

# بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ



## دورة JS DOM & JS BOM

تقديم: أنيس حكمت أنيس أبو حميد.

البريد الإلكتروني: [aneeshikmat@2nees.com](mailto:aneeshikmat@2nees.com)

الموقع الإلكتروني: [www.2nees.com](http://www.2nees.com)

# الفهرس



● المقدمة

● متطلبات الدورة

● JAVASCRIPT DOM

○ تعريف ال DOM

○ تعريف ال HTML DOM

○ HTML DOM METHOD & Property

○ JS DOM (methods, property can be used)

○ Finding HTML Elements

○ Changing HTML & CSS

○ HTML DOM Events

○ Listener Event



## JAVASCRIPT BOM ●

BOM تعريف ال ○

JS Popup ○

JS Timing ○

JS Screen ○

JS Location ○

JS History ○

JS Cookies ○

الخاتمة ●

## المقدمة



الحمد لله الذي علم بالقلم، علم الإنسان ما لم يعلم، أحمدده حمد الشاكرين، وأثني عليه بما هو أهله، والصلاة والسلام على معلم الناس الخير، وعلى آله وصحبه، وكل من دعا بدعوته واقتفى أثره إلى يوم الدين.

أما بعد فإن طلب العلم ونشره من أفضل القربات التي يتقرب بها العبد من ربه، ومن أبرز الطاعات التي تعلي منزلة المسلم، وترفع قدره عند الله تعالى، ولقد أمر الله عباده بالعلم والتعلم، والتفكير والتدبر، وحذرهم من الجهل واتباع الهوى، وبين أن العلم الذي ينفع صاحبه يوم القيامة هو العلم الذي يخلص فيه العبد لمولاه، ويبتغي فيه نيل رضاه، ويتأدب فيه بأدب الإسلام، ويتخلق بأخلاق سيد الأنام، صل الله عليه وسلم، الذي كان خلقه القرآن، والحمد لله رب العالمين.

هذا الكلام الجميل “ منقول “

# المقدمة



أحبتي في الله، في هذه الدورة المتواضعة، سنكمل ما بدأناه في مشوار تعلم وإتقان “الجافا سكربت”، وسيكون موضوع هذه الدورة في جزئين إثنين وهما:

JavaScript DOM (1)

JavaScript BOM (2)

نسأل الله تعالى، أن يوفقنا وإياكم في هذا العمل، وفي كل الأعمال الى كل خير.

# متطلبات الدورة



لكل دورة منهاج ومجموعة من المتطلبات لضمان الإستمرارية في التعلم،  
وتحصيل أكبر قدر من المعلومة القابلة للفهم والتطبيق بالشكل الصحيح،  
وهنا ستكون متطلبات الدورة هي:

(1) HTML

(2) CSS

(3) JS ( معرفة الأنواع والمتغيرات والمصفوفات وال function..الخ)

(4) محرر نصوص مثل notepad++ ومتصفح إنترنت :P.

يمكنك تحميل الدورة الخاصة بالجافا سكربت (المتطلب السابق) [من هنا](#) ^ ^

1



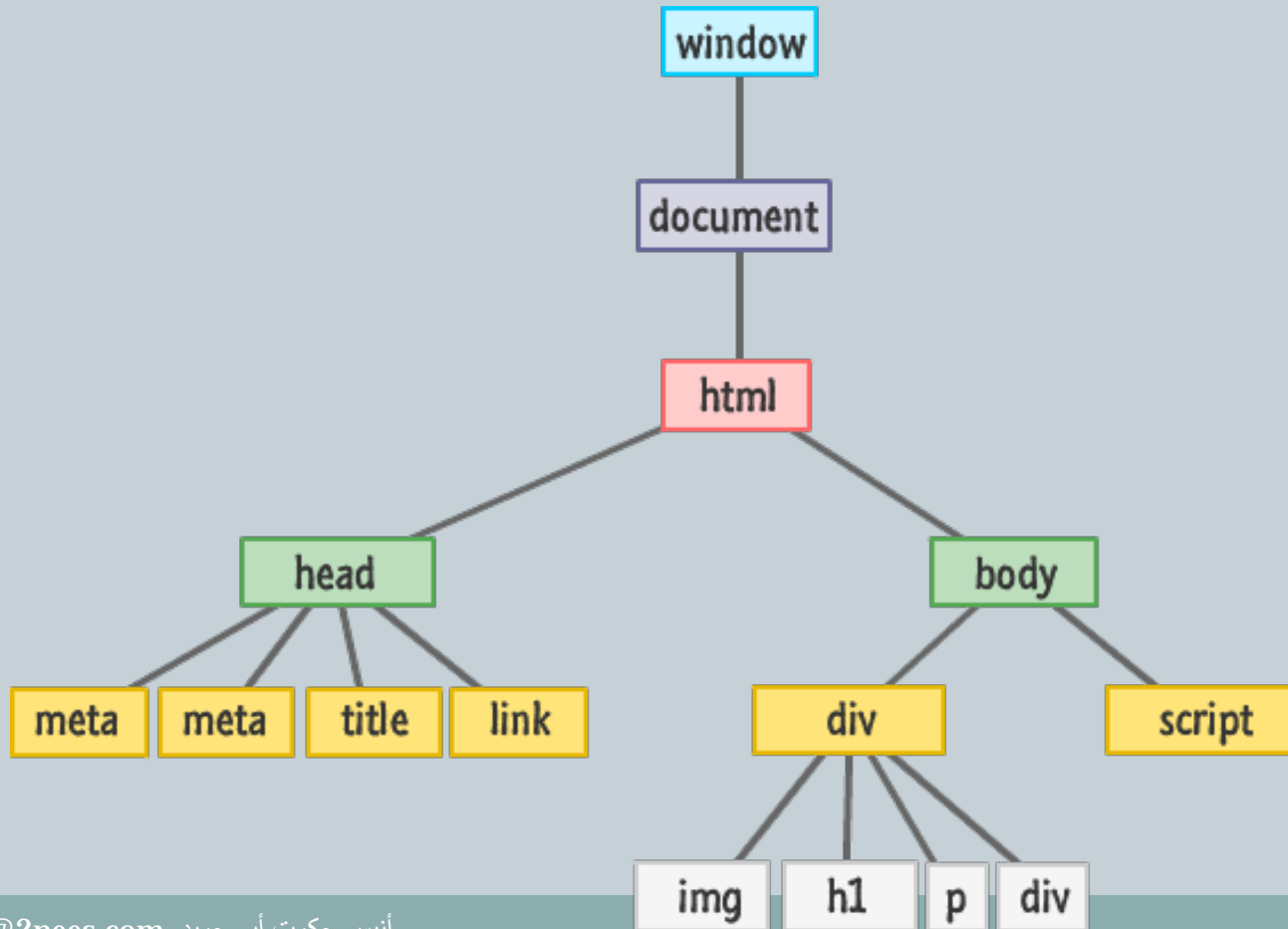
# JAVASCRIPT **DOM** (**DOCUMENT OBJECT MODEL**)

ملاحظة: عندما أتحدث عن JS DOM فأنا أختصرها من JS HTML DOM

# تعريف ال DOM



قبل البدء، شاهد هذه الصورة بتمعن، ثم انتقل للشريحة التالية ^\_^





# تعريف ال DOM



والآن، لنبدأ بتحليل الموضوع بشكل علمي بناءً على الصورة السابقة<sup>٨٨</sup>  
أولاً ال DOM هي إختصار لثلاث كلمات Document Object Model  
وهي حسب المنظمة العالمية لاعتماد معايير الويب (W3C) : “ هي نظام  
أساسي ولغة محايدة تسمح للبرامج ولغات السكريبت من الوصول الى  
المحتوى، والبنية التركيبية للملف، والتنسيق..، والتعديل عليه  
بشكل Dynamic .. ويتم تقسيم ال DOM الى 3 أقسام رئيسية وهي:

Core DOM (1

**HTML DOM** (2

XML DOM (3

# تعريف ال HTML DOM



إعتمادا على تعريف ال DOM، يمكننا تخيل تعريف ال HTML DOM ويمكن باختصار إعادة صياغة التعريف لل HTML DOM على أنها: المعايير التي تم اعتمادها من قبل ال W3C للتعامل مع جميع عناصر ال HTML، من خلال عمليات الحذف والإضافة والتعديل.

ونحن هنا، وبهذه الدورة بصدد الحديث عن التعامل مع ال HTML DOM، لذلك ذكرت في الشريحة الأولى (الجزء الأول) ملاحظة، أنني بحديثي عن JS DOM فأنا أعني JS HTML DOM ^\_^ ...

# تعريف ال HTML DOM



ما هي الأمور التي يمكنني إنجازها بعد تعلم هذا الجزء من الدورة؟  
الجواب هو:

- (1) تغيير المحتوى الموجود بال HTML بشكل Dynamically من خلال الجافا سكربت .
- (2) تغيير التنسيق الخاص بأي عنصر من عناصر ال HTML.
- (3) إضافة عناصر أو محتوى جديد على الصفحة .
- (4) حذف عنصر أو محتوى من الصفحة .
- (5) القيام بدور تفاعلي بين الصفحة وأي Event تقوم بتصميمه ..
- (6) إمكانية التعديل أو الحذف أو الإضافة على أي من خصائص العناصر ( attribute)

والآن لنبدء على بركة الله تعالى، في أول موضوع رسمي لنا في هذا الجزء ^\_^

# خاطرة



قال أبو بكر الصديق رضي الله عنه، يوم وفاة الحبيب المصطفى صل الله عليه وسلم:

ألا من كان يعبد محمدا صل الله عليه وسلم فإن محمدا قد مات ومن كان يعبد الله فإن الله حي لا يموت وقال إنك ميت وإنهم ميتون وقال وما محمد إلا رسول قد خلت من قبله الرسل أفإن مات أو قتل انقلبتم على أعقابكم ومن ينقلب على عقبيه فلن يضر الله شيئا وسيجزي الله الشاكرين قال فنشج الناس يبكون.

مقتبس من حديث رقم 3467 من صحيح البخاري – كتاب فضائل الصحابة

فهذا الحبيب المصطفى صل الله عليه وسلم قد مات، ونحن سنموت من بعده، لكن السؤال الأهم، الرسول صل الله عليه وسلم أدى الأمانة وبلغ الرسالة ونصح الأمة، أدى مهمته ونشهد له بذلك أمام الله تعالى، لكن هل قدمنا نحن أعمال تحقق الهدف من خلقنا؟ نسأل الله تعالى أن تكون أعمالنا في سبيل ما خلقنا من أجله.. اللهم آمين .

