ^ 1
1 = 0 ^ 1
0 = 0 ^ 0

```
// Bitwise Operator
console.log("~5: " + (~5));
```

```
0101  5
----
1010  10
```

not bitwise operator

~1 = 0
~0 = 1

الآن ..اذا قمت بتنفيذ الأمثلة التي تشاهدها..جميعها ستعمل بشكل الصحيح..والذي يظهر أمامك ..باستثناء (~) .

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Operators

هل قمت بتطبيق الأمثلة السابقة ؟

ماذا كان النتاج للصورة الرابعة؟ هل كان ١٠؟

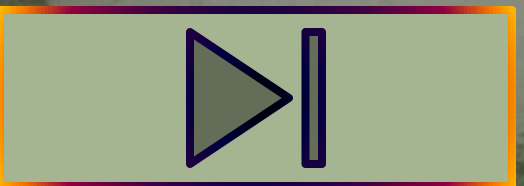
الجواب لا ..كان الجواب -٦ .. والسبب في ذلك أن الجافا سكربت تحجز ٣٢ خانة لكي تتعامل مع الرقم ..وليس ٤ فقط .. وبهذا فالشكل يكون وكأنه التالي:

00000(32 time)0101

ومع استخدام عملية ال not تصبح الأصفار ١ ويصبح ال ١ صفر..s

11111(32 time)1010

(افصد ب ٣٢ هو عدد خانات الرقم كامل)...والآن كما لاحظت ..فإن الناتج الجديد يختلف لهذا السبب.. ^_*... شاهد المثال ^_*



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Operators

٥) Unary Operator : وهي العمليات التي يمكن تنفيذها على معامل واحد ، وهذا الأمر يذكرنا بدالة استخدمناها كثيرا جدا وهي من هذا النوع ؟..وهي ال typeof .. هذه الدالة احدى ال Unary operator ..

من أشهر الأمثلة على هذا النوع من العمليات ..

ال typeof وال delete وال void وال (+) ...الخ

ال typeof تم شرحها سابقا..

ال delete تستخدم لحذف object معين..

ال void تستخدم لتتجاهل القيمة المرجعة من ال return وتجعل قيمة ال return تساوي undefined ..^_^*

ال (+) وتكتب على الشكل التالي: x = +y; وتستخدم لتحويل نص الى رقم ^_^*..والآن لنشاهد مثالا ^_^*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Operators

```
<script>
    x = 5;
    var y = "6";
    c = x + y;
    console.log("x(5) + y('6') = " + c);
    c = x + +y;
    console.log("x(5) + y('6') = {x + + y} = " + c);// this will be return 11
    console.log(delete x);// its an object.. since its a global variable
    console.log(delete y);// if you set var keyword, cant be used delete
    console.log('<a href="javascript:void(0);">Stop Link by Void</a> :  ' + void(0));
</script>
```

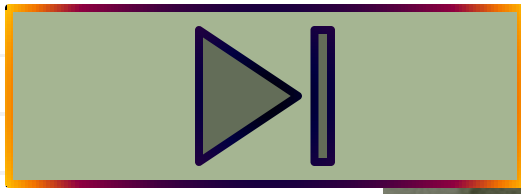
عملية الجمع

ال Unary Operator

سيتم الحذف true

لن يتم الحذف.. false

```
x(5) + y('6') = 56
x(5) + y('6') = {x + + y} = 11
true
false
<a href="javascript:void(0);">Stop Link by Void</a> :  undefined
```



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Operators

٦)Comparison Operator : هذا النوع من العمليات بختص أو يتعلق بعمليات المقارنة بين القيم المختلفة أو المتغيرات ..

للمقارنة بين المتغيرات ..يمكننا استخدام:

== و === و =! و ==! و < و > و =< و => ..

١) ال == وال === تستخدم عند المقارنة لفحص عنصرين ..هل هما متساويين أم لا .. والفرق بينهم أن ال === تقارن المساوة مع نوع المتغير

٢) ال ==! وال =! عكس النقطة واحد ..وال ==! تفارن هل القيم غير مستاوية بالنوع والقيمة

٣) > : مقارنة ..هل العنصر الأول أكبر من الثاني

٤) <: مقارنة : هل العنصر الأول أقل من العنصر الثاني.

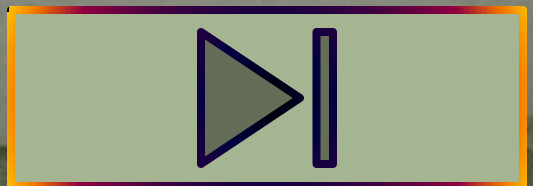
٥) =< و <=: مقارنة العنصر أكبر أو يساوي أو أقل أو يساوي ...

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Operators

```
var x = 5, y = 6, c = "5";
console.log("x = 5, y = 6, c = '5'");
console.log("==================");
console.log("x == y " + (x == y));
console.log("x == c " + (x == c));
console.log("x === c " + (x === c));
console.log("==================");
console.log("x != y " + (x != y));
console.log("x != c " + (x != c));
console.log("x !== c " + (x !== c));
console.log("==================");
console.log("x > y " + (x > y));
console.log("x > c " + (x > c));
console.log("==================");
console.log("x < y " + (x < y));
console.log("x < c " + (x < c));
console.log("==================");
console.log("x >= y " + (x >= y));
console.log("x >= c " + (x >= c));
console.log("==================");
console.log("x <= y " + (x <= y));
console.log("x <= c " + (x <= c));
```

```
x = 5, y = 6, c = '5'
==================
x == y false
x == c true
x === c false
==================
x != y true
x != c false
x !== c true
==================
x > y false
x > c false
==================
x < y true
x < c false
==================
x >= y false
x >= c true
==================
x <= y true
x <= c true
```



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Math Object

^_^..ال Math Object .. ال Math هي دالة رائعة جدا تستخدم مع الأرقام للقيام بوظائف رياضية محددة ..مثل اختيار الأرقام العشوائية وتحديد أصغر قيمة ..والتقريب ..الخ من هذه الأمور الرياضية ^_^

لاستخدام ال Math كل ما يلزمنا هو كتابة Math ومن ثم نقطة ومن ثم اسم العملية الرياضية المراد تنفيذها ... مثل Math.min(10, 15, 12)...

لهذه الدالة الكثير من العمليات الرياضية التي يمكننا استخدامها .. لذلك سنتلكم عن أكثر هذه العمليات استخداما .. ^_^

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Math Object

١) Random: وتستخدم هذه الدالة لإرجاع رقم عشوائي.
٢) Max: تستخدم هذه الدالة لإرجاع أكبر رقم.
٣) Min: تستخدم هذه الدالة لإرجاع أصغر رقم.
٤) Round: تستخدم هذه الدالة لتقريب الرقم لعدد صحيح.
٥) Ceil: تستخدم هذه الدالة لتقريب الرقم لأكبر عدد صحيح.
٦) Floor: تستخدم هذه الدالة لتقريب الرقم لإقل عدد صحيح.
٧) Constants مثل ال (PI وال SQRT2 ...)
٨) Abs: لارجاع القيمة المطلقة للرقم (تحويل من سالب الى موجب).
٩) sin, cos, tan: لإرجاع الجا أو الجتا أو الظا
١٠) Pow: تستخدم لرفع الرقم لقيمة اسية معينة ..مثل ٢ مرفوعة للقوة ٣ = ٨

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Math Object

أمثلة على الخصائص المختلفة: (الأمثلة ليست مكتوبة ..ووضعت صور لتقوم بتنفيذ الأمثلة مباشرة ورؤية النتائج)

١) random:شاهد المثال ثم انظر الى الملاحظات في الشريحة التالية: وبعد ذلك طبق المثال ..وانظر النتائج ...^_^

```
console.log("Math.random()" + Math.random());// 0 to 1 // رقم 1 لا يدخل ضمن الأرقام
console.log("Math.random() * 5 " + (Math.random() * 5)); // 0 to 4
console.log("1 + Math.random() * 5 " + (1 + (Math.random() * 5)));// 1 to 5 // 5 in
console.log("2 + Math.random() * 5 " + (2 + Math.random() * 5));// 2 to 6 // 6 in
var max = 50, min = 20;
console.log((Math.random() * (max - min + 1) + min));// 20 to 50 // 50 in
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Math Object

ملاحظات:

١) المدى للأرقام العشوائية هو من ٠ الى ١

٢) عند ضرب العدد العشوائي برقم معين ..بدون وجود عملية جمع(المثال ٢) فإن الناتج سيكون بين ال ٠ والرقم الثاني ١ بالمثال سيكون المدى بين ٠ و ٤ ... لأن ٥ - ١ = ٤

٣) عند جمع رقم مع وجود عملية الضرب ..فإن الناتج سيكون كالتالي:

القيمة الأكبر(قيمة النهاية للمدى وستكون ضمن مدى الأرقام العشوائية) - القيمة الأقل(أقل رقم عشوائي ممكن) + ١ والناتج + أقل قيمة

٥٠ - ٢٠ + ١ = ٣١ ==> اذن الشكل النهائي هو

Math.random() * 31 + 20

٤) يمكتن استخدام معادلة أخرى لإخراج أو تحديد النتائج حسب المطلوب..مثلا اذا قمت بحذف + ١ الموجودة في المثال الأخير ..فإنه الرقم الأكبر لن يكون داخل ضمن المدى للأرقام العشوائية ...

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Math Object

٢ & ٣: max & min :

الصيغة العامة:

Math.max(*n1,n2,n3,...,nX*)

Math.min(*n1,n2,n3,...,nX*)

مثال:

```
console.log((Math.min(2, 4, 1, 0, 8, 9, -5, -2)));
console.log((Math.max(2, 4, 1, 0, 8, 9, -5, -2)));
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Math Object

٤) round: لا تنسى أنها تستخدم للتقريب الرياضي.. التقريب يكون من ٠ الى ٤ الى الرقم الأقل.. ومن ٥ الى ٩ الى الرقم الأكبر...

الصيغة العامة: Math.round(x)

أمثلة:

```
console.log(Math.round(1.1));
console.log(Math.round(1.5));
console.log(Math.round(1.7));
console.log(Math.round(1.005));
console.log(Math.round(-1.1));
console.log(Math.round(-1.5));
console.log(Math.round(-1.7));
```

انتبه هنا

انتبه هنا

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Math Object

٥) ceil: تقوم هذه الدالة على تقريب الرقم الى أكبر عدد صحيح..

الصيغة العامة: Math. ceil(x)

أمثلة:

```
console.log("Math.ceil(1.1) " + Math.ceil(1.1));
console.log("Math.ceil(1.5) care: " + Math.ceil(1.5));
console.log("Math.ceil(1.7) " + Math.ceil(1.7));
console.log("Math.ceil(1.005) " + Math.ceil(1.005));
console.log("Math.ceil(-1.1) " + Math.ceil(-1.1));
console.log("Math.ceil(-1.5) Care: " + Math.ceil(-1.5));
console.log("Math.ceil(-1.7) " + Math.ceil(-1.7));
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Math Object

٦) floor: تقوم هذه الدالة على تقريب الرقم الى أقل عدد صحيح..

الصيغة العامة: Math. floor(x)

أمثلة:

```
console.log("Math.floor(1.1) " + Math.floor(1.1));
console.log("Math.floor(1.5) care: " + Math.floor(1.5));
console.log("Math.floor(1.7) " + Math.floor(1.7));
console.log("Math.floor(1.005) " + Math.floor(1.005));
console.log("Math.floor(-1.1) " + Math.floor(-1.1));
console.log("Math.floor(-1.5) Care: " + Math.floor(-1.5));
console.log("Math.floor(-1.7) " + Math.floor(-1.7));
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Math Object

٧) Constants: الثوابت..ويقصد بها الثوابت الرياضية مثل ال PI فقيمتها دائما ٣.١٤ ... لذلك يمكنك استخدامها مباشرة دون تعريف..

من الأمثلة على ال Constants:

١) PI وقيتمها ٣.١٤

٢) E وهي العدد النيبيري وقيمتها ٢.٧١

٣) LOG2E وترجع ال log لل E للأساس ٢

شاهد الأمثلة (قم بالبحث عن Math Constants js وانظر البقية ^_*) :

```
console.log("Math.PI: " + Math.PI);
console.log("Math.E: " + Math.E);
console.log("Math.LOG2E: " + Math.LOG2E);
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Math Object

٨) abs: ونستطيع من خلال هذه الدالة البسيطة بتحويل الرقم من سالب الى موجب...

الصيغة العامة: Math.abs(x) ..

ملاحظات:

١) يجب أن تكون قيمة x ..رقم

٢) اذا لم تكن رقم وكانت نص ..الناتج سيكون NAN.

٣) اذا تم وضع Null مكان x فإن الناتج هو ٠

٤) الرقم الموجب يبقى كما هو ...

```
console.log("Math.abs(-5): " + Math.abs(-5));
console.log("Math.abs(5): " + Math.abs(5));
console.log("Math.abs('anees'): " + Math.abs('anees'));
console.log("Math.abs(null): " + Math.abs(null));
console.log("Math.abs(0): " + Math.abs(0));
console.log("Math.abs(1 - 5): " + Math.abs(1 - 5));
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Math Object

٩) sin, cos, tan: باستخدام هذه الدوال، والتي تطابق تسميتها التسمية الرياضية، فإن الوظيفة أيضا هي نفسها

sin ..ترجع جا الزاوية و cos ترجع جتا الزاوية و tan ترجع ظا الزاوية ...

الصيغة العامة: Math.sin|cos|tan(x) ..

أمثلة:

```
console.log("Math.sin(60): " + Math.sin(60));
console.log("Math.cos(60): " + Math.cos(60));
console.log("Math.tan(60): " + Math.tan(60));
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Math Object

١٠) pow: هذه الخاصية مهمة جدا ^_^.. على اساس انو في اشي مش مهم P: ....^_^*

هذه الخاصية تقوم على رفع رقم معين لقوة معينة . مثل ٢ مرفوعة للقوة ٣ = ٨

الصيغة العامة: Math.pow(x, y)

ال x تمثل الرقم وال y تمثل الأس.

ملاحظة: يمكن

شاهد المثال:

```
console.log("Math.pow(2, 3): " + Math.pow(2, 3));
console.log("Math.pow(-2, 3): " + Math.pow(-2, 3));
console.log("Math.pow(-2, 2): " + Math.pow(-2, 2));
console.log("Math.pow(5, 1): " + Math.pow(5, 1));
console.log("Math.pow(1, 5): " + Math.pow(1, 5));
console.log("Math.pow(0, 3): " + Math.pow(0, 3));
console.log("Math.pow(3, 0): " + Math.pow(3, 0));
console.log("Math.pow(4, 0.5): " + Math.pow(4, 0.5));
console.log("Math.pow(4, 1 / 2): " + Math.pow(4, 1 / 2));
console.log("Math.pow(8, 1 / 3): " + Math.pow(8, 1 / 3));
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Math Object

ماذا الآن ؟!

حقيقة ..هناك بعض الأمور ..التي أحب أن اسألك عنها بعد قرائتك واطلاعك على الشرح السابق..

١) هل تسائلت كيف يمكنني طباعة رقم عشوائي صحيح؟

٢) هل تسائلت كيف يمكنني تقريب رقم لأقرب منزلتين عشريتين ؟!!

٣) وهل تسائلت ..كيف يمكنني أو هل يمكنني استخدام أكثر من Math مع بعضها البعض؟؟؟

حقيقة هذه ملاحظات مهمة ... وهناك عدة طرق للحل .. وسأقدم لكم الآن ..أسهل الحلول بإذن الله تعالى ..وهناك طرق أخرى أفضل..لكننا الآن نكتفي بالأسلوب ذو المستوى الجيد لحل المشكلة ..وأخص بهذا الكلام التقريب... ^_* ..

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Math Object

الأمثلة: (ملاحظة إن كثر عدد الخانات قد يختلف اسلوب الحل..خصوصا هذه الملاحظة للتقريب... ودرجة الدقة المطلوبة لهذا التقريب)

```
console.log((Math.round(1.76941 * 10) / 10));
console.log((Math.round(1.76941 * 100) / 100));
console.log(Math.floor(Math.random() * 50));
```

لاحظ .. في المثال واحد فإن التقريب سيكون عشري..

والثاني مئات ... (هذا الحل يسبب مشكلة في حال ظهور رقم مثل ١.٠٠٥*_^

أما الثالث فتم استخدام أكثر من Math وبذات الوقت .. تم تحويل الرقم العشوائي من عشري .... الى صحيح.. وبرأيك ...لمذاى استخدمت floor ....؟^_^

تفكر ...

أعوذ بالله من الشيطان الرجيم

(رَبَّنَا لاَ تُؤَاخِذْنَا إِن نَّسِينَا أَوْ أَخْطَأْنَا رَبَّنَا وَلاَ تَحْمِلْ عَلَيْنَا إِصْرًا كَمَا حَمَلْتَهُ عَلَى الَّذِينَ مِن قَبْلِنَا رَبَّنَا وَلاَ تُحَمِّلْنَا مَا لاَ طَاقَةَ لَنَا بِهِ وَاعْفُ عَنَّا وَاغْفِرْ لَنَا وَارْحَمْنَا أَنتَ مَوْلاَنَا فَانصُرْنَا عَلَى الْقَوْمِ الْكَافِرِينَ)

صدق الله العظيم

[سورة البقرة/٢٨٦]

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Dates

من الامور المهمة في عالم البرمجة... التعامل مع الوقت والتاريخ ...

بالنسبة للتعامل مع التاريخ ..فإنه يكون عن طريق استخدام ال Dates والموجودة داخل الجافا سكربت Object ...

التعامل مع الوقت أو التاريخ قد يكون على مستوى السنة أو الشهر أو اليوم أو الدقائق والساعات ...الخ)

والآن لنبدأ معا هذا الموضوع الشيق ^_*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Dates

*) ال Date()... إن الإستخدام أو الشكل العام للتاريخ والوقت يطبع باستخدام ال Date() ويقوم هذه الدالة على طباعة اليوم والشهر والسنة وفرق الوقيت مع غرينتش، بالإضافة الى الدولة التي أن

```
<script>
    console.log(Date());
</script>
```

```
Wed Sep 24 2014 09:51:43 GMT+0300 (Jordan Daylight Time)
```

*) أيضا يمكننا إضافة تاريخ محدد ووقت محدد نريد طباعته date(DateString) ... مثل:

```
d = new Date("5/23/2004 11:55:00");
console.log(d);
d = new Date("June, 5, 2014");
console.log(d);
```

```
Sun May 23 2004 11:55:00 GMT+0300 (Jordan Daylight Time)
Thu Jun 05 2014 00:00:00 GMT+0300 (Jordan Daylight Time)
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Dates

*) كما يمكنني تحديد الوقت بنائا على ال millisecond ..ويبدأ التأريخ من عام ١٩٧٠ ميلادي ...ولتمثيل هذه الصيغة يمكننا وضع رقم داخل ال dates...ويعتمد هذا الأسلوب على ال UTC العالمي للوقت نظام ال شاهد مثالا:

```
d = new Date(0);
console.log(d);
d = new Date(3600000);
console.log(d);
```

سيتم احتساب ساعة
كل ثانية = 1000
جزء من الثانية

```
Thu Jan 01 1970 02:00:00 GMT+0200 (Jordan Standard Time)
Thu Jan 01 1970 03:00:00 GMT+0200 (Jordan Standard Time)
```

لاحظ كيف كان القيمة ٠ مثلت نقطة البداية .. وكيف أثر فرق التوقيت على الساعة الظاهرة وكيف أثر ٣٦٠٠٠٠٠ جزء من الثانية على النتائج بجعل الساعة ٣ ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Dates

*) أيضا يمكنك تحديد الوقت والتاريخ باستخدام نظام الأرقام ال ٧ ^_^ * ..

هذا النظام يقوم على وضع الأرقام بشكل متسلسل من السنة ثم الشهر ثم اليوم ثم الساعة ثم الدقائق ثم الثواني ثم أجزاء الثانية ^_^...

لاحظ المثال:

```
var d = new Date(2014,10,10,11,33,30,0);
console.log(d + "//// new Date(2014,10,10,11,33,30,0): ////  ");
var d = new Date(2014,0);
console.log(d + "//// new Date(2014,0): ////  ");
var d = new Date(2014,11,10);
console.log(d + "//// new Date(2014,11,10): ////  ");
```

شهر

```
Mon Nov 10 2014 11:33:30 GMT+0200 (Jordan Standard Time)//// new Date(2014,10,10,11,33,30,0): ////
Wed Jan 01 2014 00:00:00 GMT+0200 (Jordan Standard Time)//// new Date(2014,0): ////
Wed Dec 10 2014 00:00:00 GMT+0200 (Jordan Standard Time)//// new Date(2014,11,10): ////
```

اذا لاحظت الأمثلة فإن الأشهر تبدأ من ٠ الى ١١ داخل الجافا سكربت ^_^

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Dates

اذا لاحظت أشكال التواريخ السابقة فهي قد تكون صعبة القرائة ..لكن هذه ليست مشكلة .. أيضا قد تحتاج ان تضع التاريخ أو يعرض على شكل ال UTC وهذا يمكنك فعله بسهولة عن طريق استخدام ال..

toDateString() وال d.toUTCString() ...

شاهد مثالا: ^_*

```
var d = new Date(2014,10,10,11,33,30,0);
console.log(d.toUTCString() + "/// new Date(20
var d = new Date(2014,10,10,11,33,30,0);
console.log(d.toDateString() + "/// new Date(2
```

```
Mon, 10 Nov 2014 09:33:30 GMT/utc
Mon Nov 10 2014/// new Date(2014,
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Dates

والآن ..بعد تعرفنا على طرق كتابة التواريخ والأوقات ..وبعدة أشكال ..يجب أن نتعلم كيف يمكننا التعمل مع هذ التواريخ أو الوقات .. كيف يمكنني جلب توقيت معين أو اضافة وقت معين .. أو حساب عدد أجزاء الثانية ..الخ من هذه الأمور.. فإن البرمجة ليست مجرد أوامر ..إنما هي فكر يتحكم بتلك الأوامر ..

والآن ..سنقوم بتقسم الدوال الى Get و Set ..

ملاحظة: معظم الدوال التي سنقوم بذكرها يمكن أن يكون لها أكثر من شكل للوقت ..مثلا getUTCDate() و getDate() ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Dates

١) Get Date Method: هذه الدوال تفيدنا بجلب قيم معينة من داخل التاريخ أو الوقت ..ومن هذه الدوال المستخدمة:

- getDate(): تقوم على ارجاع الأيام على شكل رقم من ١ الى ٣١
- getDay(): تقوم على ارجاع اليوم خلال الأسبوع من ٠ الى ٦
- getFullYear(): تقوم على ارجاع السنة من ٤ خانات
- getHours(): تقوم على ارجاع الساعة من ٠ الى ٢٣
- getMinutes(): تقوم على ارجاع الساعة من ٠ الى ٥٩
- getMonth(): تقوم على ارجاع الشهر من ٠ الى ١١
- getSeconds(): تقوم على ارجاع الثواني من ٠ الى ٥٩
- getTime(): تقوم على ارجاع الوقت من ١٩٧٠ الى الآن بأجزاء الثانية
- getMilliseconds(): تقوم على ارجاع اجزاء الثانية من ٠ الى ٩٩٩

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Dates

شاهد الأمثلة: (لا تنسى أن الهدف من وضع الصورة هو أن تقوم بتطبيق المثال وليس النظر والتجاوز فقط ... التعلم يحتاج الى التطبيق):

```
d = new Date(2014,11,10);
console.log(d.getFullYear() + "   " + d.getDate() + "  //// var d = new Date(2014,11
d = new Date();
console.log(d.getMinutes() + "  " + d.getSeconds() + "  " + d.getMilliseconds() +
            "  " + d.getHours() + "   " + d.getFullYear());
```

```
2014   10  //// var d = new Date(
42  10  218  11   2014
```

لاحظ النتائج: الآن في المثال الأول ستكون النتائج ثابتة لأنه تم تحديد التاريخ ب ١٠ / ١١ /٢٠١٤ ... بينما في المثال الثاني ..فإنه يعمل على الوقت الفعلي ..لذلك ستقى النتائج في تغير مستمر مع كل عملية تحديث للصفحة ..(reload by F5).

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Dates

والآن ..والآن ماذا ..هل ننتقل الى الجزء الثاني set method ؟!

الجواب لا ..في تفكير شرير يدور برأسي الآن ..

لو طلبت منك الآن ..أن تطبع لي الوقت الفعلي بكل عملية تحديث للصفحة بنائا على الشكل التالي:

١) 235 – 20 - 10 – 5 – 2014.... (طبعا جميع الأرقام الناتجة هنا هي افتراضية تمثل شكل المخرجات الذي أريده أن يظهر والذي يمثل السنة واليوم (١ الى ٣١) والدقائق والثواني وأجزاء الثانية).

٢) 14115492168886 ^_^ 12 * 24 - 8 / 2014 طباعة هذا الشكل: والذي يمثل السنة / الشهر – اليوم * الساعة ^_^ ثم أجزاء الثانية من ١٩٧٠ الى الآن ..)

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Dates

الحل:

```
// Answer 1 d = new Date();
console.log(d.getFullYear() + " - "  + d.getDate() + " - "  + d.getMinutes() +
        " - " + d.getSeconds() + " - " + d.getMilliseconds())
// Answer 2 d = new Date();
console.log(d.getFullYear() + " / "  + d.getMonth() + " - "  +
        d.getDate() + " * " + d.getHours() + " ^_^ " + d.getTime())
script>
```

الهدف من هذا المثال هو لفت انتباهك ..الا أنه يمكنك التلاعب بالدوال وساتخدامها حسب حاجتك ... البرمجة فن ..وليست حفظ ..

بالنسبة لحل هذا السؤال ..يمكنك حله بهذه الطريقة ..وهناك طريق أخرى مثل استخدام مكتبة جاهزة للتعامل مع ال dates او أي طريقة أخرى ..

Google It --- ^_^ --- ابحث عن change js date format ..

تفكر ...

في عام ١٩٤٣ ...انتشرت البطالة بين الشباب الألمان بشكل كبير...
فماذا كان حل هتلر لهذه المشكلة الإقتصادية ..والتي أرقت المجتمع في تلك الفترة ؟!

قام هتلر بالتحدث أمام الرابطة الاشتراكية الوطنية للمرأة فقال إنه بالنسبة للمرأة الألمانية: "لا بد أن يتركز عالمها حول زوجها وعائلتها وأطفالها وبيتها" بالإضافة الى عامل انتاج السلاح .. وصلت المانيا خلال فترة وجيزة الى ما يسمى العمالة الكاملة .....وبالتحليل الإقتصادي قالو أن "عودة النساء للمكوث في منازلهن حتى منحت الفرصة للرجال للحصول على الوظائف التي كن يشغلنها أنعشت البلاد في تلك الفترة ... وقللت من مشاكل الشباب الألمان"

هذا هو التكريم الحقيقي للمرأة ..فهي ملكة ..ويكفي للملكة جلوسها على كرسي الإمارة ..تدير شؤون مملكتها ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Dates

٢) Set Date Method: يمكننا باستخدام هذه الدالة أن نقوم بوضع وقت أو جزء من الوقت الى date ^_^...ومن الدوال المستخدمة في ذلك:

- setDate(): هذه الدالة تقوم على وضع اليوم على شكل رقم من ١ الى ٣١
- setFullYear(): هذه الدالة تقوم على وضع السنة ويمكن اضافة الشهر واليوم كخيار اضافي ^_^*
- setHours(): هذه الدالة تقوم على وضع الساعة من ٠ الى ٢٣
- setMilliseconds(): هذه تقوم على وضع أجزاء الثانية من ٠ الى ٩٩٩
- setMinutes(): تقوم هذه الدالة على وضع الدقائق من ٠ الى ٥٩
- setMonth(): تقوم هذه الادالة على وضع الأشهر من ٠ الى ١١
- setSeconds(): تقوم هذه الدالة على وضع الثواني من ٠ الى ٥٩
- setTime() : تقوم هذه على وضع أجزاء الثانية من ١٩٧٠ الى الآن.