# HTML DOM METHOD & Property



- (1) **Method**: هي ال Action الذي سأقوم باستخدامه للوصول أو تحديد العنصر الموجود في صفحة ال HTML.
- (2) **Property**: هي القيمة التي يمكننا جلبها من عنصر HTML أو تعديلها على عنصر ال HTML ...  $\wedge\_ \wedge$

لنشاهد مثالا ذلك  $\wedge\_ \wedge$ :

ملاحظة 1: **innerHTML** تستخدم لجلب البيانات أو تعديلها من عنصر HTML

ملاحظة 2: **getElementById** تستخدم للبحث وتحديد عنصر HTML من خلال ID الخاص به  $\wedge\_ \wedge$  ..

# HTML DOM METHOD & Property

aneeshikmat@2nees.com أنيس حكمت أبو حميد

مثال:

```
<body>  
<h1 id="hTitle"> Anees Will Get This Text (DOM)</h1>  
<p id="pText"></p>  
<script>  
  document.getElementById("pText").innerHTML = "Anees Set This Text By JS DOM";  
  alert(document.getElementById("hTitle").innerHTML);  
</script>  
</body>
```

**Anees Will Get This  
Text (DOM)** النتيجة

Anees Set This Text By JS DOM

JavaScript Alert

Anees Will Get This Text (DOM)

OK

لاحظ هنا أن ال `getElementById` هي ال Method حسب التعريف وأن ال `innerHTML` هي ال Property احتوت على "قيمة"



# JS DOM (methods, property can be used)



الآن، سنتطرق بإذن الله تعالى، الى مجموعة من أشهر **methods** التي يمكننا استخدامها مع ال **JS** لتغيير عناصر ال **HTML** بجزئياتها المختلفة نذكر منها مع الوظيفة:

ملاحظة: سنتطرق بإذن الله تعالى، الى أمثلة تشرح هذه الدوال بعد ذلك أو في المواضيع القادمة إن شاء الله تعالى <sup>^</sup>\_<sup>^</sup>.

**(1) Method** نستخدمها للبحث عن عنصر في صفحة ال **HTML** وهي:

**document.getElementById()** وتستخدم للبحث عن عنصر من خلال ال **ID** (1)

**document.getElementsByTagName()** وتستخدم للبحث عن عناصر حسب اسم ال **tag** مثل “ul” (2)

**document.getElementsByClassName()** وتستخدم للبحث عن عناصر تحتوي كلاس معين مثل “className” (3)

# JS DOM (methods, property can be used)



(2) Method نستخدمها للتغيير على عناصر ال HTML وهي:

(1) `element.innerHTML`: وتستخدم لجلب محتوى عنصر ال HTML أو تغييره.

(2) `element.attribute`: وتستخدم لجلب قيم ال *Attribute* الخاصة بعنصر

HTML على شكل `NamedNodeMap object`.

(3) `element.setAttribute(attribute, value)`: وتستخدم لإضافة *Attr* الى

عنصر HTML مع تحديد قيمتها مثل اضافة كلاس باسم `className`.

ملاحظة: بكل تأكيد ستجد هناك وسائل مثل `getAttr..` و `hasAttr..` الخ .. لكن الموضوع هنا تغيير قيمة العنصر، ومن ثم بكل تأكيد لن أتطرق لكل الدوال التي يمكن استخدامها، فيمكنك البحث عنها بذاتك.

(4) `element.style.property`: وتستخدم هذه لتغيير التنسيق (*style*) الخاص بعنصر

ما على صفحة ال HTML.



# JS DOM (methods, property can be used)



(3) **Method** نستخدمها لإضافة أو حذف عناصر ال HTML وهي:

(1) `document.createElement()`: تستخدم لإنشاء عنصر HTML جديد.

(2) `document.removeChild()`: وتستخدم لحذف عنصر HTML.

(3) `document.appendChild()`: وتستخدم لإضافة عنصر HTML على عنصر

موجود.

(4) `document.replaceChild()`: وتستخدم لاستبدال أحد عناصر ال HTML

بعنصر آخر.

(5) `document.write(text)`: تستخدم هذه الدالة لإضافة محتوى HTML (بال tag)

أو بدون، أو إضافة سكريبت كود الى صفحة ال HTML، وغالبا ما تستخدم هذه لغايات

ال TEST.

# JS DOM (methods, property can be used)



## (4) إضافة Events Handlers الى HTML DOM:

(1) `:document.getElementById(id).onclick=function(){code}`

وتستخدم هذه الصيغة للبحث عن عنصر الذي يحتوي " ال id " المراد، ومن ثم تنفيذ أمر معين (function) ما بنائاً على onclick event ^^

(2) بنفس الأسلوب يمكننا استخدام onchange وغيرها من ال Event مكان ال Click.

من الأمثلة على استخدام هذه العمليات، ال Validation (التحقق من الأخطاء) ، أو إظهار نصوص وإخفائها (collapse) .. الخ

# JS DOM (methods, property can be used)



(5) HTML DOM Object تحتوي من ال DOM 1 العديد من الخصائص التي يمكننا استخدامها عن طريق ال Object `^_^`، لنشاهد أهمها `^^`:

(1) `document.anchors`: وتستخدم لجلب جميع ال `<a>` “ من الصفحة والتي قد تحتوي صفة ما، ترتيب ما، أو حتى عددهم باستخدام `length` بعد ال `anchors` `^^`.

(2) `document.baseURI`: تستخدم لجلب الرابط (Absolute URI) الخاص بالصفحة.

(3) `document.body`: تستخدم لجلب ال `<body></body>` وما يحتويه `^^`.

(4) `document.cookie`: هذه الخاصية المهمة تستخدم لاسترجاع جميع ال `cookie` الخاصة بي `^_^`.

(5) `document.doctype`: تستخدم لإرجاع نوع ال `doctype` الخاص بالصفحة `^^`.

(6) `document.documentElement`: تستخدم لإرجاع `<html></html>` وما تحتوي.

# JS DOM (methods, property can be used)



## (5) تابع HTML DOM Object

- (7) `document.documentURI`: تقوم على إرجاع الرابط الخاص بموقع ملف معين، وتعمل على جميع أنواع الملفات..
- (8) `document.domain`: تقوم على إرجاع الدومين الخاص بالموقع مثل [www.2nees.com](http://www.2nees.com)
- (9) `document.embeds`: تقوم على إرجاع جميع ال `<embed>` في الصفحة.
- (10) `document.forms`: تقوم على إرجاع جميع ال `<form>` في الصفحة.
- (11) `document.head`: تقوم على إرجاع ال `<head></head>` في الصفحة وما تحويه.
- (12) `document.images`: تقوم على إرجاع جميع ال `<images>` في الصفحة.
- (13) `document.inputEncoding`: هذه الخاصية الجميلة ترجع ال Encode الخاص بالصفحة مثل: UTF-8 أو ISO.. الخ

# JS DOM (methods, property can be used)



## (5) تابع HTML DOM Object

(14) `document.links`: تستخدم هذه لإرجاع جميع العناصر من نوع `<a>` أو `<area>` والتي تحتوي على `href` `^_^`.

(15) `document.scripts`: تستخدم هذه الدالة لإرجاع جميع ال `<script></script>` المستخدمة في صفحة ال HTML.

(16) `document.title`: تقوم على إرجاع العنوان الخاص بالصفحة من ال `tag` `<title>`.

(17) `document.URL`: تستخدم هذه لإرجاع ال URL الخاص بالملف.

ملاحظة: وظيفة لك: يمكنك البحث عن الفرق بين ال URI وال URL وال URN، كما يمكنك البحث عن الفرق بين ال `object` التي قمت باستخدامها لجلب الروابط `^_^`.

# JS DOM (methods, property can be used)



والآن يا صديقي العزيز، لننتقل الى الخطوة التالية! \_! P:، لكن قبل الانتقال لنستعرض بعض الأمثلة على ما ذكرنا، وبكل تأكيد لن أقوم بتنفيذ جميع ما ذكرت، لكن سأختار الآن عينة انتقائية، وبعد ذلك سننتقل بإذن الله تعالى الى موضوع جديد، كما أننا في المواضيع القادمة سنتطرق الى استخدام ما ذكرنا دون العودة الى

الشرح مجدداً ^\_^  
مثال :

(هل يمكنك كتابة

الشفيرة البرمجية

الخاصة بهذه النتيجة

قبل الانتقال

للشريحة التالية؟ )

## Anees Will Get This Text (DOM)

[2nees.com](#)

```
Elements Network Sources Timeline Profiles Resources Audits Console
<top frame>
innerHTML Example Anees Will Get This Text (DOM)
document.doctype Example <!DOCTYPE html>
document.links Example ▶ [a, item: function, namedItem: function]
document.title Example Example 2
document.body Example ▶ <body>...</body>
```

ناتج تطبيق المثال - بعد مشاهدتك للنتائج انتقل للشريحة التالية

# JS DOM (methods, property can be used)



aneeshikmat@2nees.com أنيس حكمت أبو حميد

الشيفرة البرمجية الخاصة بالشاشة السابقة:

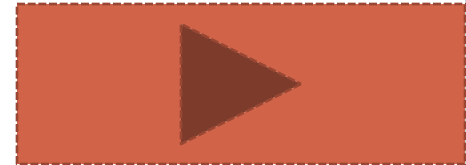
```
<!DOCTYPE html>
<html>
  <head>
    <title>Example 2</title>
  </head>

  <body>
    <h1 id="hTitle"> Anees Will Get This Text (DOM)</h1>
    <a href="http://www.2nees.com">2nees.com</a>
    <button onclick="addAndAppend()">Click Here</button>

    <script>
      var btnCount = 1; // Count for btn Text
      function addAndAppend() {
        var btn = document.createElement("BUTTON");
        var btnText = document.createTextNode("New Button " + btnCount);
        btnCount++;
        btn.appendChild(btnText);
        document.body.appendChild(btn);
      }

      console.log("innerHTML Example " , document.getElementById("hTitle").innerHTML);
      console.log("document.doctype Example " , document.doctype);
      console.log("document.links Example " , document.links);
      console.log("document.title Example " , document.title);
      console.log("document.body Example " , document.body);
    </script>
  </body>
</html>
```

لا تنسى مشاهدة ال Console



# خاطرة



أعوذ بالله من الشيطان الرجيم

{ قل إن صلاتي ونسكي ومحياي ومماتي لله رب العالمين لا شريك له وبذلك أمرت وأنا أول المسلمين } .