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Dates

الآن شاهد الأمثلة (طبق لكي تستطيع حل السؤال بالشريحة التالية):

```
console.log(
d = new Date();
d.setFullYear(2014,11,10);
console.log(d.getFullYear() + "    "  + d.getDate()
d.setDate(7);
console.log(d.getFullYear() + "    "  + d.getDate()
d.setSeconds(10);
console.log(d.getSeconds() + "    "  + d.getDate()
```

```
************************************ Set Method ************************************
2014    10  ////  d.setFullYear(2014,11,10);  d.getFullYear() + '  '  + d.getDate()
2014    7  ////  d.setDate(7);  d.getFullYear() + '  '  + d.getDate()
10    7  ////  d.setSeconds(10);  d.getSeconds() + '  '  + d.getDate()
```

النتائج

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Dates

سؤال ^_^: بنائـا علـى مـا تعلمنـا ..سـأطلب منـك الآن ..أن تقـوم باسـتخدام ال Date() وال Set وال Get التي تعلمناها ..بطباعة ماذا سيكون التاريخ بعد ٢٠ يوم من التاريخ التالي: ٢٤-٩-٢٠١٤ ... ماذا ستفعل ؟

كمثال: اليوم هو ٢٤-٩-٢٠١٤ ..بعد ٥ أيام سيكون التاريخ هو ٢٩-٩-٢٠١٤

الآن دورك ^_^... طبعا هنا لتستطيع حل هذه المعادلة ..تذكر أنك تعرف عن عملية الجمع ... ^_^*

(((((((((((((((((٠ _____الحل سهل_____ ٠))))))))))))))))

الناتج هو : 2014/10/14

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Dates

الحل مع تطبيق هذه الأمثلة مهم جدا جدا:

```
// Answer 1
d = new Date(2014,8,24);
d.setDate(d.getDate() + 20);
console.log(d.getFullYear() + "/" + (d.getMonth() + 1) + "/" + d.getDate() +
// Answer 2
d = new Date();
d.setDate(d.getDate() + 20);
console.log(d.getFullYear() + "/" + (d.getMonth() + 1) + "/" + d.getDate() +
// Answer 3
d = new Date();
d.setDate(d.getDate() + 20);
console.log(d.toDateString() + "  ////d = new Date(); d.toDateString();");
```

```
2014/10/14  ////d = new Date(2014,8,24);  d.setDate(d.getDate() + 20);
2014/10/14  ////d = new Date(); d.setDate(d.getDate() + 20);
Tue Oct 14 2014  ////d = new Date(); d.toDateString();
```

هل ركزت بالنتائج؟؟؟!!

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Dates

نعم في الحل ١..عندما قمت بطلب الجمع لشهر ٩ فإنني سأقوم بوضع الشهر ٨ (تذكر من ٠ الى ١١) وهذا يعني شهر ٩ ..أما بخصوص عملية الجمع + ١ الموجودة فهي لطباعة الشهر بالرقم الذي يفهمه المستخدم .. يعني الناتج سيكون هو ١٤/٩/٢٠١٤ .. وليصبح الناتج صحيحا بالنسبة للمستخدم الذي سيقرأ التاريخ ..علينا أن نضيف واحد ..^_^* ..وبذلك وكأننا أصبحنا نتعامل على أساس ١ الى ١٢ ^_^* فيصبح الشهر هو ١٠ بدلا من ٩...

أما الثاني فهو سيقوم بأخذ الوقت الحالي .. ومن ثم قمت بإضافة ال ٢٠ يوموبما أن تاريخ اليوم هو ٢٤/٩/٢٠١٤ أيضا فإن الناتج هو ١٤/٩/٢٠١٤ ..لأن النظام يرقم الأشهر من ٠ الى ١١ ..وبنفس الأسلوب ثمنا بجمع الرقم ١

أما الثالث.. فلاحظ أنني بدون جمع الرقم واحد ..طبع الناتج بشكله الصحيح ..نعم عندما تقوم بطباعة التاريخ كامل مباشرة ..ليس باستخدام ال GET او Set سيقوم النظام لوحده بعمل المطلوب ^_^....

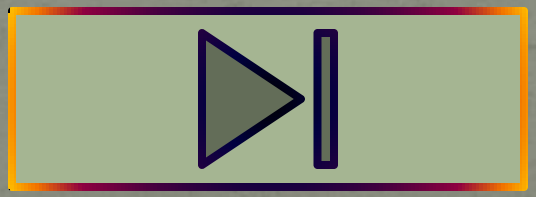
أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Dates

الآن ..بحمـــد الله تعـــالى وفضـــله ..انتهينـــا مـــن الحـــديث عـــن ال Dates

أرجو أن تكون قد طبقت الأمثلة ..فهي مهمة والفروق دقيقة تظهر أثناء التطبيق والتجربة ...

والآن ..لنشاهد تطبيق جميع الأمثلة معا ...



تفكر ...

من أقوال البخاري رحمه الله تعالى

"ما أردت أن أتكلم بكلام فيه ذكر الدنيا إلا بدأت بحمد الله والثناء عليه".

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Boolean

لقد تحدثنا في أول هذه الدورة المتواضعة ..عن المنطق في الحاسوب (Boolean) ... وقلنا أن هذا المنطق إما أن يكون صحيحا وإما أن يكون خاطئا ..(True or False) ... والآن ..سنتطرق معا .بإذن الله تعالى الى مزيد من التفاصيل والمعلومات حول هذا الموضوع ..

لمذا نستخدم المنطق في الحاسوب؟

ج) أريدك أن تعرف أن الحاسوب كله منطق .. ومنطيقه مبني على ٠ أو ١ ..خطأ أو صح ..نعم أو لا ..لذلك حينما نريد أن نقوم بكتابة أي برنامج ..يلزمنا التفكير بالطريقة أو الآلية التي يمكننا من خلالها تطوير البرامج والتطبيقات .. وهذا أمر مهم جدا ..وخصوصا حين يحين الوقت للتعامل مع الشروط وعمليات المقارنة ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Boolean

ماذا الآن ؟!

أول أمر أريدك أن تعرفه ..أن ناتج أي علاقة منطقية من التي تعلمناها سابقا هو إما • أو ١ ...مثل 2 > 5 ..هل الخمسة أقل من ٢ ؟ ..الجواب لا ..اذا فهذا ناتج بالحاسوب (false)...

الآن لمعرفة ناتج علاقة منطقية ..يمكنك استخدام طريقتين ..

١) باستخدام ال boolean
٢) بوضع العلاقة المنطقية مباشرة في جملة طباعة ..

طبعا هذا الكلام أو الطريقتين ..يندر أن نستخدمهم لوحدهم لمعرفة ناتج العلاقة المنطقية ..والسبب في ذلك ..أنك ستقرأها لوحدك أو ستعرف النتيجة بخبرتك أو بالنظر الى العلاقة فقط ..^ ^..وتستخدم غالبا لفحص ناتج العلاقة .. وهذاأمر مهم فيما بعد ..عندما نصل لأي أمر سيحصل به شرط أو مقارنة...

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Boolean

أمثلة:

```
document.getElementById("bool-1").innerHTML = Boolean(2 > 5);
document.getElementById("bool-2").innerHTML = 2 > 5;
document.getElementById("bool-3").innerHTML = Boolean(2 < 5);
document.getElementById("bool-4").innerHTML = 2 < 5;
```

false
false
true
true

ما النتائج التي رأيتها ... ؟هل تطابق هذه الصورة ؟

اذا كانت الإجابة نعم ..فهذا الأمر رائع جدا.. أنت الآن مستعد للإحتراف.. ^_* ويبقى علينا الآن أمر واحد قبل الإنطلاق السريع ..وهو .....

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Boolean

الآن ..سأخبرك بأمر مهم ..

هناك قاعدة تقول ..."كل شيء يمثل قيمة حقيقة فهو true وكل شيء ليس له قيمة حقيقة فهو false.."

*) الجزء الأول من القاعدة " كل شيء يمثل قيمة حقيقة فهو true" وهذا يعني أن ال true وال 1 وال ٢ وال ٣...الى n والنصوص وناتج العلاقات المنطقية الصحيحة ..في قيم حقيقة ..فتكون دائما true..

شاهد المثال:

```
le.log("Boolean(true) = " + Boolean(true));
le.log("Boolean('true') = " + Boolean("true"));
le.log("Boolean('false') = " + Boolean("false") + " // False S
le.log("Boolean(5) = " + Boolean(5));
le.log("Boolean(5 > 2) = " + Boolean(5 > 2));
le.log("Boolean(5 + 3 - 4 * 8) = " + Boolean(5 + 3 - 4 * 8));
le.log("Boolean(Math.PI) = " + Boolean(Math.PI));
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Boolean

هل طبقت المثال السابق؟ ..ماذا كانت النتائج؟ بكل تأكيد جميعها true ..

*) أما الجزء الثاني من القاعدة " وكل شيء ليس له قيمة حقيقة فهو false"
هذا يعني أن أي قيمة غير معرفة أو تعطيني ٠ أو false أو غير صحيحة داخل علاقة منطقية أو المتغيرات الفارغة أو النصوص الفارغة ..هي False ..

شاهد المثال:

```
lean(false) = " + Boolean(false));
lean(0) = " + Boolean(0));
lean(-0) = " + Boolean(-0));
lean(15 / 'Text') = " + Boolean(15 / "Text"));
lean('') = " + Boolean(''));

lean(variable) = " + Boolean(variable));
lean(null) = " + Boolean(null));
```

NaN
Empty String

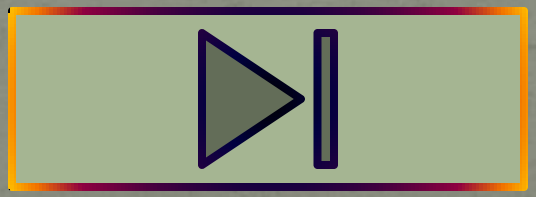
Empty Var value
null

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Boolean

الآن ..بعد استعراضنا الأمثلة السابقة ..يفترض أن تكون الآن ..قادرا على تخمين ناتج أي علاقة منطقية ..وهذا الهدف من هذا الدرس ..وهذا أمر مهم ...

والآن ..أريدك أن تقوم باستعراض المثال مرة أخرى ...لكن ..قم بالتغيير والتجربة والتعديل بالقيم ..وانظر النتائج ...



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

المصفوفات ... لقد تحدثنا عن المصفوفات ..وذكرنا معلومات كثيرة حولتها ..لكن الآن وصلنا لمرحلة ..يجب أن نعرف كيف نعرفها ..نستخدمها ..نضيف ..نحذف.. نحدث المعلومات الخاصة بها ... بالإضافة الى ما هي الدوال الرائعة والتي يمكن استخدامها مع هذه المصفوفات ..

والآن لنبدأ على بركة الله تعالى ...

المصفوفات هي نوع من أنواع البيانات ..صمم خصيصا لحفظ أكثر من قيمة داخل متغير واحد ويمكن تعريف المصفوفة بطريقتين :

```
var x = [];// Good way
var y = new Array();// Bad way
```

```
var x = [1, 5, 54 , 1];// Good way      ١
var y = new Array(1, 5, 54, 1);// Bad way  ٢
document.getElementById("bool-1").innerHTML = x[0] + " " + y[0];
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

في الطريقتين السابقتين ..قمنا بتعريف مصفوفة ..

لكن الطريقة الأولى هي الجيدة، وقمنا مباشرة بتعريف المصفوفة عن طريق استخدام ال [ ] .. ووضع القيم بداخلها

أما في الطريقة الثانية وهي الطريقة السيئة، فإنه قمنا باستدعاء ال constructor الموجود بالبنية التركيبية للجافا سكربت ..(استدعاء غير مبرر)!!

والآن سبدأ بإذن الله تعالى الشرح ..بنائا على الطريقة الأولى .. وبذات الوقت سنتطرق لاختلاف في بعض الأمثلة أو ملاحظات في حال أنك قمت باستخدام الأسلوب الثاني ...

JavaScript Arrays

عند كتابة أي مصفوفة يرجى مراعاة ما يلي:

١) استخدم ال [ ] لتعريف المصفوفة.

أنيس حكمت أبوحميد aneeshikmat@gmail.com

٢) استخدم الفاصلة للفصل بين العناصر

٣) آخر عنصر في المصفوفة لا يأتي بعده فاصلة ..(قد يسبب مشكلة مع اختلاف المتصفحات).

٤) الأسطر والمسافات يتم تجاهلها من قبل المصفوفة

٥) كل قيمة في المصوفة لها عنوان ..المصفوفة تحفظ على شكل key/value

٦) يمكن اعطاء المصفوفة القيم مباشرة ..ويمكن تجاهل ذلك وإعطاء المصفوفة القيم في مراحل أخرى حسب الحاجة.

٧) ليتم استدعاء قيمة من داخل مصفوفة يتم وضع اسم المصفوفة ثم [key] مثل arrayName[o] ..وهنا قمت بجلب العنصر الأول من المصفوفة

٨) المصفوفات يتم عنونتها ابتدائا من ال ٠، وال٠ يمثل أول قيمة.

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

شاهد أمثلة على الملاحظات السابقة:

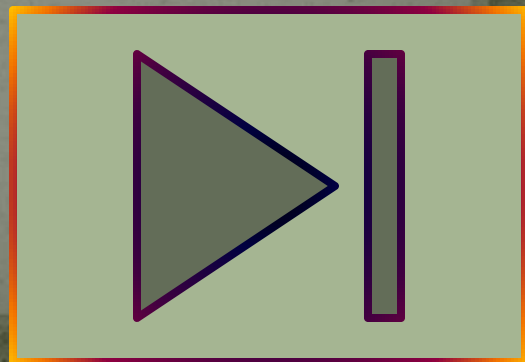
```
var x = [1, 5, 54 , 14];
console.log("x[1] = " + x[1]);// result is 5
console.log("x[2] = " + x[2]);// result is 54
// --
var x = [1, 5, 54 , 14];
// or
var z = [
1,
2,
3
];
```

اعطاء القيم مباشرة

أكثر من سطر..لكن يتم تجاهل ذلك..ويتم تجاهل المسافات..

```
var a = [];
a[0] = "anees ";
a[1] = "HIKMAT ";
a[2] = "Abu-hamid ";
console.log(a[0] + a[1] + a[0] + a[2]);
```

اعطاء القيم بعد تعريف المصفوفة



أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

والآن ..ماذا ..اييييه مش عارف في كثير أشياء :P ^_^ *

هل تذكر ال typeof ؟؟ ... ماذا لو قمت باستخدامها مع المصفوفات ..؟؟ ماذا سيكون الناتج؟

سيكون الناتج هو object ..!!! ..نعم object ..فالمصفوفة نوع خاص من ال object ... وهي فعليا أسلوب أفضل للأمور التي تحتاج الى تمثيل من الصفوفات .. ويكون ال key الخاص بهذه المصفوفة جميعها أرقام ..أما ال object فقد تكون اسم نصي .. ومن هذا المنطلق يظهر لنا مفهوم جديد يسمى بال Associative Arrays...

ال Associative Arrays هي مصفوفة تكون عناوينها نصوص بدلا من الأرقام ... وهذا النوع تدعمه العديد من لغات البرمجة ..لكن مصفوفات الجافا سكربت لا تدعم هذا النوع ..ويمكنك استخدام ال object اذا احتجت الى ذلك ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

اذا ..وبنائا على ما ذكرناه في الشريحة السابقة ..

فإننا ننوه الى التالي بشكل مفصل..

استخدم المصوفة حين تكون العناوين التي لديك أرقاما ...وهي نقطة القوة لدى المصفوفة ...

استخدم ال object حين تكون العناوين التي لديك نصوصا ...

ولا تنسى أن مصفوفات الجافا سكربت .. لا تدعم ال Associative Arrays

..وللبرهنة على ذلك ..شاهد المثال بالشريحة التالية:

*) ملاحظة: arrayName.length تعني عدد العناصر الموجودة داخل المصفوفة ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

```
var x = [1, 5, 54 , 14];
var y = new Array();
y["a1"] = 1;
y["a2"] = 2;
y["a3"] = 3;

document.getElementById("bool-1").innerHTML = "x.length = " + x.length;
document.getElementById("bool-2").innerHTML = "y.length = " + y.length;
document.getElementById("bool-3").innerHTML = "Array.isArray(x) = " + Array.isArray(x);
document.getElementById("bool-4").innerHTML = "Array.isArray(y) = " + Array.isArray(y);
document.getElementById("bool-5").innerHTML = "typeof(x) = " + typeof(x);
document.getElementById("bool-6").innerHTML = "typeof(y) = " + typeof(y);
```

ترجع عدد عناصر المصفوفة

تستخدم للتأكد هل هذه مصفوفة أم لا ..والناتج إما true or false

النتائج

x.length = 4
y.length = 0
Array.isArray(x) = true
Array.isArray(y) = true
typeof(x) = object
typeof(y) = object

ملاحظة: بالنسبة لل isArray فهي خاصية مدعومة بالمتصفحات الحديثة ..وغير مدعومة من قبل ال ie8 ..وهناك طرق أخرى لمعرفة ذلك ..

Google it: Check if object is array

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

هل لاحظ المثال السابق...؟؟ ..

نعم ..لقد كان طول المصفوفة التي أعطيناها عناوين نصية مثل"a1" ..صفر... بينما المصفوفة x والتي أخذت عناوين رقمية ..كان الناتج هو ٤ ....

وبذات الوقت ..عند فحص المصفوفتان للتأكد هل هي مصفوفة أم لا ..سترجع مصفوفة ..وسترجع الإثنتان object...

والآن ..في أول موضوعنا قلنا أنه هناك أسلوبين لتعريف المصفوفة ..الأسلوب السيء والأسلوب الجيد .. وذكرنا السيئة الأولى للأسلوب السيء ..وهو وجود استدعاء غير مبرر لل constructor الخاص بالمصفوفة والمبني تلقائيا بالجافا سكربت ...والثاني هو ..انظر الشريحة التالية *__*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

```
var z1 = new Array(10, 20, 30);
var z2 = [10, 20, 30];
var z3 = new Array(10);
var z4 = [10];

console.log(z1[0]);
console.log(z2[0]);
console.log(z3[0]);
console.log(z4[0]);
```

نفذ هذا المثال ولاحظ
الكارثة الناتج من جراء
استخدام الأسلوب السيء

كارثة !! ... مصطلح تقني هذا..ما عليكم منو ^_* .....
نفذ المثال للأهمية ..وانظر الناتج ؟...

برأيك لماذا كانت نتيجة ال z3 ..تختلف عن المصفوفات الأخرى ؟!

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

إن ما رأيته من نتيجة وهي ال undefined .. كانت لأنك قمت بتعريف ١٠ عناصر للمصفوفة وجميعهم undefind ..وبمعنى آخر ..قمت بانشاء مصفوفة ستحتوي على ١٠ قيم ..ولم تعطيها أياَ منها ...

لذلك اعتبرت ال 10 ليست كقيمة ..وإنما كعدد للعناصر المراد انشائها ...!!

لذلك انتبه ^_^*

والآن ..لنأتي الى موضوع آخر ...كيف يمكنني اضافة عنصر جديد للمصفوفة بعد تعريفها ؟

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

الجواب هو: يمكننا اضافة عنصر جديد للمصفوفة عن طريق

١) إما تحديد عدد العناصر..ومن ثم وضع عدد العناصر الناتج ك key للقيمة الجديدة

٢) إما أن تحدد ال key بشكل يدوي مباشرة.

٣) باستخدام ال push.

٤) باستخدام ال unshift.

٥) ياستخدام ال splice.

حقيقة ..لكل من هذه الوسائل استخدامتها ...ولكل طريقة ميزة خاصة بها ^_^.. ولكن هناك بالمجمل طريقة سيئة ..وطريقة جيدة ^_^

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

الطريقة ١)

اذا كنت تذكر ..فإننا قلنا أن المصفوفة يبدأ ترقيمها من الصفر ..ولهذا اذا وجد عندي المصفوفة التالي [1,2,3] ..فعناوين الأرقام ستكون من ٠ الى ٢

الآن ..اذا قمنا باسترجاع ال length ..ماذا سيكون الناتج ؟ سيكون الناتج هو ٣ .. لأننا قمنا بارجاع عدد العناصر ^_^..

اذن ماذا تلاحظ الآن ..؟؟

نعم .. إن القيمة الراجعة والتي تمثل عدد العناصر ..هي العنوان الذي يجب أن يأتي بعد العنوان الحالي ..للقيمة الجديدة .يعني بعد ال ٢ سيأتي ٣ ^_^*

شاهد المثال: ---->

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

```
var x = [1, 5, 54 , 14]; // GOOD ^_^
x[x.length] = 99;
console.log(x.toString());
x[x.length] = 100;
console.log(x.toString());
```

النتائج

```
1,5,54,14,99
1,5,54,14,99,100
```

هل لاحظت كيف تم اضافة العنصرين الجديدين ؟
الأمر سهل جدا... نحن الآن نقوم بعد العناصر ..ومن ثم أخذ عددهم ووضع القمية ك key ..

ملاحظة: في المثال استخدمت خاصية ال toString .. وهنا يكون عملها مع المصفوفة لتحويلها من مصفوفة الى نص من نوع string..بالنسبة لي كان لتسهل طباعة المصفوفة مباشرة *_^
.. لكن قد نستخدمها لأمور أخرى كثيرة ^_^

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

الطريقة ٢: يمكن باستخدام هذه الطريقة اضافة عنصر الى مصفوفة مباشرة عن طريق اختيار رقم key معين ...مثال:

```
// 2
var x = [1, 5, 54 , 14];
console.log("x.join('^_^') = " + x.join(' ^_^ '));
x[10] = 10;
console.log("x[10] = " + x[10]);
console.log("x.join('^_^') = " + x.join(' ^_^ '));
```

الإضافة

لاحظ الإضافة ..
لقد قمت بتحديد العنوان الخاص بالقيمة الجديدة بشكل يدوي..
فكان العنوان العاشر وبقيمة تساوي 10...

لاكن ..هناك مشكلة ..

المشكلة تكمن في أن هذه الطريقة ستقوم بانشاء أو ترك العناوين التي قبلها فارغة ..وهذا أمر سيء جدا .. لذلك انتبه ..

بما أننا قمنا بانشاء العنصر الجديد في الموقع العاشر ..فإنه من الموقع الرابع لغاية الموقع التاسع ..سيكون undefined ...

ملاحظة: ال join تعمل نفس عمل ال toString ..لكن تختلف عنها ..بأنك تستطيع تحديد نوع الفاصل الذي تحتاجه ...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

الطريقة ٣ (باستخدام ال push): هذه الطريقة جميلة جدا لإضافة عناصر جديدة لمصفوفة معينة ...

ال push هي دالة في الجافا سكربت يمكن استخدامها مع المصفوفات لإضافة عنصر جديد لآخر المصفوفة .. وناتج الإرجاع(return) الخاص بهذه المصفوفة هو طول المصفوفة الجديدة ^_*

شاهد المثال:

لاحظ سهولة استخدام هذه الدالة...

```
var x = [1, 5, 54 , 14];
x.push("anees using push method");
x.push(99);
console.log("x after push" + x.toString());
// سيقوم بارجاع 7 .. وهو عدد العناصر بالمصفوفة الجديدة
console.log(x.push("This is Return Push"));
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

الطريقة ٤ (باستخدام ال unshift): تقوم هذه الدالة على اضافة عنصر للمصفوفة ..ولكن بالموقع الأول لها ..ومن ثم تقوم بإزاحة كافة العناصر خطوة الى الأمام ..يعني صحاب الموقع الثاني ..بصير ثالث .. وصاحب الموقع الثالث بصير رابع ...الخ

ناتج الإرجاع لهذه الدالة ..يكون الطول الجديد للمصفوفة أيضا .. (اذا هذه الدالة مشابه لل push ..باستثناء مكان اضافة العنصر ^_^).

شاهد المثال:

```
var x = [1, 5, 54 , 14];
x.unshift("anees using unshift method");
x.unshift(99);
console.log(x.toString());
// سيقوم بارجاع 7 .. وهو عدد العناصر بالمصفوفة الجديدة
console.log(x.unshift("This is Return unshift"));
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

الطريقة ٥ (باستخدام ال splice): هذه الدالة رائعة جدا ..ومن مزايها ..أننا نستطيع أن نقوم بالإضافة والحذف بذات الوقت ..أيضا بإمكاننا تحديد الموقع المراد اضافة العنصر اليه ... إنه أمر جميل ..

لهذه الدالة 4 parameters وهي كالتالي:

arrayName. splice(I, R, V);

وأقصد بهذه الإختصارات ما يلي:

I: وهو الموقع المراد بدء الإضافة من عنده ..مثلا ٣ ..يعني ابدأ من الموقع ٣..

R: وأقصد بها عدد العناصر المراد حذفها والتي تأتي بعد اضافة العناصر باستخدام هذه الدالة ..يعني ٢ ..فسيقوم بحذف عنصرين من المصفوفة وهما العنصرين الذان يأتيان بعد ما سيتم اضافته على المصفوفة (ستتضح بالمثال)

V: وأقصد به الفيمة/ الفيم المراد اضافتها ....

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

شاهد المثال الآن:

```
var x = [1, 5, 54 , 14];
x.splice(1, 0, "anees");
console.log(x.toString());
x.splice(2, 2, "taher", "saed");
console.log(x.toString());
x.splice(0, 1, "ahmad", "muath", "Rashad");
console.log(x.toString());
```

هل يمكنك توقع النتائج؟

في أول سطر (اللون الأصفر): فإنني قلت له أضف anees الى المصفوفة في العنوان الأول.. ولا تقم بحذف أي من العناصر ..فيصبح شكل المصفوفة كالتالي:

```
1,anees,5,54,14
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

في السطر الثاني صاحب اللون النهدي:

فإنني قلتا له اذهب الى الموقع الثاني .. احذف عنصرين وهما الموقع الثاني والثالث..ومن ثم أضف القيم الجديدة وهي taher و saed ..فيصبح شكل النتائج هو :

```
1,anees,taher,saed,14
```

لاحظ أنه تم حذف الرقم ٥ والرقم ٥٤ ..من المصفوفة %_%

أما في السطر الثالث وهو اللون الفستقي الفاتح(ما الي على الألوان :P): فإنني قلت له قم باضافة القيم ahmad, muath, rashad الى المصفوفة من الموقع ٠ .. وقم بحذف عنصر واحد من المصفوفة ..فيكون المحذوف الموقع ٠ وهو الرقم ١...

تفكر ...

شر البلاد بلاد لا صديق بها ** وشر ما يكسب الإنسان ما يصم

المتنبي في مدح الصديق

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

والآن ..كما تكلمنا عن خطوات الإضافة الى مصفوفة ..سنتحدث عن خطوات أو طرق الحذف ..

^ _____ *

لكن بما أننا فهمنا الإضافة ..فإن الحذف أمر سهل..

طرق الحذف:

١)pop() : قم بحذف آخر عنصر

٢) shift(): قم بحذف أول عنصر

٣) splice(): وهنا نكتفي بوضع ال Parameters التالية I و R ^_^ *
يعني بحدد الموقع ..وعدد عناصر الحذف من هذا الموقع ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

شاهد الأمثلة:

```
var x = [1, 5, 54 , 14, 12, 125, 5124 , 11234];
console.log("Array: " + x.toString());
x.splice(6, 2);
console.log("x.splice(6, 2): " + x.toString());
x.pop();
console.log("x.pop();: " + x.toString());
x.shift();
console.log("x.shift();: " + x.toString());
```

لاحظ أن ال splice ..قامت بحذف آخر رقمين.. (احذف من الموقع السادس رقمين) .. وال pop قامت بحذف آخر عنصر وال shift قامت بحذف أول عنصر.. لتخرج النتائج بالشكل التالي:

```
Array: 1,5,54,14,12,125,5124,11234
x.splice(6, 2): 1,5,54,14,12,125
x.pop();: 1,5,54,14,12
x.shift();: 5,54,14,12
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