صدق الله العظيم



# Finding HTML Elements



بعد حديثنا المتواضع عن ال `method` وكيفية إستخدامها، وبعد شرح الفكرة العامل لكل تصنيف قمنا بذكره، ننتقل الى تخصيص عملية الشرح حسب كل تصنيف، وذلك بذكر أمثلة متنوعة على كل جزئية، خصوصا ما يهمنا في التعامل مع الجافا سكربت ..

وباختصار، فإن هذا الموضوع سيتحدث عن البحث والوصول الى عناصر ال `HTML` بالصفحة <sup>^^</sup> وما سيتم استخدامه من خصائص هي ال `ID` والكلاس وال `Tag` وال `Css selector` وال `Html Obj` ..

# Finding HTML Elements



أولاً: من خلال ال ID:

```
<div id="hTitle"> Anees Will Get This Text (DOM) </div>
```

```
// Find And Access Element By id  
var findById = document.getElementById("hTitle");  
console.log("A) Original Text By ID" , findById.innerHTML);  
findById.innerHTML = "A) Anees New Text ^_^ (look at console)";
```

A) Anees New Text ^\_^ (look at console)

Elements Network Sources Timeline Profiles Resources Audits Console

<top frame>

A) Original Text By ID Anees Will Get This Text (DOM)

# Finding HTML Elements



aneeshikmat@2nees.com أنيس حكمت أبو حميد

ثانيا: كم خلال ال Tag Name:

```
<p class="fClass">Find By tag 1 (this have a fClass)</p>
<p>Find By tag 2</p>
<p>Find By tag 3</p>
```

```
// Find And Access Element By Tag Name
var findByTagName = document.getElementsByTagName("p");
console.log("B) Original Text By Tag" , findByTagName[1].innerHTML);
findByTagName[1].innerHTML = "B) Update After Find & access by <b>Tag Name</b> for Index (1)";
```

Find By tag 1 (this have a fClass)

B) Update After Find & access by **Tag Name** for Index (1)

Find By tag 3

Elements Network Sources Timeline Profiles Resources Audit

<top frame>

B) Original Text By Tag Find By tag 2

> |

# Finding HTML Elements

aneeshikmat@2nees.com أنيس حكمت أبو حميد



ثالثا: من خلال  
:Class Name

```
<p class="fClass">Find By tag 1 (this have a fClass)</p>
```

```
<p>Find By tag 2</p>
```

```
<p>Find By tag 3</p>
```

```
<span class="fClass">This span And Have fClass 4</span><br />
```

```
<span class="fClass">This span And Have fClass 5</span><br />
```

```
// Find And Access Element By Class Name
```

```
var findByClassName = document.getElementsByClassName("fClass");
```

```
console.log("C) Original Text By Tag" , findByClassName[0].innerHTML);
```

```
findByClassName[0].innerHTML = "C) Update After Find & access by <b>Class Name</b> for Index (0)";
```

C) Update After Find & access by **Class Name** for Index (0)

Find By tag 2

Find By tag 3

This span And Have fClass 4

This span And Have fClass 5

🔍 Elements Network Sources Timeline Profiles Resources Audits Conso

🗑️ <top frame>

C) Original Text By Tag Find By tag 1 (this have a fClass)

# Finding HTML Elements

aneeshikmat@2nees.com أنيس حكمت أبو حميد



رابعاً: من خلال  
query  
:Selector All

```
<p class="fClass">Find By tag 1 (this have a fClass)</p>  
<p>Find By tag 2</p>  
<p>Find By tag 3</p>
```

```
<span class="fClass">This span And Have fClass 4</span><br />  
<span class="fClass">This span And Have fClass 5</span><br />
```

```
// Find And Access Element By query Selector All  
var findByQuerySelectorAll = document.querySelectorAll("span.fClass");  
console.log("D) Original Text By Tag" , findByQuerySelectorAll[0].innerHTML);  
findByQuerySelectorAll[0].innerHTML = "D) Update After Find & access by <b>query Sele
```

Find By tag 1 (this have a fClass)

Find By tag 2

Find By tag 3

D) Update After Find & access by **query Selector All** for Index (0) For **Span with fClass**  
This span And Have fClass 5

Elements Network Sources Timeline Profiles Resources Audits Console

<top frame>

D) Original Text By querySelectorAll This span And Have fClass 4

# Finding HTML Elements

Name:

Age:

Email:

Anees Hikmat  
24  
aneeshikmat@2nees.com  
Submit



aneeshikmat@2nees.com أنيس حكمت أبو حميد

## خامسا: من خلال HTML Object :Collections

```
<form id="formCollections">
  Name: <input type="text" name="txt1" value="Anees"><br />
  Age: <input type="number" name="txt2" value="24"><br />
  Email: <input type="email" name="txt3" value="aneeshikmat@2nees.com"><br /><br />
  <input type="button" value="Submit" onclick="formCollectionsAction();" />
</form>
<div id="formCollectionsOutput"></div>
```

```
// Find And Access Element By HTML Object Collections
function formCollectionsAction() {
  var formCollections = document.forms["formCollections"];
  var fValue = "";
  var i;
  for (i = 0; i < formCollections.length; i++) {
    fValue += formCollections.elements[i].value + "<br />";
  }
  document.getElementById("formCollectionsOutput").innerHTML = fValue;
}
```

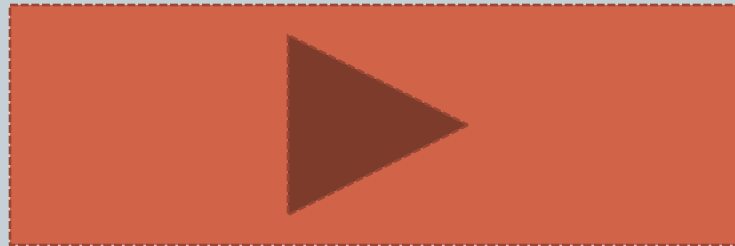
Form ID

# Finding HTML Elements



^\_^ هل رأيت سهولة وجمال الأمر، بكل بساطة كل ما عليك القيام به هو استخدام ما تعلمناه من دوال، ومن ثم الإنطلاق والإبداع ضمن هذا العالم، والآن أرجوا أنك قمت بتطبيق جميع الأمثلة السابقة، والافلن تستفيد خبرة برمجية !!

إذا قمت بتنفيذها أو قد واجهت مجموعة من الأخطاء، فل تشاهد تطبيق لجميع الأمثلة السابقة:



# خاطرة



فلا تصحب أبا الجهل \*\*\* وإياك وإياه  
فكم من جاهل أردى \*\*\* حليماً حين آخاه  
يقاس المرء بالمرء \*\*\* إذا ما المرء ماشاه  
و للشيء من الشيء \*\*\* مقاييس وأشباه  
وللقاب على القاب \*\*\* دليل حين يلقاه

علي بن أبي طالب رضي الله عنه وأرضاه



# Changing HTML & CSS



أولاً: Changing Html: من أشهر الأمثلة التي استخدمناها بكثرة في هذه الدورة والى الآن ال innerHTML ^\_^، والآن لنطلع على غيرها:

(1) `document.write()`: تحدثنا سابقاً أن هذه الدالة تستخدم للكتابة على صفحة ال HTML ويمكن ان تقوم بتنفيذ شفرة برمجية (جافا سكربت) وطباعة نتائجها أو يمكننا مباشرة استخدام HTML.. شاهد المثال:

ملاحظة: (إذا قمت باستخدام هذه الدالة بعد تحميل الصفحة، سيقوم بحذف المحتوى القديم).

```
<script>
```

```
document.write(Date()); // Write Js  
document.write("<h2>2nees.com</h2>"); // Write Html
```

```
</script>
```

Tue Sep 01 2015 14:25:10 GMT+0300 (Jordan Daylight Time)

# Changing HTML & CSS



(2) تغيير القيمة الخاصة ب Attribute: يمكننا تغيير أيضا القيمة الخاصة ب Attr، فالتعديل لا يقتصر على المحتوى  $^_^$ ، والصيغة العامة لذلك هي:  
`document.getElementById(id).attribute=new value`

أنيس حكمت أبو حميد aneeshikmat@2nees.com

مثال:

```
<a href="#" id="changeTo2nees">Change # to 2nees.com</a><br />

<script>
  document.getElementById("changeTo2nees").href = "http://www.2nees.com";//
  document.getElementById("changeTo2nees").target = "_blank";// Change Target
</script>
```

[Change # to 2nees.com](#)

Q Elements Network Sources Timeline Profiles Resources Audits Console

<!DOCTYPE html>

<html>

><head>...</head>

><body>

<a href="http://www.2nees.com" id="changeTo2nees" target="\_blank">Change # to 2nees.com</a>

# Changing HTML & CSS



ثانياً: Changing Css: هنا سنقوم باستخدام ال DOM لتغيير خصائص التنسيق لعنصر ما  $^_^$ ، وبكل بساطة الأمر لا يتعدى سوى إضافة `style` ومن ثم التنسيق المطلوب  $^^$ ، شاهد الصيغة العامة:  $^_^$

`document.getElementById(id).style.property=new style`

مثال:

أنيس حكمت أبو حميد neeshikmat@2nees.com

```
document.getElementById("changeColorByID").style.color = "red";// id (one element)
var elements = document.getElementsByClassName("changeColor");// Class thats mean more than 1 element
for (i = 0, len = elements.length; i < len; i++) {
    elements[i].style.color = '#ff00ff';
}
```

```
<div id="changeColorByID">Welcome Man *_*</div>
<div class="changeColor">Anees Hikmat</div>
<div class="changeColor">AABU ^_^</div>
```

Welcome Man \*\_\*  
Anees Hikmat  
AABU ^\_^

# Changing HTML & CSS



والآن لنشاهد ناتج تطبيق الأمثلة السابقة معا:

(لا تستعرض الشيفرة البرمجية قبل محاولتك تنفيذ أو تطبيق ما تم ذكره في الشرائح السابقة)