والآن ..برأيك .. هل بقي شيء ؟

^___* ..لسا في ^_* مفاجئات حلوة ^_^

الآن سنتكلم عن تعديل قيمة موجودة ^_^.. برأيك ..كيف يمكنك تعديل قيمة موجودة ؟

لتعديل أي قيمة مخزنة داخل مصفوفة.. نقوم باستخدام ال key الخاص بها ونسند لها القيمة الجديدة ..شاهد المثال:..

```
var x = [1, 5, 54 , 14];
x[0] = "anees";
x[1] = "Hikmat";
x[3] = "^_^";
console.log(x.join(" "));
```

```
anees Hikmat 54 ^_^
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

الآن .. بعدما قمنا بالحديث عن طرق اضافة وحذف وتعديل عناصر المصفوفة ..ما رأيك أن نرى ..كيف يمكننى ترتيب عناصر المصفوفة، وكيف يمكنني طباعة المصفوفة بشكل عكسي ^_^، وكيف يمكنني طباعة أكبر قيمة وأصغر قيمة موجودة داخل المصفوفة.وكيف يمكنني دمج مصفوفتين أو أكثر معا ^_^. بالإضافة الى ذلك ..كيف يمكنني أخذ جزء من مصفوفة ووضعها في متغير آخر لتصبح مصفوفة أخرى؟!!

^_^ ..لقد أعجبتك المواضيع ^_^ ..اذا هيا لنرى كيف يمكننا ذلك ^_^:

١) reverse()
٢) sort()
٣) concat()
٤) slice()

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

١) ال reverse: تقوم هذه الدالة على طباعة عناصر المصفوفة بشكل عكسي، هذه الدالة لا تستقبل أي parameters ..

شاهد المثال:

```
var x = [1, 5, 54 , 14];
var y = ["anees", "hikmat", "anees", "abu-hmaid"];
console.log(x.reverse().toString());
console.log(y.reverse().toString());
```

ماذا ستكون النتائج برأيك ^_^

```
14,54,5,1
abu-hmaid,anees,hikmat,anees
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

٢) ال sort: هي دالة تستخدم لترتيب المصفوفة، والترتيب إما أن يكون هجائي أو رقمي، ويمكن أن يكون تنازلي أو تصاعدي ^_^..

ملاحظات:

أ) الحالة الإفتراضية للترتيب هي هجائي وتصاعدي.

ب) لتحديد الترتيب على الشكل الرقمي ..فإننا نقوم بارسال function على أنه arguments باستخدام ال sort ..(ستتضح بالمثال).

ج) اذا قمت بوضع الرقم ٤٠ و ٥ بالترتيب الهجائي ..فإن ال ٤٠ قبل الخمسة ..

والآن الى الأمثلة ^_^

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

مثال أ) ترتيب الهجائي(تصاعدي وتنازلي):

```
var y = ["anees", "hikmat", "anees", "abu-hmaid"];
console.log(y.sort().toString());
console.log(y.sort().reverse().toString());
```

تصاعدي

تنازلي

هل رأيت القكرة البرمجية التي قمت باستخدامها ؟!..

إنك كمبرمج ..يهمك كيفية التلاعب بالدوال ..حتى تنفذ ما تريد ..ليس المهم أن أحفظ .. لكن أن أفهم ..وأستدعي وأستخدم الخصائص كما أريد عند احتياجي لها ^_^

```
abu-hmaid,anees,anees,hikmat
hikmat,anees,anees,abu-hmaid
```

تصاعدي

تنازلي

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

مثال ب) التعامل مع الأرقام(تنازليا وصاعديا):

قلنا للتعامل مع الأرقام سنضطر الى ارسال function باستخدام ال sort ..

شاهد المثال:

```
var x = [1, 5, 54 , 14];
console.log(x.sort(function(a, b){return a-b}).toString());
console.log(x.sort(function(a, b){return a+b}).toString());
```

تصاعدي

تنازلي

هل لاحظت .. إن عملية الطرح تقوم على الترتيب التصاعدي، والجمع على الترتيب التنازلي..

```
1,5,14,54
54,14,5,1
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

مثال ج: ..لو طلبت منك الآن ..طباعة أكبر رقم موجود داخل مصفوفة معينة ..؟ أو طباعة أقل رقم موجود داخل مصفوفة معينة ؟

كيف ستقوم بالحل ؟

أرجوا أن تقوم بحل هذا السؤال ..ومن ثم النظر الى الشريحة التالية .

أريدك أن تقوم بترتيب أفكارك واستغلال ما تعلمنا حتى نؤدي هذه الفكرة اليسيطة .. ^_*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

حل السؤال بكل بساطة ..يكون عن طريق ترتيب المصفوفة تصاعديا أو تنازليا
..وأول قيمة في المصفوفة هي أكبر أو أصغر قيمة حسب الترتيب ..

*

^

```
var z = [25, 50, 13 , 14];
z = z.sort(function(a, b){return a-b});// Sort Array ASC
console.log(z.toString());
console.log("Lowest Number is: " + z[0]);

z = z.sort(function(a, b){return a+b});// Sort Array DESC
console.log(z.toString());
console.log("Largest Number is: " + z[0]);
```

```
13,14,25,50
Lowest Number is: 13
50,25,14,13
Largest Number is: 50
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

٣) ()concat: تستخدم هذه الخاصية لدمج أكثر من مصفوفة معا ..والناتج يكون مصفوفة جديدة ..

ويمكن أن تستقبل هذه الدالة مصفوفة أو أكثر لعملية الدمج ...

شاهد المثال:

```
var newArray = x.concat(y);
console.log(newArray.toString());
var newArray2 = x.concat(z, newArray);
console.log(newArray2.join(" / "));
```

لاحظ هنا أنه تتم عملية الدمج بين المصفوفة x والمصفوفة y ..

ولاحظ أن عملية دمج أخرى تتم بين المصفوفة x والمصفوفة z و newArray...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

٤)()slice: تقوم هذه الدالة على اقتطاع جزء من المصفوفة وارجاعه على شكل مصفوفة جديدة..

الصيغة العامة: arrayName.slice(start, end);

ال start: هي نقطة البداية ..وهي قيمة عددية صحيحة اجبارية.

ال end: هي نقطة النهاية ..وهي قيمة عددية صحيحة اختيارية... في حال لم توضع هذه القيمة ..يكون مقدار القطع من نقطة ال start الى النهاية ..

ملاحظة: النهاية تكون القيمة -١ ... مثلا لو قمت بوضع arr.slice(2, 5); فإن القطع سيكون العنصر ٢ + ٣ + ٤ .. والخامس غير داخل ^_*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Arrays

شاهد المثال:

```
var z = [25, 50, 13 , 14, "anees", "hikmat", 1991];
var newArray = z.slice(3, 6);
console.log(newArray.toString());
var newArray2 = z.slice(4);
console.log(newArray2.toString());
```

```
**********
14,anees,hikmat
anees,hikmat,1991
```

والآن .. وقبل أن ننتقل لموضوع جديد .. لو قمنا بوضع المواقع بالسالب ..ماذا سيحدث ^_^... جرب ..وانتبه للناتج ..هذه الوظيفة البسيطة لك .. وهذا مثال يمكنك تنفيذه للتلاحظ النتائج ...

```
var newArray3 = z.slice(-4, -2);
console.log(newArray3.toString());
```

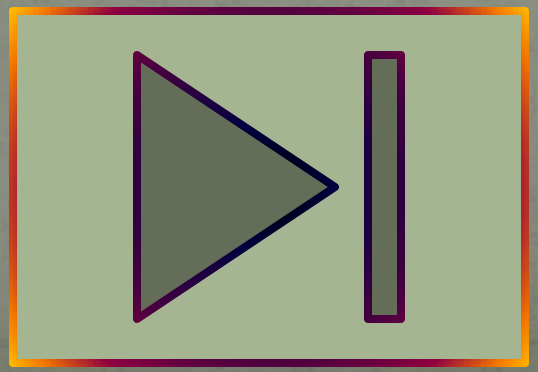
أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Arrays

والآن ..نكون بحمد الله تعالى ..انتهينا من المصفوفات ...موضوع سهل جدا ..لكن يلزمك تنفيذ الأفكار ..اذا لم بتنفيذ الأفكار .. فأنت لم تكتسب شيئا ..

لأن المعرفة إن لم ترتبط بالممارسة ..لن تصبح علم وخبرة ...

والآن ...لنشاهد جميع الأمثلة التي ذكرناها معا ^_^ (حاولت أن لا أضع أي تعليق بالشيفرة البرمجية الخاصة بالمثال ..لكي تقوم بتتبع النتائج ..وتختبر قدراتك ... ^_^*



تفكر ...

عن عبد اللَّهِ بن عمرو بن العاص رضي اللَّه عنهما قال : لم يكن رسولُ اللَّه صَلّى اللهُ عَلَيْهِ وسَلَّم فَاحِشاً ولا مُتَفَحِّشاً . وكانَ يَقُولُ : « إِنَّ مِن خِيارِكُم أَحْسَنَكُم أَخْلاقاً »

أنيس حكمت أبوحميد aneeshikmat@gmail.com

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Type Conversion

الآن ..بعد حديثنا المطول عن أنواع البيانات ..وتفاصيلها والتعامل معها ..نريد أن ننتقل الى جزء آخر .. وهو التعامل مع أنواع البيانات المختلفة معا ..

وأقصد بذلك ..ماذا سيكون ناتج جمع نص ورقم؟ ماذا سيكون ناتج طرح رقم من ؟ كيف يمكنني جعل الأرقام كالنصوص؟ ...الخ

هذا ما يطلق عليه ب Type Conversion وهي عملية تحويل نوع من أنواع البيانات ..الى نوع آخر ويكون هذا التحويل بشكلين ^_^

١) إما أن يكون التحويل تلقائي ..
٢) وإما أن يكون التحويل باستخدام ال JS Function ^_^*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Type Conversion

١) التحويل التلقائي:

- عملية جمع رقم مع null يكون الناتج هو الرقم ..وال null يتم تحويلها الى ٠
- عملية جمع نص مع null يتم تحويل ال null الى نص ..فتصبح بالشكل التالي "null" .. وبهذا يكون الناتج دمج النص الأول مع الثاني "anull"
- عملية جمع نص مع رقم ..يتم تحويل الرقم الى نص (١ الى "١")، ويتم دمج النص والرقم معا "a١"
- عملية طرح رقم من نص.. يتم تحويل النص الى رقم ..في حال كان النص مثل هذا "١" .. فيصبح الشكل 9 = "1" – 10 .. أما اذا كان هذا الشكل ١٠ – "a١" = NaN ..ليس برقم.

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Type Conversion

- عملية طباعة أي object ..يتم مباشرة تحويلها الى نص باستخدام ال toString .. وتكون المخرجات بالكشل التالي:
أ) تخزين object داخل متغير ومحاولة طباعته مباشرة ..الناتج سيكون "[object Object]" ..وهذه الرسالة كثير ما تظهر، ويكون عن طريق الخطا ..بدلا من أن يستدعي القيمة الخاصة بال object ..يقوم بوضع ال object نفسه ... لذلك ان رأيت هذا ..^_^.. اعرف وين تروح ^_^

ب) اذا كانت مصفوفة المراد طباتها.. وقمت بوضع المصفوفة فإن الناتج الراجع هو تحويل ال toString ..[1,2.3] تصبح 1,2,3 ... ^_^* شفناها كثير بدروس المصفوفات ^_^

ج) عملية طباعة ال new date() مباشرة ..يتم تحويلها الى string أيضا ويكون الناتج هو (التاريخ كامل –راجع درس ال date)

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Type Conversion

- الأرقام والجمل المنطقية يتم تحويلها الى نص أيضا ..فيصبح شكلها كالتالي:

أ) true تصبح "true"

ب) false تصبح "false"

ج) ١٥ تصبح "١٥"

```
document.getElementById("d1").innerHTML = {age:24};
document.getElementById("d2").innerHTML = [1,2,3];
document.getElementById("d3").innerHTML = new Date();
document.getElementById("d4").innerHTML = 10 + null;
document.getElementById("d5").innerHTML = "10" + null;
document.getElementById("d6").innerHTML = "10" + 1;
document.getElementById("d7").innerHTML = "10" - 1;
document.getElementById("d8").innerHTML = false; // Thi
document.getElementById("d9").innerHTML = 10;// this is
```

```
[object Object]
1,2,3
Mon Sep 29 2014 11:24:
10
10null
101
9
false
10
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Type Conversion

٢) التحويل باستخدام Function:

- التحويل من رقم الى نص ويكون ذلك بطريقتين الأولى استخدام Global method وهي String والثانية باستخدام toString ..
مثل: String(5) أو 5.toString() ..
- التحويل من منطق الى نص ..مثل تحويل false الى "false" ويكون هذا عن طريق استخدام ال String وال toString أيضا
- تحويل التاريخ الى نص ويكون أيضا باستخدام الدالتين السابقتان من الدالة Date() مثل String(Date()) أو Date().toString()

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Type Conversion

- تحويل نص الى رقم يمكنك ذلك عن طريق استخدام ال Number أو عن طريق parseInt أو parseFloat ..الخ من دوال الحساب التي ذكرناها ^ *

ملاحظة: تحويل نص فارغ الى رقم يكون الناتج صفر، وتحويل أي رقم عشري مثل ٢,١٣ يكون هو نفسه لكن بنوع رقمي ٢,١٣ ... والنص الرقمي فقط يحول الى رقم أيضا مثل "٥" الى ٥... وغير هيك يحول الى NaN ...

- تحويل المنطق الى رقم يكون باستخدام الدالة Number
- لتحويل التاريخ الى رقم نستخدم أيضا Number ..(ناتج هذه العملية هو نفسه ناتج ال (getTime ...

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Type Conversion

والآن لنشاهد مثال ^_^: (جرب ونفذ):

```
document.getElementById("d10").innerHTML = String(5);
document.getElementById("d11").innerHTML = (12 + 13).toString();//
document.getElementById("d12").innerHTML = String(false);
document.getElementById("d13").innerHTML = Date().toString();
document.getElementById("d14").innerHTML = Number("5") + 5;
document.getElementById("d15").innerHTML = Number("5 5");
document.getElementById("d16").innerHTML = Number("5anees") + 5;
document.getElementById("d17").innerHTML = Number("");
document.getElementById("d18").innerHTML = Number("3.14") + 5;
document.getElementById("d19").innerHTML = Number(false);
document.getElementById("d20").innerHTML = Number(true);
document.getElementById("d21").innerHTML = Number(new Date());
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Type Conversion

والآن ..ما رأيك ببعض الأسئلة التي تدعونا الى التفكير قليلا ..^_^*

أسئلة أنت ستجيبها (Tricks):

```
Number(true) + 5;
Number(true) + "anees";
Number("") - 20;
Number("5") + String(10);
Number(null) - String(3.14);
```

حاول توقع الإجابة ..ثم نفذ الشيفرة البرمجية ..وانظر النتائج ^_^

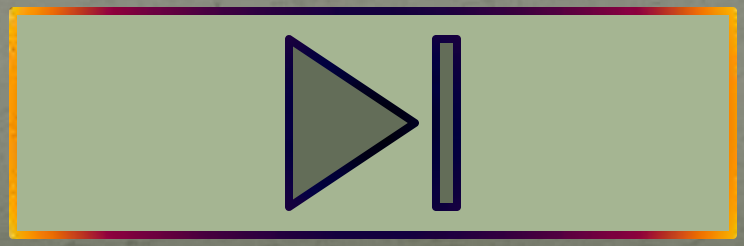
aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Type Conversion

والآن ..أصدقائي ..نكون انتهينا أيضا من هذا الموضوع المهم ...

وهو مهم جدا ..لتستطيع معرفة النتائج الممكنة والمشاكل التي يمكن أن تحصل بالإضافة الى التحكم بالمخرجات ...كل ذلك يكون بمعرفة انواع البيانات.. وناتج العلاقات التي بينها ...

والآن ..لنشاهد جميع الأمثلة ...(أيضا لم أقم بالتعليق على الشيفرة البرمجية تتبع الكود .. ثم نفذ..وعد واسترجع اذا أخطأت ..^_*)..



تفكر ...

الإدريسي (عالم مسلم)

أحد كبار الجغرافيين في التاريخ ومؤسسي علم الجغرافيا، كما أنه كتب في التاريخ والأدب والشعر والنبات ودرس الفلسفة والطب والنجوم في قرطبة.

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Conditions[if—else]

الجمل الشريطة ^_* .. تستخدم الجمل الشرطية لتحديد سلوك البرنامج بنائا على محددات أو معطيات معينة(شرط)، ويلعب دورا مهما جدا جدا في برمجة الصفحات الإلكترونية ..وفي جميع لغات البرمجة ..

الصيغة العامة

```
if (condition1) {
    // Any Code
}else if (condition2) {
    // Any Code
}else if (condition3) {
    // Any Code
}else if (condition4) {
    // Any Code
}else {
    // Any Code
}
```

هنا تعني نفذ الشيفرة البرمجية اذا تحقق الشرط

هنا تعني ..اذا لم يتم تنفيذ الشرط الأول ..وأحتاج الى التحقق من شرط آخر ..أقوم باستخدام هذا الأسلوب.. وهو ليس الزامي

في حال لم يتم تنفيذ الشرط أو الشروط التي قبلها ..يتم تنفيذها ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[if—else]

الآن ..لنقوم بشرح الصيغة السابقة:

١) if: تستخدم ال if لوضع شيفرة برمجية يتم تنفيذها في حال كان الشرط ناتجه هو true ..

٢) else: تستخدم ال else لتنفيذ الشيفرة البرمجية في حال لم يتحقق الشرط / الشروط التي تسبقه.(أي عندما تكون المرجعات من ال if / else if هي false... ^_^*

٣) else if: تستخدم هذه الجملة لتنفيذ شيفرة برمجية بنائا على شرط آخر اذا لم يتم تنفيذ الشرط الموجود بال if التي قبلها ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[if—else]

ملاحظات:

١) else - if هي كلمات محجوزة داخل الجافا سكربت.

٢) تكتب ال if وال else ..على شكل lowercase ..والا سيظهر لك خطا..

٣) يمكنك كتابة عدد غير محدد من ال else if ...

٤) حيز تنفيذ ال if يكون اذا كان نااتج الشرط true

٥) حيز تنفيز ال else يكون اذا كان ناتج الشرط/ الشروط التي سبقتها جميعها false.

٦) جيز التفيذ لل else if يكون اذا لم يتحقق الشرط / الشروط الذي قبلها(false) وكان ناتج الشرط الخاص بها هو true..

٧) يكون تنفيذ الشرط / الشروط بشكل متسلسل وصولا الى ال else وهي آخر مرحلة .. في حال نفذ أحد الشروط..لا يكمل الباقي ..وإن لم يتحقق شيء ..ولم يوجد else ..لن يقوم بتنفيذ شيء..وسيكمل عمل البرنامج بالشكل الطبيعي

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[if—else]

مثال ١(طبق ..لا تنسى التنفيذ):

```
var x = 7, y = 7;

// Example 1
if(x == 7){// if x equal 7
    document.getElementById("d1").innerHTML = "if(x == 7){ " + x + " } ";
}
```

سيتم تنفيذ الشرط مباشرة ..لأن ال
x = 7 ^_^..

ملاحظة مهمة: اذا قمت بوضع x = 7 داخل الشرط ..فإن الشرط سيتنفذ دائما
..انتبه أننا نقارن باستخدام ال == ..أما ال = فهي تستخدم لإسناد القية ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[if—else]

مثال ٢) قيم x و y موجودة في المثال ١..

```
// example 2
if(x != y){// if x not equal y
    document.getElementById("d2").innerHTML = "if(x == 7){ " + x + " } ";
}else{ // else (here, thats mean if x == y)
    document.getElementById("d2").innerHTML = "if(x != y){ }else { y = " + y + ", x = " + x + " } ";
}
```

لن يتم تنفيذها لأن x = y

سيتم تنفيذ ال else

اذا لاحظت ..فإن ال else هي من تم تنفيذها لأنه لم يتحقق الشرط الذي قبله ..

وكملاحظة ..اذا أردت قرائة وكتابة الشروط ..فحاول قرائة ال else بشكل آخر.؟..ماذا ستكون قرائتك لها ؟

ال else في هذا المثال تعني if(x == y) ^__^*

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[if—else]

مثال ٣) قيم x و y موجودة في المثال ١ ..

```
// example 3
if(x == 13){// if x equal 13
    document.getElementById("d3").innerHTML = "IF";
}else if (x == y) {// else if x equal y
    document.getElementById("d3").innerHTML = "(else if) x = " + x;
}else{// else (here, thats mean if x not equal 13, and x not equal y
    document.getElementById("d3").innerHTML = "Else";
}
```

هنا سيقوم بتنفيذ ال else if لأن الشرط الأول لن يتحقق ...

هل لاحظت .. قيمة ال x لا تساوي ١٣ ..لذلك ..ذهب فوجد شرط آخر (ويعني اذا لم تكن x تساوي ١٣ ..فتحقق هل ال x تساوي y) ..وكانت الإجابة نعم ... وبهذا ..يتم تنفيذ الشرط ..ويكمل سير البرنامج.. دون الدخول الى ال else ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[if—else]

الآن ..ما رأيك أن تتبع تنفيذ ال function (الموجود الصورة) على فرض أنه تم استدعاء هذا ال function ٤ مرات ..ماذا سيكون الناتج ..

```
// Example 4
function conExample() {
    if (x <= y) {
        document.getElementById("d4").innerHTML = "(if) x = "
                + x + " This result if (x <= y) " + x + " , " + y; ;
    }else if (x == 14) {
        document.getElementById("d4").innerHTML = "(else if) y = "
                + y + " This result if (x == 14) " + x + " , " + y;
    }else if (x % y == 0 && y * 4 > x) {
        document.getElementById("d4").innerHTML = "(else if (x % y == 0 && y
    }else {
        document.getElementById("d4").innerHTML = "(else) x + y = " + (x + y)
    }

    x += 7;
}
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

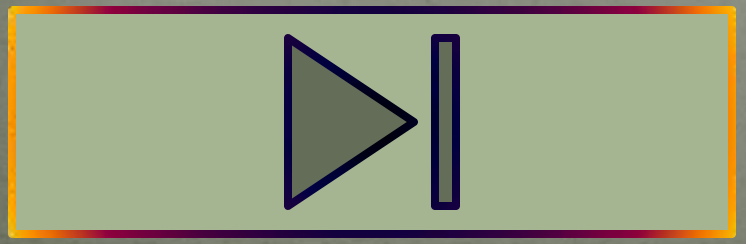
JavaScript Conditions[if—else]

والآن ...هل استطعت تتبع الحل .. المهم لدي في الدرجة الأولى هو معرفة ما سيتم تنفيذه مع كل عملية استدعاء ... واذا عرفت قيمة كل متغير بكل مرحلة .. فأنت بالتأكيد ستسطيع الإجابة ..

مثال سهل ..ورائع للبدء بالتقدم في المستوى ^_*

والآن ..لنشاهد الأمثلة مع حل السؤال السابق ^_^...

أرجوا أن تكون اهتممت بالتطبيق ...^_*



تفكر ...

حدثنا يحيى بن بكير حدثنا الليث عن عقيل عن ابن شهاب أن سالما أخبره أن عبد الله بن عمر رضي الله عنهما أخبره أن رسول الله صلى الله عليه وسلم قال المسلم أخو المسلم لا يظلمه ولا يسلمه ومن كان في حاجة أخيه كان الله في حاجته ومن فرج عن مسلم كربة فرج الله عنه كربة من كربات يوم القيامة ومن ستر مسلما ستره الله يوم القيامة

فتح الباري في شرح صحيح البخاري

أنيس حكمت أبوحميد aneeshikmat@gmail.com

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[Switch]

ال Switch بنفس فكرة ال if/else ..تستخدم لتنفيذ مهام معينة بنائا على تحقق شرط ..

لكن لماذا يكمن أن نستخدم ال switch؟

نقوم باستخدام ال switch في حال تواجد أكثر من شرط ونريد تنفيذ أمر واحد فقط .. (تشبه فكرة استخدام ال if else) ....لكن الفرق الذي يحصل هو بآلية التنفيذ.. بحيث

١) تحسب القيمة المراد التحقق منها مرة واحدة فقط .. لأنها تستخدم في مكان واحد وهو داخل ال switch ...

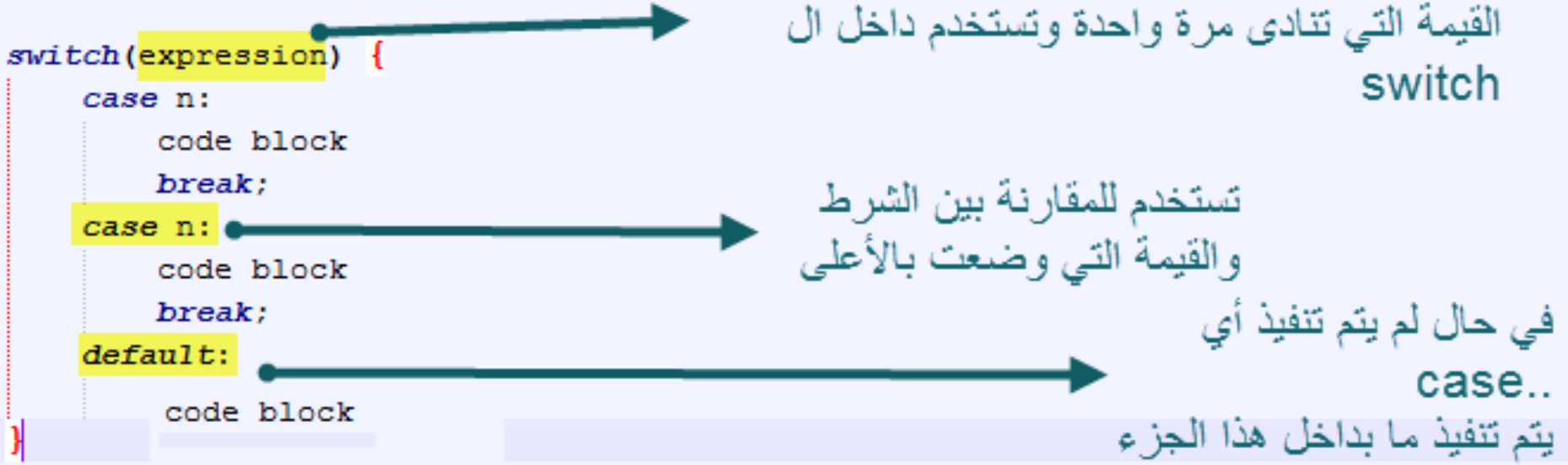
٢) القيمة التي في النقطة الأولى، يتم مقارنتها مع كل case ..(case ترمز الى حالة جديدة للشرط)...

٣) اذا ساوت القيمة الموجودة في ال case القيمة الموجودة في النقطة الأولى ..فإن الشرط ينفذ..

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Conditions[Switch]

الصيغة العامة لل switch هي:



هل لاحظت هذه الدالة ...إنها جميلة جدا أثناء التعامل مع الشروط المحددة على فرض طباعة أيام الأسبوع ..أو أسماء الأشهر ...الخ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[Switch]

مثال ١ :

```
// Example 1
var Item = 5;
switch(Item){
    case 1:
        document.getElementById("d1").innerHTML = "1";
        break;
    case 2:
        document.getElementById("d1").innerHTML = "2";
        break;
    default:
        document.getElementById("d1").innerHTML = "default";
}
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[Switch]

ا

لآن ..اذا لاحظت ..فإنني بعد كل جملة case أقوم بوضع break ..فما وظيفتها برأيك؟

تقوم ال break على الخروج من ال switch (تخرج من ال block التي تنفذ به)... ولذلك ..مجرد ما جدنا القيمة ..لا يوجد هناك داع للاستمرار في فحص العناصر ...^__*... فأقول له توقف ^^..ولكن ماذا سيحصل اذا قمت بتركه يستمر ؟! ...بكل بساطة ^_* ..سيقوم بتنفيذ جميع الأسطر التي بعده حتى يتوقف عند break آخرى .. أو defualt ...أو انهى تنفيذ جميع ال case..!!