[Change # to 2nees.com](#)

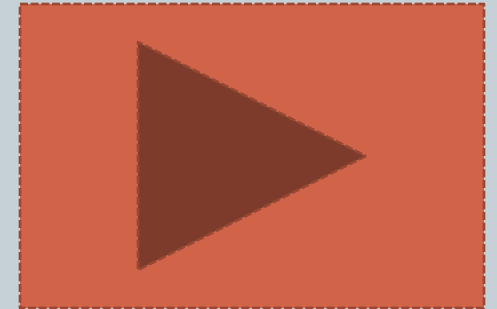
Welcome Man \* \_ \*

Anees Hikmat

AABU ^ \_ ^

Tue Sep 01 2015 14:54:52 GMT+0300 (Jordan Daylight Time)

**2nees.com**



# HTML DOM Events

aneeshikmat@2nees.com أنيس حكمت أبو حميد



صفحات الويب تحتوي العديد من الأمور التفاعلية بينها وبين المستخدم، مثل النقر على أزرار وروابط وتحديد نصوص وتحريك مؤشر الفأرة... الخ، وهذا الأمر يتطلب تدخل ال Script حتى يتسنى التحكم في هذه ال Action التي قام بها المستخدم ، وهذا ما يسمى بال Event  $.. \wedge \_ \wedge$  والآن لنشاهد مثالاً على استخدام ال Event الأكثر شهرة: **(onclick) P**

```
<div onclick="this.innerHTML='Good Bye'">Welcome Man *_* (click here)</div>  
<div onclick="changeText(this)">Anees Hikmat (click here)</div>
```

```
<script>
```

```
function changeText(elementClicked) { // Change text by function  
    elementClicked.innerHTML = "Major: CIS";  
}
```

```
</script>
```

قبل النقر

بعد النقر

Welcome Man *_* (click here)	Anees Hikmat (click here)	Good Bye	Anees Hikmat (click here)
------------------------------	---------------------------	----------	---------------------------

# HTML DOM Events



والآن لنشاهد بعضا من ال event التي يمكننا إستخدامها :

- (1) **onchange**: عندما يحدث تغيير على عنصر ال html (التي تقبل الإدخال) مثل الكتابة داخل **textbox**.
- (2) **onmouseover**: ينطلق هذا Event عندما يمر مؤشر الفأرة فوق عنصر ال HTML.
- (3) **onmouseout**: ينطلق هذا ال Event عندما يصبح مؤشر الفأرة خارج العنصر.
- (4) **onmousedown**: ينطلق هذا ال Event عندما يتم النقر على زر الفأرة، وينتهي عندما ترفع أصبع يدك على الفأرة لينطلق ال Event التالي
- (5) **onmouseup**: ينطلق هذا ال Event مباشرة بعد عند ترك الأصبع للزر ( بعد عملية الضغط Down).

ملاحظة: ترتيب عمليات النقر على زر الفأرة يتم أولا **down** ثم **up** ثم **click**

# HTML DOM Events



(6) **onfocus**: ينطلق هذا ال Event عندما يتم تحديد عنصر ال **html**. (سيتم عرض مثال لتتضح الفكرة بعد ذكر بعض ال Events بإذن الله تعالى).

(7) **onload**: ينطلق هذا ال Event بعد الإنتهاء من تحميل عنصر ال HTML المراد.

(8) **onkeydown**: ينطلق هذا ال Event عند الضغط على أحد أزرار لوحة المفاتيح (عندما يصبح زر لوحة المفاتيح بالأسفل)

(9) **onkeyup**: ينطلق ال Event عندما يصبح زر لوحة المفاتيح في الأعلى.

(10) **onkeypress**: ينطلق هذا ال Event عند الضغط على لوحة المفاتيح، لكن هذه الخاصية لا تعتمد جميع أزرار لوحة المفاتيح (فيستثنى مثلا **shift, ctrl, alt**..الخ)

# HTML DOM Events



(11) **onblur**: ينطلق هذا ال Event عند مغادرة عنصر ال HTML (بمعنى آخر، الحدث المقابل لل onfocus)

(12) **onsubmit**: ينطلق هذا ال Event عند الضغط على Submit الخاصة بال Form. (When Form Is Submitted).

(13) **oncopy**: ينطلق هذا ال Event عند القيام بعملية ال copy لنص معين  $\wedge\_ \wedge$  بأي طريقة من الطرق الثلاث (ctrl & c أو من خلال الفأرة أو المتصفح).

(14) **onscroll**: ينطلق هذا ال Event عند القيام بعملية scrolling.

(15) **onresize**: ينطلق هذا ال event عند القيام بتغيير حجم شاشة العرض لل document.



# HTML DOM Events



هذه ال Events التي ذكرناها هي جزء من كل، أي يوجد هناك الكثير من ال Events غيرها، وحاولت الإقتباس هنا من كل جزئية بعض ال events فمثلا من ال Clipboard Event قمت بأخذ ال oncopy وهناك مثلا onpaste لم أتطرق لها ^\_^، لذلك أريد أن تبحث عن ال Events الأخرى، ولن تختلف الفكرة عما ذكرناه، فهذه ال Events سهلة الإستخدام، والآن سنقوم بشرح طريقة الإستخدام الخاصة بال Events التي تطرقنا لها بإذن الله تعالى، مع بعض الملاحظات ومثال على جزء مما ذكرنا من Events .. ^\_^

# HTML DOM Events



ملاحظة: يمكنك عمل `assign` ل `Event` معين من خلال `JS DOM` مثل:

```
document.getElementById("btn").onclick = yourScript;
```

والآن لنشاهد مثال على ما ذكرنا وتطرقنا اليه:

ناتج تطبيق المثال



لاحظ الفرق بين نتيجة ال `keydown` وبين ال `keyup` والسبب في ذلك يعود الى أن ال `keydown` تنطلق بمجرد أن ينزل زر لوحة المفاتيح وبهذا فإن أول تنفيذ سيكون قبل معرفة ال `value` التي تم إدخالها ^\_^

# HTML DOM Events



صفحة ال HTML:

```
<div id="section-1-wrapper">
  <div onclick="this.innerHTML='Good Bye'">Welcome Man *_* (click here)</div>
  <div onclick="changeText(this)">Anees Hikmat (click here)</div>
  <div onmouseover="this.style.background='#ff00ff'"
        onmouseout="this.style.background='#00ff00'">
    Mouse Over Here And Then Out :P
  </div>
  <div id='cTxt' oncopy="this.style.color='#ff0000'">Copy 2nees.com</div>
</div>

<br />
<input type="text" id="assignEvent" placeholder="work with txt ^^"
        onfocus="focusFunction(this)"
        onblur="blurFunction(this)"
        onkeydown="keyDownFunction(this)"
        onkeyup="keyUpFunction(this)"/>

<div id="txtKeyDown">Key Down Result</div>
<div id="txtKeyUp">Key Up Result</div>
```

# HTML DOM Events



aneeshikmat@2nees.com أنيس حكمت أبو حميد

## الجافا سكربت في الصفحة:

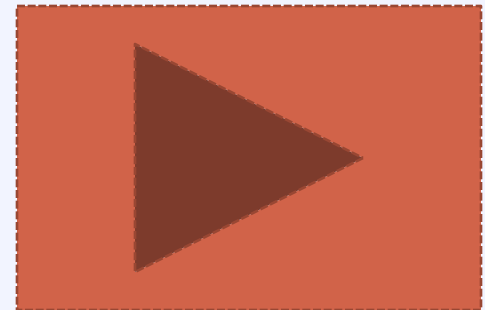
```
<script>
  function changeText(elementClicked) { // Change text by function
    elementClicked.innerHTML = "Major: CIS";
  }

  function focusFunction(id) {
    id.style.background = '#ab1100';
  }

  function blurFunction(id) {
    id.style.background = '#ffff00';
  }

  function keyDownFunction(id) {
    id.style.background = '#f0f0f0';
    document.getElementById("txtKeyDown").innerHTML = "Key Down: " + id.value;
  }

  function keyUpFunction(id) {
    id.style.background = '#00ff00';
    document.getElementById("txtKeyUp").innerHTML = "Key Up: " + id.value;
  }
</script>
```



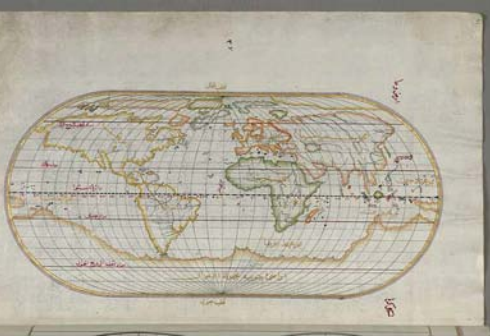
# خاطرة



هل تعرف من هو بييري ريس؟

هو بحار مسلم، استطاع رسم خرائط تشمل معظم أجزاء العالم، خرائطه هذه وضعت علماء الغرب في ورطة، لأن بييري ريس انتهى من رسم الخريطة قبل 27 عام من اكتشاف كولومبوس للقارة الأمريكية!! وكانت تسمى بخرائطهم بـ “ أنتيليا ”، والأعجب من هذا أن دقة هذه الخرائط، جعلتهم يتحدثون أن هناك رجال فضاء ساعدوهم في هذا “ ههههه ”، لأن هناك تطاق بين هذه الخرائط وبين صور الأقمار الصناعية بشكل مخيف!!

والسؤال المهم، من أرسل كولومبوس في هذا الوقت الى أمريكا؟ وما هي توجهاته؟ وما الذي قام بفعله هؤلاء بالهنود الحمر؟، ولماذا؟  
مجرد أسئلة بريئة!



# Event Listener



`addEventListener()` تستخدم هذه الدالة JS DOM لإعطاء عنصر من عناصر ال HTML أحد أو مجموعة من ال `Event/s` التي تعلمناها وذكرناها سابقا، وتتميز هذه الدالة بإمكانية

(1) يمكنك إضافة أكثر من `Event Click` لنفس العنصر ودون أن يتم عمل `override` للأول!، بمعنى آخر لو فرضنا أن هناك رابط `x`، وقمت باستخدام هذه الدالة لإضافة `click` الأول يقوم بطباعة “ أنيس ” وال `click` الثانية تقوم بطباعة “ حكمت ”، فسيقوم بتنفيذ وطباعة “ أنيس حكمت ” عند الضغط على الرابط `x`.