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[Switch]

أما بخوص الكلمة الأخرى فهي default.. وكما تلاحظ ..فهي آخر ما ينفذ حسب التسلسل ..وهي التي تقوم بتنفيذ كود معين اذا لم يتحقق أي شرط ..أيضا اذا لاحظت ..فإننا لم نضع break بعدها ..لأنها آخر عنصر ..فلا داعي لذلك ^_^

مثال ٢:

```
// Example 1
var Item = 1;
switch(Item){
    case 1:
        document.getElementById("d2").innerHTML = "1";
        console.log(1);
    case 2:
        document.getElementById("d2").innerHTML = "2";
        console.log(2);
    case 3:
        document.getElementById("d2").innerHTML = "3";
        console.log(3);
        break;
    default:
        document.getElementById("d2").innerHTML = "default";
}
```

مثال مهم ..نتائج مهمة ...انتبه للنتائج

لا يوجد break

سيتم تنفيذها ..

توقف هنا...

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Conditions[Switch]

هل لاحظت المثال السابق ؟!..

اذا نظرت الى console فستجد أنه قام بطباعة الرقم ١ و ٢ و ٣.... ومن ثم توقف..وكانت النتيجة النهائية الظاهرة على المتصفح هي ٣ .. وذلك لأنك لم توقف الإستمرارية الخاصة به ...^__*

لكن هل هذا الأمر سيء ؟ ..إنه من المفيد جدا في بعض الأحيان أن لا تستخدم ال break ؟!!!!!

نعم لا تتعجب .. والآن تتسآل ..ما هي هذه الحالة؟

لو افترضنا وجود أكثر من شرط .. هذه الشروط تصنف الى مجموعات ..كل مجموعة تشترك بنفس النتائج أو الكود الذي يسنفذ ...هنا ستحتاج هذه الحركة ^_^..شاهد المثال بالصفحة التالية:

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[Switch]

مثال ٣:

```
// Example 3
var Item = 4;
switch(Item){
    case 1, 2, 3:
        document.getElementById("d3").innerHTML = "1, 2, 3";
        break;
    case 4:
    case 5:
    case 6:
        document.getElementById("d3").innerHTML = "4, 5, 6";
        break;
    default:
        document.getElementById("d3").innerHTML = "default";
}
```

لاحظ طريقة التنفيذ ..الشرطالأول يعني ١ أو ٢ أو ٣ .. ونفس الشيء الشرط الثاني ...والتوقف يكون عند ال break ..لذلك يكون ناتج تنفيذ الشفرة للمجموعة كاملة نفس الشيفرة البرمجية...

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[Switch]

والآن ..ماذا .. هل جمل الشرط فقط ذات تنفيذ مباشر ..يعني إما أن تساوي ١ أو تساوي ٢ وهكذا !!...

الجواب طبعا لا ..بل يمكنك استخدام العلاقات المنطقية أيضا ..

مثال ٤)

```
// Example 4
var Item = 4;
switch(true){
    case Item > 5 && Item != 7:
        document.getElementById("d5").innerHTML = "> 5";
        break;
    case Item <= 5 && Item == 4:
        document.getElementById("d5").innerHTML = "Item <= 5";
        break;
    default:
        document.getElementById("d5").innerHTML = "default";
}
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Conditions[Switch]

والآن ..سؤال موجه لك ^_____*

في الموضوع السابق if—else ..قمنا بتنفيذ مثال ..كلما قمت بالضغط على الزر الموجود بالصفحة ..اختلفت القيمة حسب الشرط..

الآن أنا أريدك أن تقوم بنفس الفكرة ..لكن مع ضغطة زر يقوم بطباعة اسم اليوم مثل سبت ..اذا ضغط ..بصير أحد ..اذا ضغط كمان مرة ..بصير اثنين ..وهكذا .. واذا وصل الى الجمعة ..برجع بعيد الدورة من السبت ..

سؤال بسيط ..(استخدم ال switch) .... بعد الإنتهاء من حل السؤال ..انتقل للشريحة التالية ^_^

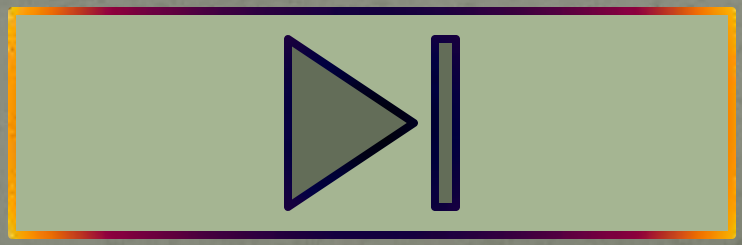
أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Conditions[Switch]

والآن ..هل قمت بحل المثال ؟

طبعا لحل هذا المثال أكثر من أسلوب وطريقة .. وستحد حل هذا المثال ..مع جميع الأمثلة التي تم ذكرها هنا بإذن الله تعالى ..

لا تنسى أن التطبيق هو المهم .. اذا لم تطبق ..فأنت لم تحصل على أكبر قدر ممكن من المعرفة ...



تفكر ...

اللهم اغفر لي ولوالدي وارحمهما كما ربياني صغيرا

اللهم إني أسألك برحمتك يا رحيم يا عزيز يا غفور
أن تغفر لي ولأبي وأمي ولجميع المسلمين أحيائهم وأمواتهم
إنك سميع عليم رحيم رزاق كريم

الحمد لله رب العالمين

أنيس حكمت أبوحميد aneeshikmat@gmail.com

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Loop

بعد حديث طويل ..حول مفاهيم الجافا سكربت ..بدئنا ننتقل ..خطوة بخطوة نحو الطريق لتنفيذ وتفعيل مجموعة الخصائص التي تعلمناخا ..

ونحن الآن بصدد شرح موضوع مهم جدا في عالم البرمجة ..وهو مفهوم الدوران أو التكرار بالجافا سكربت..(Loop)

لنقوم بعملية التكرار لتنفيذ أمر معين (loop) ..علينا معرفة ما هو الامر المناسب لذلك ..والآن ..طرق عمل loop داخل الجافا سكربت:

١) for loop
٢) while loop
٣) do while loop
٤) for/in loop

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

١) for: تستخدم ال forلتنفيذ دوران بعدد معين من المرات، تحدد بدايته ونهايته منذ تعريف ال for، وينتهي تنفيذها عند عدم تحقق شرط الإنتهاء ..

الصيغة العامة:

```
for(Start; Condtion; changeStartValue) {
    // Any code
}
```

لاحظ .. أن for هي كلمة محجوزة، والآن ..لنبدأ بشرح مكوناتها ..

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Loop

أ) start: ويقصد بها القيمة التي سيدأ من عندها الدوران.

ب) Condition: ويقصد بها الشرط اللازم ليستمر عنده الدوران.

ج) changeStartValue: ويقصد بها كود ينفذ لتغيير القيمة الإبتدائية حتى يتم عدم تنفيذ الشرط ..في النهاية ..

والآن لنشاهد مثالا على ذلك ـمثال ١:

```
for(var i = 1; i <= 5 ; i++){
    console.log(i);
}
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Loop

طريقة عمل المثال السابق:

أولا يقوم بتنفيذ ال start .. وهي المتغير المسمى i والذي قيمته الإبتدائية ٠ ..

ثانيا: نتحقق هل الشرط ينفذ أم لا ..(بمعني آخر يرجع true أم false) ... اذا أرجع الشرط true .فإنه سيتابع العمل ..واذا أرجع false ..سيخرج من جملة الدوران ،،

ثالثا: سيقوم بتنفيذ الشيفرة البرمجية الموجودة ..وهي طباعة قيمة ال i الحالية داخل ال console ..

رابعا: يذهب الى changeStartValue ..ويقوم على جمع (اضافة) i الى قيمة i الحالية ..فإذا كانت i = 0 ..فإنها ستصبح i = 0 + 1 وتساوي ١ ..

خامسا: نعود الى النقطة (ثانيا) ...وتستمر الدورة هكذا حتى نصل الى عدم تحقق الشرط.

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

مثال٢:

```
// Example 2
for(var i = 10; i > 0 ; i = i - 2){
    console.log(i);
}
```

لاحظ في هذا المثال ..كان التكرار بشرط متناقص..وقمت بتحديد مقدار التناقص ب ٢ .. والقيمة الإبتدائية بعشرة ..ولتنفيذ الكود ..وضعت شرط تحقق بأن تكون ال I أكبر من ٠ ...

النتائج:

```
10
8
6
4
2
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

مثال ٣: لقد تطرقنا في الدروس الماضية ..الى شرح المصفوفات ... ولكن اذا أردت منك طباعة عناصر مصفوفة .. مكونة من ١٠٠٠ عنصر ..هل ستقوم بكتابة arr[o] ثم arr[١] ثم arr[....n] ..؟؟؟

الجواب ..طبعا لا .. واذا قلت لك اطبع العناوين الفردية أو الزوجية في المصفوفة .. أو اجري أي عملية حسابية على المصفوفات سيكون هذا عن طريق ال loop ..^_^..

شاهد المثال:

```
// Example 3
var array1 = ["anees", "hikmat", 1991, 11, 10];
var info = '';
for(var i = 0; i < array1.length ; i++){
    info += array1[i] + " ^_^ ";
}
document.getElementById("d1").innerHTML = info;
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Loop

هل ركزت جيدا في المثال السابق؟

ملاحظات:

١) اذا لاحظت ..في تعاملنا مع المصفوفات نبدأ بالقيمة ٠ .. والسبب في ذلك أن عنونة المصفوفات تبدء من الصفر (راجع درس المصفوفات اذا لم تعرف هذه المعلومة).

٢) قمنا باستخدام احدى الدوال التي تستخدم مع المصفوفات وهي ال length ...هذه الدالة ترجع لنا عدد عناصر المصفوفة ..لذلك تم استخدام الشرط ..أقل (<) .. وبهذا يكون البدء من صفر والإنتهاء عند length -1 وكمثال اذا أرجعت الدالة أن طول المصفوفة ١٠ .. فهذا يعني أن الدوران يجب أن يبدأ من ٠ وينتهي عند ٩ ..وبهذا يكون عدد العناصر هو ١٠

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

مثال ٤:

```
// Example 4
var array1 = ["anees", "hikmat", 1991, 11, 10, 5, 6, 8, 54, 12, 88];
var info = '';
var InnerDiv = document.getElementById("d2");
for(var i = 0; i < array1.length; i++){
    if((i % 2 == 0) && i < 9){
        info += array1[i] + "<br />";
    }
}
InnerDiv.innerHTML = info;
```

anees
1991
10
6
54

لاحظ ..في هذا المثال ..قمنا بطباعة المعلومات الموجودة بالمصفوفة والتي تمثل العناوين (key) بها رقما زوجيا ..وبشرط أن تكون أقل من ٩ قيمة الi ... بإمكانك التلاعب والتحكم بالشرط ..كما تريد..

بالنسبة للمتغير InnerDiv ..فهذه طريقة يمكنها أن تختصر علينا الكثير من الوقت أثناء التنفيذ.. فأقوم بتخظين ال obj داخل متغير ..(وضعتها للفكرة)

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

٢) ننتقل الى ال while: تستخدم هذه الكلمة المحجوزة أيضا لإنشاء دورا مثل ال for.. لكن نستخدم هذا الأسلوب ..في حال أردنا تنفيذ شرط معين لمدة طويلة ..ويتوقف عندما يصبح الشرط غير صحيح ...في هذا النوع لا يشترك وضع بداية ونهاية ..ولكن يجب أن يوضع شرط يتحقق في مرحلة معينة حتى يتم الخروج من الدوران ..والا سيستمر بعدد غير نهائي..

الصيغة العامة:

```
// Example 3
while (condition) {
    // Any Code
}
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

كما تلاحظ في الصيغة .. فإنه يوجد شرط ..ثم يدخل على الدوران ..وهو بمعنى آخر if ..تنفذ أكثر من مرة ^_^.. هذا الشرط ..بمجرد أنه أصبح يرجع false .. فذلك يعني انتهاء الدوران ...^_^*

مثال١:

```
var x = 1;
while (x < 5) {
    console.log("While ^_^ = " + x);
    x++;
}
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Loop

كما تلاحظ في المثال السابق

أولا ..أولا يقوم ال while بالتحقق من الشرط..قبل الدخول وتنفيذ أي سطر برمجي ..وكان الشرط هو x < 5.. وقيمة x هي ١ ..اذن الشرط الصحيح ..فيقوم بالدخول الى الدوران .. السطر الأول سيقوم بطباعة قيمة x داخل ال console .. لكن انتبه الآن ل x++ .. فهذه هي التي ستجعل من الشرط غير صحيح... واذا قمت بتجربتها ..سيدخل في لوب غير منتهي ..حتى يظهر لك المتصفح رسالة تقول Crashed .. ^_* ...

اذن الشرط ..ثم التنفيذ الشيفرة البرمجية الموجودةة داخل الدوران... ثم الشرط مرة أخرى...

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

مثال٢:

```
// Example 6
x = 5;
while (true) {
    console.log("While true = " + x);
    if(x == 10){
        break;
    }
    x++;
}
```

انتبه لهذا المثال جيدا:

لاحظ أنني لم أحدد ..بداية ..نهاية ..شرط منطقي !! ..ومع ذلك فإن الدوران سيتوقف عند ١٠... وذلك بسبب ال break ..هل تذكر حينما ذكرناها بال switch ..هي بنفس المبدأ ..لكن هنا ..ستتجعلك تخرج خارج الدوران (توقف الدوران اجباري)..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

٣) do while: هي وسيلة دوران أيضا، لكنها تختلف عن ال for أو ال while ..بإنها تنفذ مرة واحدة على الأقل..(حتى وإن لم يتحقق الشرط ..فإنه سيقوم بتنفيذ الشيفرة الموجودة لمرة واحدة ...