(2) يمكنك من خلال إستخدامها وضع أكثر من `Event` لنفس العنصر، يعني مثلا `click, onfocus, oncopy`..الخ، وذلك من خلال السكربت.

# Event Listener

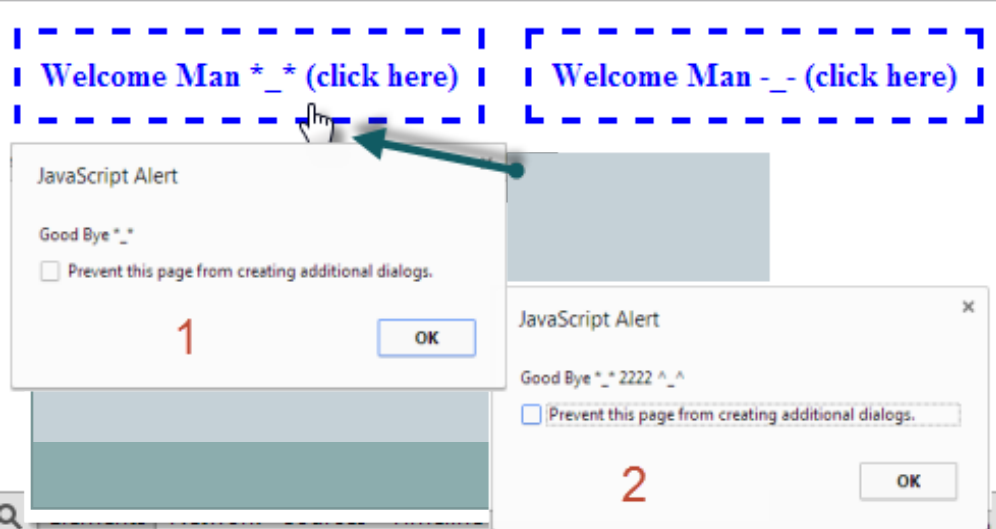


(3) يمكننا أيضا استخدام ال Event من خلالها مع ال window، ولا نكتفي فقط بعناصر ال HTML ^\_^

(4) أيضا من مزاياها أنها أسهل في التحكم، والتعديل، ووضوح الشيفرة البرمجية من الطريقة التقليدية ^\_^.

(5) ويمكنك بسهولة حذف ال Event Listener الذي عملت عليه فقط من خلال جملة واحدة ^\_^ وهي `removeEventListener()`

والآن لنشاهد مثالا: (هذه نتائج المثال)



# Event Listener



الشيفرة البرمجية:

```
<div id='d-1'>Welcome Man *_* (click here)</div>  
<div id='d-2'>Welcome Man -_- (click here)</div>
```

```
<script>
```

```
document.getElementById("d-1").addEventListener("click", function(){  
    alert("Good Bye *_*");  
});  
document.getElementById("d-1").addEventListener("click", function(){  
    alert("Good Bye *_* 2222 ^_^");  
});
```

many event handler

```
document.getElementById("d-2").addEventListener("click", displayMessage);  
function displayMessage() {  
    alert("Good Bye -_-");  
}
```

basic example  
(using function)

```
</script>
```



# Event Listener



aneeshikmat@2nees.com أنيس حكمة أبو حميد

مثال 2:

```
var assignEvent = document.getElementById("assignEvent");
assignEvent.addEventListener("focus", focusFunction);
assignEvent.addEventListener("blur", blurFunction);
assignEvent.addEventListener("keydown", keyDownFunction);
assignEvent.addEventListener("keyup", keyUpFunction);

function focusFunction() {
    assignEvent.style.background = '#ab1100';
}

function blurFunction() {
    assignEvent.style.background = '#ffff00';
}

function keyDownFunction() {
    assignEvent.style.background = '#f0f0f0';
    document.getElementById("txtKeyDown").innerHTML = "Key Down: " + assignEvent.value;
}

function keyUpFunction() {
    assignEvent.style.background = '#00ff00';
    document.getElementById("txtKeyUp").innerHTML = "Key Up: " + assignEvent.value;
}
```

Add many of event for  
one element

# Event Listener



مثال 3:

```
// Passing Parameters to function
document.getElementById("d-3").addEventListener("click", function() {
    getPara("anees", "hikmat");
});
function getPara(fname, mname){
    alert(fname + " ^_^ " + mname);
}
```

في هذا المثال (سنقوم بارسال قيم من خلال ال Event Listener الى ال Function المراد استخدامه، وهذا أمر مهم، فإذا دقت النظر في الأمثلة السابقة ستجد أننا لم نرسل متغيرات، ولم نقم حتى بوضع () بعد اسم ال function الذي نقوم باستدعائه ^\_^.. لذلك انتبه لهذه النقاط ^\_^

# Event Listener



مثال 4: ( Bubbling أم Capturing ) Event Propagation ^\_^  
عندما يقوم المستخدم بالنقر على نص معين موجود داخل <p> وهذه ال  
<p> موجودة داخل <div>، هل سيتم تنفيذ ال click على p فقط ؟ أم على  
div فقط ؟ أم سيتم تنفيذ ال click على p ثم div ؟. أم سيتم تنفيذ ال click  
على div ثم p ؟

ما رأيك صديقي في الموضوع؟ وهل تراه مهما ؟

إن هذا الموضوع صديقي مهم وجميل جدا للتحكم في مجموع العمليات التي  
نرغب بتنفيذها والوقت المناسب لذلك، لذا فهذا الأمر مهم، ولم يوجد عبثا، أما  
بخصوص الأسئلة التي تم طرحها، فأجيبك بأنك يمكنك إختيار الإجابة التي  
ترغبها من خلال الشيفرة البرمجية الخاصة بك ^\_^...

# Event Listener



**Bubbling**: هذه تعني أن سير العمل لتنفيذ ال Event (وليكن click) إذا تم النقر على النص بالشكل التالي: `div -> p`.

**Capturing**: هذه تعني أن سير العمل لتنفيذ ال Event (وليكن click) اذا تم النقر على النص بالشكل التالي: `div -> p`.

الصيغة الافتراضية هي: `addEventListener(event, function, propagation)`; الحالة الافتراضية هي `false` (Bubbling) و `true` تعني `capturing` وظيفة:

بخصوص التحكم بخيار الضغط على ال `p` وتنفيذ ال Event دون ال `div` أو العكس، أو الغاء فعالية النقر (No Action)، فهذا يكون من خلال البحث والقراءة عن إحدى هذه المواضيع .. `stopPropagation()` أو `preventDefault()`

# Event Listener



aneeshikmat@2nees.com أنيس حكمت أبو حميد

والآن نعود لمشاهدة المثال 4:

```
<div id="myDiv">
  <p id="myP">Click this paragraph, I am <b>Bubbling.</b></p>
</div><br>

<div id="myDiv2">
  <p id="myP2">Click this paragraph, I am <b>Capturing.</b></p>
</div>

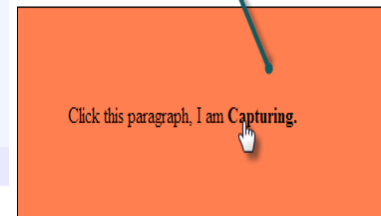
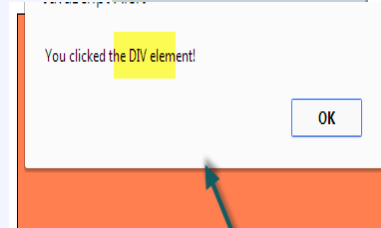
<script>
  // This Example from w3schools http://www.w3schools.com/js/tryit.asp?filename=t
  document.getElementById("myP").addEventListener("click", function() {
    alert("You clicked the P element!");
  }, false);

  document.getElementById("myDiv").addEventListener("click", function() {
    alert("You clicked the DIV element!");
  }, false);

  document.getElementById("myP2").addEventListener("click", function() {
    alert("You clicked the P element!");
  }, true);

  document.getElementById("myDiv2").addEventListener("click", function() {
    alert("You clicked the DIV element!");
  }, true);
</script>
```

مثال جميل على الموضوع من  
w3schools  
(ستجده مع الأمثلة)



# Event Listener



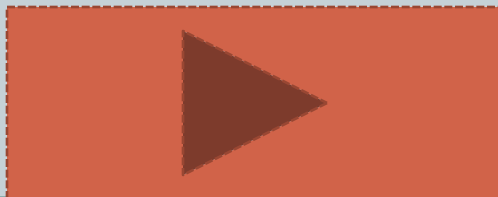
وآخر مثال وهو عن ال `removeEventListener`، والصيغة العامة له تشبه ال `addEventListener`، وعندما ترغب بحذف `Event` عليك كتابة ال `remove` بذات الطريقة الخاصة بال `add`... شاهد المثال:

```
document.getElementById("d-2").addEventListener("click", displayMessage);
```

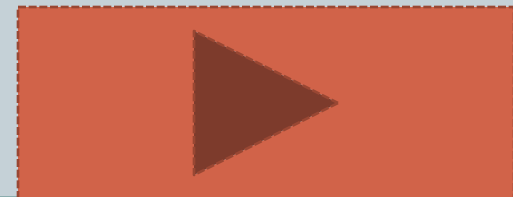
```
function displayMessage() {  
    alert("Good Bye --");  
}
```

```
function removeHandler() {  
    document.getElementById("d-2").removeEventListener("click", displayMessage);  
}
```

مثال 1



مثال 2



# خاطرة



لا إله إلا الله الملك الحقّ المبين، لا إله إلا الله العدل اليقين، لا إله إلا الله ربّنا وربّ آبائنا  
الأولين، سبحانك إني كنت من الظالمين، لا إله إلا الله وحده لا شريك له، له الملك وله  
الحمد يحيي ويميت، وهو حيّ لا يموت، بيده الخير، وإليه المصير، وهو على كلّ شيءٍ  
قدير، لا إله إلا الله إقراراً بربوبيّته، سبحان الله خضوعاً لعظمته، ولا حول ولا قوّة إلا  
بالله العليّ العظيم، لا إله إلا الله الملك المبين، لا إله إلا الله العدل اليقين، لا إله إلا الله  
وحده لا شريك له، له الملك وله الحمد وهو على كلّ شيءٍ قدير.