الصيغة العامة:

```
// Example 7
do {
    // Code
}
while (condition);
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

هل لاحظت في الصيغة العامة ..طريقة ترتيب التنفيذ؟

ستجد أنه اولا افعل (نفذ الكود) ثم تحقق من شرط ال while .. لذلك يجب أن ينفذ هذا الكود لمرة واحدة على الأقل..

مثال ١:

```
// Example 7
var i = 0;
do {
    console.log("Ex-7:(i) = " + i);
}
while (i > 11);
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

اذا لاحظت المثال الأول ..ما تتوقع أن يكون النتائج ؟..

الجواب هو هذا ^_^

```
Ex-7:(i) = 0
```

مع أن الشرط لم يتحقق الى أنه قام بتنفيذ الدوران ^_*.

مثال٢:

```
// Example 8
var i = 0;
do {
    i++; // Start from 1
    if(i % 2 == 0){
        continue;
    }

    console.log("Ex-8:(i) = " + i);
}
while (i <= 11);
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Loop

هل لاحظت في المثال الثاني ..

لقد قمت بادخاله بدوران .. كما تعلمنا بال while التقليدية .. بالإضافة الى ذلك .. لا تنسى اضافة i++ ..(شرط الذي سيجعل الدوران يتوقف &_*) ..

* ال continue: هي keyword داخل الجافا سكربت ..تستخدم لعمل قفزة أو تجاوز في جمل الدوران ..ولهذا ..اذا قمت بتنفيذ المثال السابق ..فسيقوم بطباعة الأعداد الفردية ..فقط .. لإنني قلت له اقفز الى شرط الدوران مجددا.. ولا تقم بتنفيذ الأسطر التي تأتي بعد continue .. %_%

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

٤) for/in loop: يستخدم هذا للقيام بدوران داخل object معين ..فإذا كنت تذكر ..فإن ال object ..تكون ال key الخاصة به عبارة عن string ..وهذا يعني أنه ليس رقم .. ؟!.. فكيف يمكنني المرور على عناصره ؟

لذلك وجدت ال for/in loop ... شاهد الصيغة العامة:

```
// Example 9
var x, object = {xxx: "xxx"};
for (x in object) {
    // code
}
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Loop

لاحظ الصيغة العامة:

X: تمثل القيمة التي سيتم حفظ ال key بها ..

ال object ..هو أي متغير من نوع object ..

جملة الدوران تكون بصيغة التالية (المتغير الذي سيستخدم لحفظ ال key الحالي ثم in ثم ال object الذي سيتم أخذ ال key منه ..)

شاهد المثال:

```
var me = {fName:"anees", mName:"hikmat", age:24};
var x, res = "";
for (x in me) {
    res += x + ": " + me[x] + "\n";
}
console.log(res);
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Loop

هل لاحظت المثال السابق ..

هل قمت بتنفيذه ؟ ..يجب أن تكون النتائج بالشكل التالي ^_^:

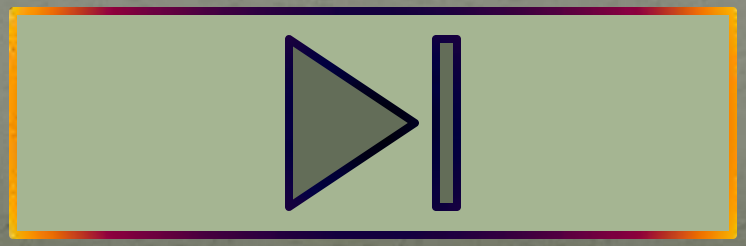
```
fName: anees
mName: hikmat
age: 24
```

ال fname وال mname و ageهي ال x

وتمثل قيم ال key ...

وال n\ هي التي قامت بتنفيذ السطر الجديد ^_^

والآن ..لنشاهد جميع الامثلة السابقة معا ..(يرجى أن تتبع الشيفرة البرمجية)



تفكر ...

*الدُموع كائِناتٌ فُضولِية كُلما حَدثَ شيءٌ مؤلِمْ خَرجَت لِتُشااهِد *! *

أنيس حكمت أبوحميد aneeshikmat@gmail.com

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Handling Error

والآن ..بإذن الله تعالى ..في آخر موضوع في هذه الدورة المتواضعة ..وهو ال handling error..

لقد تكلمنا في أول الدورة عن أهمية ال depug واستخدام ال console ..
والآن سنأتي لإسلوب حماية .. من الأخطأ التي يمكن أن تحصل أحصيتها او لم أحصها .. وذلك عن طريق استخدام ال try/catch ^_^..

أولا ..وقبل البدء بهذا الموضوع الرائع ^_*..أحب أن أخبرك ..أنه لا يوجد مبرمج مهما بلغ من احترافية ..أن يكتب برنامج بدون أخطاء .. وإن كان الخطأ ..نسيان وضع فاصلة منقوطة مثلا ^_* ... لذلك هذا أمر اعتيادي وليس عيبا ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Handling Error

والآن ..لنتكلم عن

Try وال catch وال throw وال finally ..

ال try: يوضع داخل ال try الشيفرة البرمجية التي يمكن أن يقع بها الخطأ ^_^

ال catch: ويوضع بها ماذا سيحدث اذا وقع الخطأ ..

ال throw: تستخدم لارجاع رسالة خطأ معينة .. مكتوبة من قبل المبرمج ^_^

ال finally: يوضع بها كود ينفذ بعد عملية تنفيذ ال try/catch بغض النظر إن كانت النتيجة صحيحة ..أم خاطئة ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Handling Error

الصيغة العامة:

```
try {
    // Code
}catch(errMessage) {
    // handle errors
}
// OR
try {
    // Code
}catch(errMessage) {
    // handle errors
}finally{
    // Code will execute alwyes..
}
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Handling Error

شاهد الأمثلة:

مثال ١)

```
try {
    var x = 2=2 / 3 * 5;
}catch(errMessage) {
    console.log(errMessage.message);
}
```

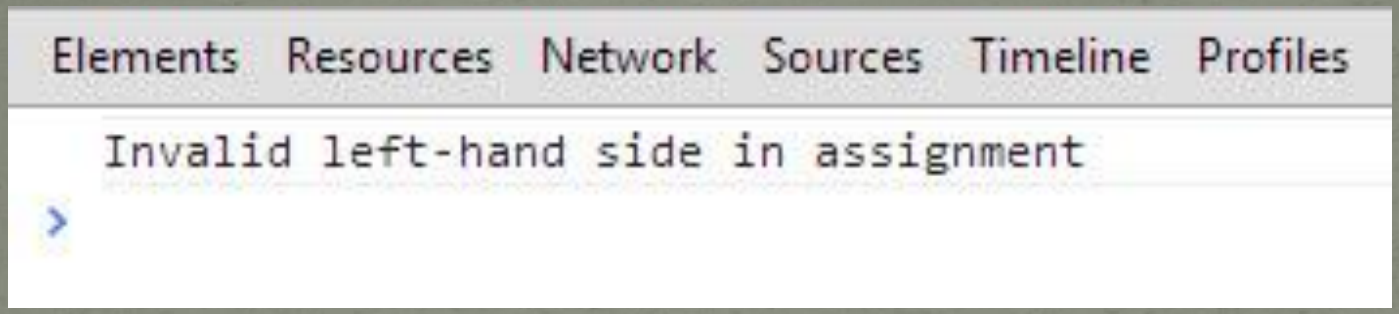
لاحظ ..لقد قمت باستخدام ال message لطباعة الخطأ الذي حصل .. لاحظ كيف تم التحكم بالأخطاء الظاهرة عن طريق ال try/catch ..

والآن ..لنرى النتائج في هذه الحالة --- (لا تنسى القيام بتنفيذ الكود الآن ..ثم الانتقال الى الموضوع الجديد)

aneeshikmat@gmail.com أنيس حكمت أبوحميد

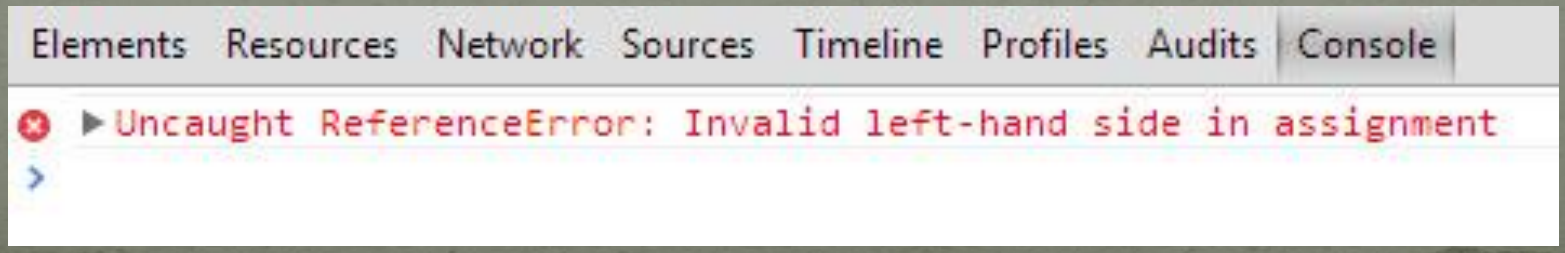
JavaScript Handling Error

النتيجة عند استخدام ال try/catch :



هذه النتيجة التي ظهرت لإنني قمت بطباعة الخطأ باستخدام ال console

والآن شاهد هذه النتيجة في **حال لم أستخدم** ال try/catch..



aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Handling Error

اذا لاحظت ..فإنه في هذه الحالة ..وفي هذا المثال ..لو لم أستخدم ال try/catch .. لتوقف تنفيذ السكربت ..بسبب وجود الخطأ..

لكن بوجود ال try/catch .. فإن العمل بالسكربت سيستمر ^_^.. وهذه نقطة مهمة جدا ^_^

والآن للنطلق الى ال throw.. هذه ال keyword الجميلة تستخدم لبناء أخطاء خاصة بنا ..يتم التعامل معها ..لنشاهد المثال معا ^_^

--- الى الشريحة التالية:

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Handling Error

مثال٢:

```
<div id="d1"></div>
    <input type="text" id="txtAge" />
    <input type="button" id="btnAge" value="Set Age" onclick="setAge();"/>
</div>
<script>
    function setAge()
    {
        try {
            var age = document.getElementById('txtAge').value;
            if(!age){
                throw "age field is null" + age;
            }else if(isNaN(age)) {
                throw "Its not a number" + age;
            }else if(age > 120){
                throw "Very OLD ??!!!" + age;
            }else if(age < 18) {
                throw "You are baby !!" + age;
            }else{
                alert("Hi, you are welcome");
            }
        }catch(errMessage) {
            console.log(errMessage);
        }
    }
```

أنيس حكمت أبوحميد aneeshikmat@gmail.com

JavaScript Handling Error

في المثال الثاني كما تلاحظ ..فإنني قمت بحصر الأخطاء التي يمكن أن تحدث أثناء ادخال العمر .. والآن لطباعة كل خطأ على حدى ..فقمت باستخدام ال throw .. ولطباعة الخطأ الذي قمنا باستخدامه ..نكتفي بطباعة ال errMessage .. ^_^*

طبعا تنفيذ هذا المثال مهم جدا ..^_^..

الآن قم بتنفيذه ..وكتابة ال input field ....هل يمكنك كتابة مثال بنفس الطريقة ؟!

هيا قم بتجربة ذلك ..للإسم ..على أن لا يكون الإسم أكبر من ٢٠ حرف ..وأن لا يحتوي أرقام ^_^*..جرب ذلك ...بعد تنفيذك للمثال السابق ..

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Handling Error

مثال٣:

```
try {
    var x = 4 * a;// a is not declared
}catch(errMessage) {
    console.log("Ex 3: " + errMessage.message);
}finally{
    console.log("Finally when is error ^_^, its executed");
}

try {
    var x = 2 * 9;
}catch(errMessage) {
    console.log("Ex 3-2: " + errMessage.message);
}finally{
    console.log("Finally when no error ^_^, its executed");
}
```

aneeshikmat@gmail.com أنيس حكمت أبوحميد

JavaScript Handling Error

والآن ..لاحظ استخدام ال finally ..

في المثال السابق سيتم تنفيذ الكود في جميع الحالات ..اذا وقع خطأ ..أو لم يقع خطأ ..

شاهد النتائج للمثال السابق ^_^

```
Ex 3: a is not defined
Finally when is error ^_^, its executed
Finally when no error ^_^, its executed
```

تفكر ...

اللهم ءاتنا في الدنيا حسنة وفي الآخرة حسنة وقنا عذاب النار

أنيس حكمت أبوحميد aneeshikmat@gmail.com

ماذا بعد

بعد قرائتك لهذه الدورة ..يمكنك الإنتقال الى ما يسمى بال JS DOM وال JS BOM ...

وسيكونان في دورة جديدة ..بإذن الله تعالى، في المستقبل القريب إن كان لنا عمر وقدر في ذلك ...

ولذلك الحين .. يمكنك البحث والقرائة عن هذان المفهومان بمستواك الحالي وبكل سهولة ان شاء الله... ^_*

أنيس حكمت أبوحميد aneeshikmat@gmail.com

الخاتمة

وهكذا لكل بداية نهاية ، ولكل جهد وعمل نتيجة .. ولكل عمل هدف ..نسأل الله تعالى أن نكون قد وفقنا في سرد المفاهيم التي تم ذكرها بإسلوب غير ممل بإسلوب قد جذبك قدر المستطاع ..

ونسأل الله تعالى أن يتقبل منا هذا العمل المتواضع ..ويغفر لنا زلاتنا، وعثراتنا .. وأن يرزقنا ويتقبل منا برحمته وكرمه ومنه وفضله .. والحمد لله رب العالمين..

اللهم اغفر لي ولوالدي وللمسلمين أحيائهم وأمواتهم إنك على كل شيء قدير ..
اللهم نسألك موجبات رحمتك ..ونعوذ بك من سخطك والنار ..
اللهم اغفر لي وارحمني وارزقني الرزق الحلال الطيب .. لي ولجميع شباب المسلمين يا رب العالمين ... اللهم صل على سيدنا محمد حتى يرضى ..

يمكنك زيارتنا على الموقع الالكتروني
http://www.2nees.com
للمزيد من الدورات والدروس التعليمية المجانية

أخوكم: أنيس حكمت أبو حميد

وآخر دعوانا أن الحمد لله رب العالمين

أنيس حكمت أبوحميد aneeshikmat@gmail.com