2



# JAVASCRIPT **BOM** (**BROWSER OBJECT MODEL**)



# تعريف ال BOM



ال BOM هي إختصار ل Browser Object Model، وهذه تعني الدوال والخصائص الموجودة في متصفح الإنترنت، والتي تسمح للجافا سكربت بالتعامل معها ^\_^..

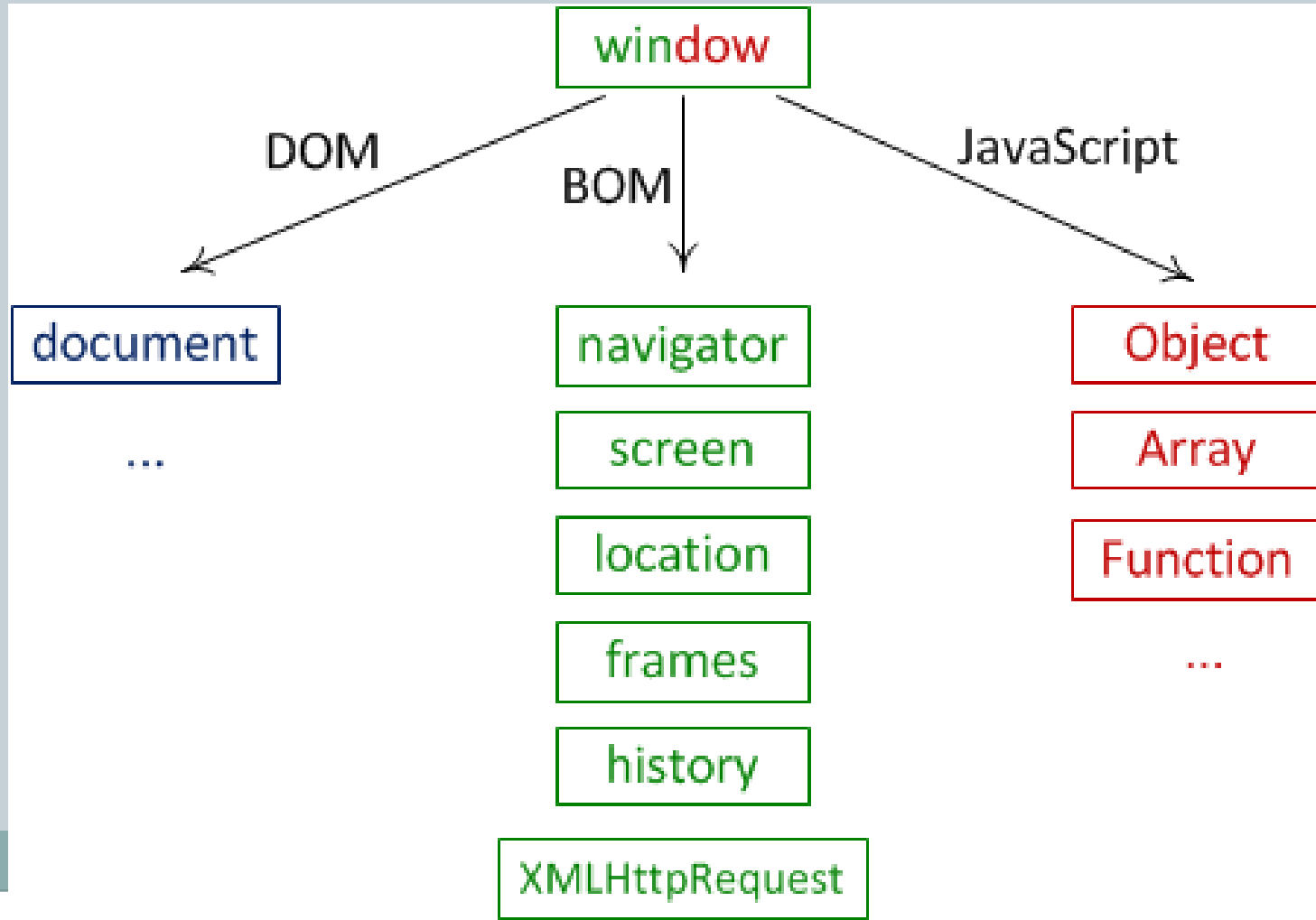
وبما أن التعامل هنا يعتمد على متصفح الإنترنت، فهذا يعني أن ما هو موجود في متصفح ما، ليس بالضرورة أن يكون موجودا في متصفح آخر، وهذا يعني أيضا “ لا توجد معايير رسمية لل BOM “، لكن مع هذا فإن الخاصية بالأهمية بمكان، ولهذا فإن المتصفحات وخصوصا الرئيسية منها، تحاول الحفاظ على ما يتم استخدامه بكثرة في الغالب ^\_^ .. والآن لنبدء على بركة الله تعالى، بدراسة هذه الجزئية، ولنبدء بالغوص في ال BOM \*\_\*

# تعريف ال BOM

أنيس حكمت أبو حميد [aneeshikmat@2nees.com](mailto:aneeshikmat@2nees.com)



والآن شاهد هذه الصورة، والتي توضح مفهوم ال BOM بشكل سهل ^\_^



# تعريف ال BOM



أول ملاحظة خطيرة ببالك الآن، وبعد مشاهدة الصورة السابقة كيف لل Window أن تكون في parent لكل من document وما تعلمناه عن ال HTML DOM وعن ال BOM نفسه وما تتعلق به من خصائص، وعن الجافا سكربت `^_^`، والجواب بكل بساطة يكمن في تعريف ال `^^window`

ال window هي object تم دعمه من جميع المتصفحات، ويحتوي على أشهر الدوال والخصائص (Global method & property) بالإضافة الى المتغيرات، وال function، والتي من خلالها يتم تمثيلها لك من خلال نافذة المتصفح، وجميع هذه العناصر من متغيرات ودوال وغيرها، تصبح بشكل تلقائي كعضو داخل ال `^_^ window object` P:

# تعريف ال BOM



والآن أريدك أن تعطيني رأيك بهذه الصيغة:

```
window.document.getElementById("header");
```

هل تشبه صيغة تعرفها؟

وهل تختلف عن `document.getElementById("header");`؟

الجواب لا تختلف، والسبب في ذلك يعود أننا قلنا في خضم حديثنا ال

`window` هي ال `parent` وأنها تمثل `object` وأن ما تحتها يصبح تلقائياً

تابع لهذا ال `object` .. `^_^`، وبهذا فإن الثانية هي ذاتها الأولى `^_^`

# JS Popup



ال `Window ^_^`، في التعريف قمنا بالحديث عن ال `window` وتعرفنا عليها، والآن لنتحدث عن أشهر الدوال والخصائص التي يمكننا استخدامها، والتي تدرج بالنهاية تحت موضوع ال `window`.. وأول `object` هو ال `JS`

## Popup

- (1) `alert()`: تستخدم هذه لإظهار رسالة على شكل `Popup ^_^`.
- (2) `confirm()`: تستخدم هذه لإظهار رسالة على شكل `Popup` فيها أزرار للتأكيد أو الإلغاء `^_^`.
- (3) `prompt()`: تستخدم هذه لإظهار رسالة على شكل `Popup`، وتحتوي على مربع نص لإدخال قيمة، مع زر للموافقة والإلغاء.

# JS Popup



aneeshikmat@2nees.com أنيس حكمت أبو حميد

والآن، لنشاهد مثالاً معاً :^\_^

```
<input type="button" value="alert box" onclick="alertFunction();" />
<input type="button" value="prompt box ((input at least 3 char))" onclick="promptFunction();" />
<input type="button" value="confirm box" onclick="confirmFunction();" />
```

```
<script>
```

```
function alertFunction() {
    window.alert("Hello ^_^ I am Anees ^_^ window.alert");// we can remove window to be more
    alert("Hello ^_^ I am Anees ^_^ alert");
}
```

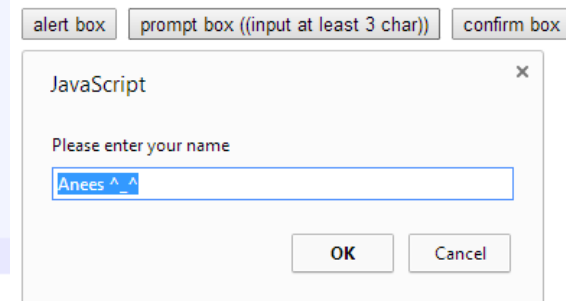
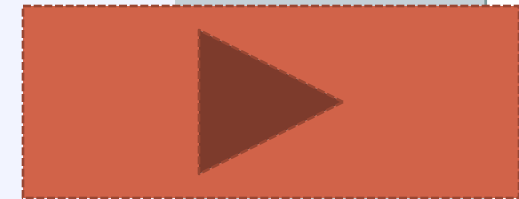
```
function promptFunction() {
    var name = prompt("Please enter your name", "Anees ^_^");
    if (name.length >= 3) {
        document.write("<b> Hello " + name + "</b>");// Remember (write will be remove content)
    }
}
```

```
function confirmFunction() {
    var conValue = confirm("Are you Anees ?");

    if(conValue){
        document.write("<b>You Are Welcome ^_^</b>");
    }else{
        document.write("<u>sorry, but you are welcome too :P</u> ");
    }
}
```

```
</script>
```

طبعا يمكنك كتابة أي من الدوال السابقة  
مع أو بدون  
window ^\_^



# JS Timing



إن التعامل مع الوقت في الجافا سكربت من الأمور المهمة جدا، والرائعة جدا، والتي تنقذ من مشاكل برمجية كثيرة، ولهذا سنتطرق الآن بإذن الله تعالى الى دالتين مهمتين وهما:

(1) **setInterval()**: تستخدم هذه الدالة لتنفيذ شيفرة برمجة مرة تلو المرة، دون توقف بنائاً على فترة زمنية محددة، تقوم أنت كمبرمج بتحديدتها، كما أنه يتوفر لك خيار برمجي آخر لإنهاء هذا التكرار في لحظة معينة <sup>٨٨</sup> والصيغة العامة هي:

```
setInterval("javascript function", milliseconds);
```

وبكل تأكيد فإن ال window موجودة لكن قمنا بحذفها لجواز ذلك <sup>٨٨</sup>، كما ذكرنا سابقا وكما فعلنا مع ال alert، ولن أذكرك بهذه المعلومة مرة أخرى P:

# JS Timing



ملاحظات على `setInterval`:

(1) بالصيغة العامة لهذا ال `Function` نجد أن “ javascript `function` ” تعني أنك يجب أن تستخدم `function` ولا يجوز استخدام غير ال `function` .^\_^

(2) `Milliseconds` هو الوقت المطلوب لكل دورة (أي الوقت اللازم ليتم تنفيذ الشيفرة البرمجية في كل مرة، فلو قمنا بوضع 1000، فهذا يعني 1 ثانية، أي أن الشيفرة البرمجية سيتم تنفيذها كل ثانية مرة واحدة .^\_^  
(1s = 1000 ms)

(3) تستخدم `clearInterval()` لإيقاف ال `setInterval` والصيغة العامة لها هي: `clearInterval(intervalVariable)`، وال `intervalVariable` هذه تعني المتغير الذي اسند اليه دالة الوقت مثل `intervalVariable = setInterval("...", 2000);`



# JS Timing



**2.** `setTimeout()`: تقوم هذه الدالة على تنفيذ الشيفرة البرمجية التي تحتويها لمرة واحدة فقط، ويتم تحديد الوقت اللازم حتى يتم تنفيذ الشيفرة البرمجة بال `ms` أيضا ..  
الصيغة العامة:

```
setTimeout("javascript function", milliseconds);
```

بكل تأكيد ما تم ذكره من ملاحظات على `setInterval` ينطبق هنا أيضا، لكن الدالة المسؤولة هنا عن الغاء هذا المؤقت هو  
`clearTimeout(timeoutVariable)`

والآن .. لنذهب ونشاهد أمثلة متنوعة على ال `JS Timing` ^\_^

# JS Timing



aneeshikmat@2nees.com أنيس حكمت أبو حميد

مثال:

```
<input type="button" value="Clear Interval" onclick="clearIntervalFunction();" />
<input type="button" value="Start Time Out" onclick="startTimeFunction(this);" />
<input type="button" value="Clear Time Out" onclick="clearTimeFunction();" />
```

```
<script>
var intervalVariable = setInterval(function () {
    var randomIndex = Math.floor((Math.random() * 3) + 0);
    var color = new Array("red", "blue", "green");
    document.body.style.backgroundColor = color[randomIndex];
}, 500);
```

```
function clearIntervalFunction() {
    clearInterval(intervalVariable);
}
```

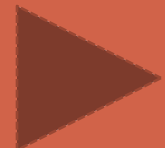
```
var startTimeVariable;
function startTimeFunction(id) {
    alert("after 3 second, button color will be changed, ((( If you click on clear after
    startTimeVariable = setTimeout(function () {
        id.style.color = "#0000ff";
    }, 3000);
}
```

```
function clearTimeFunction() {
    clearTimeout(startTimeVariable);
}
```

Clear Interval Start Time Out Clear Time Out

لون متغير بشكل عشوائي

إذا أردت تطبيق المثال على **clearTimeout** فعليك أن تنقر على زر ال **Clear Time Out** مباشرة بعد الضغط على **Start Time** (تملك 3 ثواني فقط (P:



# خاطرة



اللهم قونا بالحق وللحق وفي الحق  
وأعدنا من شرور أنفسنا

سبحانك اللهم وبحمدك  
أشهد أن لا إله إلا أنت  
أستغفرك وأتوب إليك

# JS Screen



يستخدم هذا ال Object لجلب المعلومات الخاصة بشاشة العرض للمستخدم، وأهم هذه المعلومات هي الإرتفاع والعرض الخاص بالشاشة ^\_^.

الخصائص التي سنتحدث عنها:

- (1) `screen.width`: وتستخدم هذه لجلب العرض الخاص بالشاشة باليكسل.
- (2) `screen.height`: وتستخدم هذه لجلب الإرتفاع الخاص بالشاشة باليكسل
- (3) `screen.availWidth` و `screen.availHeight` تستخدم هذه لجلب الإرتفاع والعرض الخاص بالشاشة باليكسل “ لآكن يتم طرح الحجم الخاص ببعض مزايا العرض لنظام التشغيل مثلا من الحجم، مثل ال `windows` `taskbar` ^\_^

والآن .. انتقل للشريحة التالية لمشاهدة المثال: ^\_^

# JS Screen



مثال:

```
<div id="output"></div>
<script>
  var outPut = "";
  outPut += "Screen Width: " + screen.width;
  outPut += "<br />Screen Height: " + screen.height;
  outPut += "<br />Screen availWidth: " + screen.availWidth;
  outPut += "<br />Screen availHeight: " + screen.availHeight;

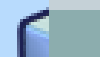
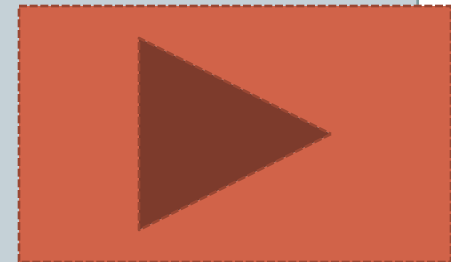
  document.getElementById("output").innerHTML = outPut;
</script>
```

Screen Width: 1366  
Screen Height: 768  
Screen availWidth: 1366  
Screen availHeight: 728

لاحظ الإختلاف بالإرتفاع  
وذلك بسبب وجود شريط  
المهمات بالأسفل كما ترى

^ ^

—



# JS Location



يستخدم هذا ال object لجلب العنوان الخاص بالصفحة (URL)، وإعادة توجيه الصفحة الى صفحة أخرى، كما يمكن الحصول على معلومات محددة من ال URL مثل ال host name وال path الخاص بالصفحة ٨٨

والآن لنشاهد أهم الدوال التي سنقوم باستخدامها:

- (1) `location.href`: تستخدم لجلب ال URL الخاص بالصفحة.
- (2) `location.hostname`: تستخدم لجلب ال hostname مثل [www.2nees.com](http://www.2nees.com)
- (3) `location.pathname`: تستخدم لجلب ال path الخاص بملف `/js/aneesCustomScript.js`

# JS Location



(4) `location.protocol` يستخدم لجلب ال `protocol` الخاص بالصفحة  
مثل `http` `^_^`

(5) `location.assign()` تستخدم لعمل `load` لملف آخر في الملف  
الحالي...

أنيس حكمت أبو حميد aneshikmat@2nees.com

```
Location href: http://2nees.com/ar  
Location hostname: 2nees.com  
Location pathname: /ar  
Location protocol: http:
```

والآن لنشاهد الأمثلة: مثال 1:

```
var outPut = "";  
outPut += "Location href: " + location.href;  
outPut += "<br />Location hostname: " + location.hostname;  
outPut += "<br />Location pathname: " + location.pathname;  
outPut += "<br />Location protocol: " + location.protocol;  
  
document.getElementById("output").innerHTML = outPut;
```

# JS Location



مثال 2:

```
<input type="button" value="Load new document" onclick="loadNewDoc()">
<script>
    function loadNewDoc() {
        location.assign("http://www.2nees.com");
    }
</script>
```

هنا سيتم عمل load ل 2nees.com في ال document الذي أنا فيه، وهذا يعني أنه سيتمكنني الضغط على زر back للرجوع في المتصفح خطوة للخلف، وهذا ما يختلف عن استخدام ال **location.replace** ! **location.replace** تقوم بتبديل نفس ال document الذي أنا فيه، وهذا يعني عدم تخزين ال Session الخاص بال hestory وبهذا لن يكون هناك زر back

^ ^  
\_



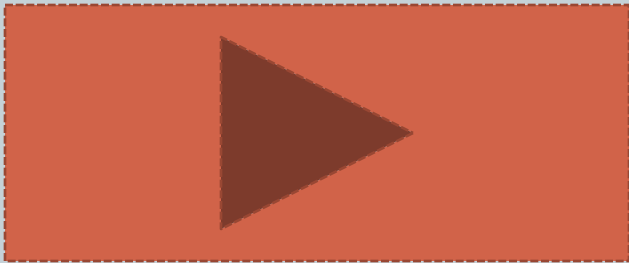
# JS Location



مثال 3:

```
<input type="button" value="Redirect To new Page" onclick="redirectToNewUrl()">
<script>
  function redirectToNewUrl() {
    window.location = "http://www.2nees.com/parallex";
    //location.href = "http://www.2nees.com/parallex";
    //location.replace("http://www.2nees.com/parallex");
  }
</script>
```

شاهد كيف يمكنك إعادة توجيه الصفحة الى الرابط الذي ترغب به ^\_^  
والآن، لنشاهد جميع الأمثلة معا ^\_^



# JS History



هذا ال object يحتوي على ال History الخاص بالمتصفح، وطبعاً ولخصوصية هذا الأمر خصوصاً لدى المستخدمين، فإنه قد يحتوي نوعاً من القيود في حرية الوصول إلى الروابط المحفوظة في ال Session History وهناك 2 method يمكننا استخدامها وهما:

(1) **history.back()**: وهذه تستخدم للرجوع إلى الصفحة السابقة (أي تعمل كما يعمل زر back في المتصفح).

(2) **history.forward()**: وهذه تستخدم للذهاب للصفحة التالية (صفحة كنت فيها ثم عدت إلى صفحة قبلها، وتشبه عمل زر forward في المتصفح) ^\_^ ... والآن لنذهب ونشاهد مثلاً ^\_^

# JS History

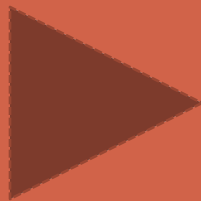


مثال:

```
<input type="button" value="First - Click here then click on browser back btn" onclick="redirectToNewUrl()">
<input type="button" value="Second - Go Forward" onclick="goForward()">
<script>
  function redirectToNewUrl() {
    window.location = "http://www.2nees.com";
  }

  function goForward() {
    history.forward()
  }
</script>
```

لتطبيق المثال، أولاً قم بالضغط على زر إعادة التوجيه، ثم قم بالضغط على زر  
Back... بعد ذلك قم بالضغط على زر Go Forward ^\_^



# خاطرة



اللهم قونا بالحق وللحق وفي الحق  
وأعدنا من شرور أنفسنا

سبحانك اللهم وبحمدك  
أشهد أن لا إله إلا أنت  
أستغفرك وأتوب إليك

# JS Cookies



هذا الموضوع مهم جدا، فهو يتحدث عن الكعكات، والكعكات طعام لذيذ مع الشاي P: !\_! بعذر شكلي رحلت لموضوع ثاني P: ^\_^

ال Cookies هي مجموعة من البيانات التي يتم تخزينها في ملف TXT file على جهاز الحاسوب الخاص بال user، وسبب إبتكار هذه الفكرة الرائعة، الحاجة الى طريقة نستطيع من خلالها الحصول على المعلومات الخاصة بالمستخدم بعد مغادرته للموقع، لأن السيرفر سيقوم بعمل shutdown لل connection عندما يتم الإنتهاء من عرض البيانات على المتصفح، وبهذا يكون دور ال Cookies الرئيسي هو “ حفظ معلومات المستخدمين، لتقديم خدمة أفضل (معلومات المستخدمين قد تكون المعلومات التي يضعها الموقع للمستخدم لتقديم خدمة أفضل) ” .... (ولن أتحدث هنا عن الجانب السلبي P: )

# JS Cookies



كيف يتم حفظ ال Cookies داخل الملف؟

يتم حفظ ال Cookies داخل الملف على شكل Name - Value بحيث يمثل ال Name الإسم البرمجي (الإسم الذي يمثل المعلومة التي نرغب باستدعائها مثل userName) ، وال Value هي القيمة الخاصة بهذا الإسم ^\_^.

مثال على cookies:

name = Anees

age = 24

email = aneeshikmat@2nees.com

# JS Cookies



والآن، لنبدأ حديثنا حول الموضوع الأول في ال Cookies وهو  
أولاً: Create a Cookie :

يمكننا إنشاء ال Cookie من خلال الجافا سكربت، وذلك عن طريق استخدام  
document.cookie، الأمر سهل، لكن قبل أن أكمل، ما هو رأيك بال  
document.cookie.. هل هي DOM أم BOM ؟

الجواب يجب أن يكون DOM بكل تأكيد ^\_^، ال Cookie بذاته BOM،  
لكن ما نستخدمه لنحفظ المعلومات من الصفحة وغيرها من العمليات سيكون  
من خلال ال document أي سيكون DOM.. ^^ (وجب التنبيه)

# JS Cookies



عند إنشاء ال cookies يمكنك التحكم بعدة أمور، مثل المدة الزمنية لهذا ال Cookie، وامتداد الصفحة الخاص بهذا ال Cookie، ويمكن تحديد ال Domain، ويمكن تحديد ان هذا ال Cookie لا يجب أن يعمل الى مع ال Secure Protocol مثل ال Https، وطبعا جميع هذه الامور خيارات إفتراضية، تلزم أنت فقط ب ال Name – Value ^\_^

مثال: (انتبه جيدا لما ورد فيه، ثم انتقل للشريحة التالية):

```
document.cookie="name=Anees Hikmat Abu Hmaid";  
document.cookie="username=anees ^_^; expires=Tue, 08 Sep 2015 06:45:34 GMT";  
document.cookie="website=2nees.com ^_^; expires=Tue, 08 Sep 2015 06:45:34 UTC";  
document.cookie="yourname=saed ^_^; expires=Tue, 08 Sep 2015 06:45:34 UTC; path="/";
```



# JS Cookies



## ملاحظات:

- (1) أول صيغة هي الصيغة الأبسط، والإزامية لإنشاء أي **Cookie**.
- (2) ثاني صيغة قمنا بتحديد ال **Expire date**، ولاحظ أن طريقة التعريف تتم من خلال كتابة ; للفصل بين الخيارات، ومن ثم وضعنا ال **expire**.
- (3) ال **expire date** يجب أن يكون **UTC** أو **GMT** (وكلاهما واحد)، وهذا يعني أنه يمكنك استخدام **toGMTString** أو **toUTCString** للحصول على الوقت المطلوب بهذه الصيغة **^\_^**.
- (4) في حالة لم تقم بتحديد المدة لحياة ال **cookie**، فإنه ينتهي بمجرد إغلاق المتصفح.

# JS Cookies



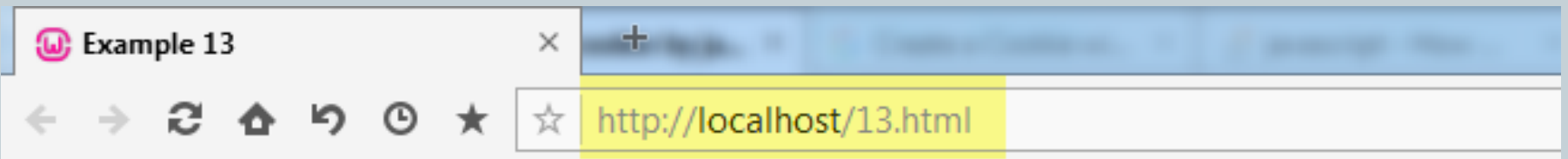
ثانيا: كيف يمكننا جلب ما لدينا من cookie؟

الجواب باستخدام `document.cookie` ^\_^ أيضا P: كيف ذلك؟

شاهد المثال: (لاحظ أن جميع ال Cookie التي لدي سيتم جلبها على شكل

(string

```
document.getElementById("output").innerHTML = document.cookie;
```



**Open Example on Local Server**

name=Anees Hikmat Abu Hmaid; username=anees ^\_^; website=2nees.com ^\_^; yourname=saed ^\_^

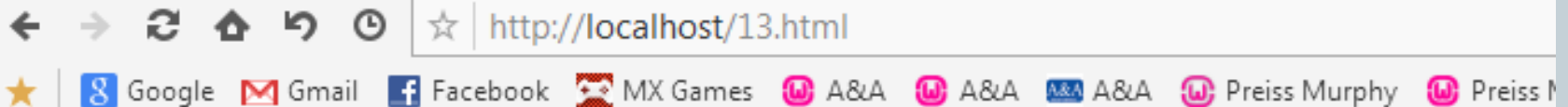
# JS Cookies



ثالثا: كيف يمكننا تعديل قيمة Cookie موجودة ؟  
عملية التعديل على ال Cookie تعتمد على name وهنا website ،  
ويمكنني تغيير القيمة والوقت وال path .. الخ

```
function changeCookieValue()  
{  
    document.cookie="website=http://www.2nees.com; expires=Tue, 08 Sep 2016 06:45:34 UTC";  
    document.getElementById("output").innerHTML = document.cookie;  
}
```

Example 13



Open Example on Local Server

name=Anees Hikmat Abu Hmaid; username=anees ^\_^; yourname=saed ^\_^; website=http://www.2nees.com

change cookie(website) value

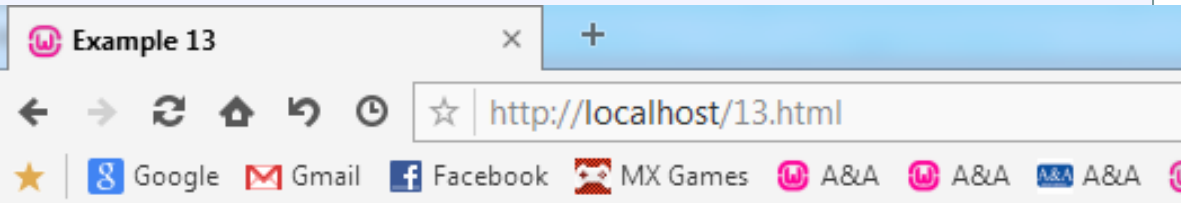
# JS Cookies



رابعاً: كيف يمكننا حذف Cookie موجود ؟  
عملية الحذف لل Cookie تعتمد على name أيضاً، ونقوم بتصفير الوقت،  
ليكون ال Cookie وكأنه منتهي .. شاهد المثال:

```
function deleteCookieValue ()  
{  
    document.cookie="yourname=; expires=Thu, 01 Jan 1970 00:00:00 UTC";  
    document.getElementById("output").innerHTML = document.cookie;  
}
```

لا يوجد قيمة لل  
yourname



Open Example on Local Server

name=Anees Hikmat Abu Hmaid; username=anees ^ ^; website=2nees.com ^ ^

delete cookie(yourname)

لاحظ أن عملية الحذف قد تمت ل

yourname

# JS Cookies



والآن، لقد فهمنا الشكل الأساسي للتعامل مع ال `cookie`، هل هذا الشكل منطقي أم فيه مشكلة نوعا ما؟!

هذا الشكل الذي ذكرناه هو الشكل البسيط المفيد في حالة وجود `cookie` واحدة -\_-، لأن هذا الشكل كما تلاحظ يقوم بإرجاع الاسم والقيمة فقط، ولجميع العناصر، ولم أقم بتحديد أيضا العنصر الذي أُرغب به (بحث)، وجميع ال `cookie` التي سيتم إضافتها، فعليا ستقوم بإضافة ال `cookie` الى `file` مجددا وليس عملية `overwritten`.. لهذا سنقوم باستخدام `3 Function` مشهورات للقيام بهذه العمليات... لنشاهد معا ذلك ^\_^

# JS Cookies



\* أول ال Function هو المسؤول عن ال Set Cookie:

```
function setCookie(cname, cvalue, exdays) {  
    var d = new Date();  
    d.setTime(d.getTime() + (exdays*24*60*60*1000));  
    var expires = "expires="+d.toUTCString();  
    document.cookie = cname + "=" + cvalue + ";" + expires;  
}
```

هذا ال Function بأبسط صورته، ويقوم بكل بساطة بإنشاء ال Cookie بما نرسل له من متغيرات، كما يمكنك “التعديل عليه” حسب الرغبة، فيمكنك إضافة متغير ال Path أو إضافة شرط بخصوص ال expires اذا لم أرغب بكونها الزامية... الخ، وطريقة الإستخدام هي `setCookie('name', 'abc', 2)` وهذا يعني قم بإنشاء cookie لمدة يومين، اسمه name وقيمه abc ...

# JS Cookies



\* ثاني ال Functions هو ال get Cookie :

```
function getCookie(cname) {  
    var name = cname + "=";  
    var ca = document.cookie.split(';');  
    for(var i=0; i<ca.length; i++) {  
        var c = ca[i];  
        while (c.charAt(0)==' ') c = c.substring(1);  
        if (c.indexOf(name) == 0) return c.substring(name.length,c.length);  
    }  
    return "";  
}
```

هذا ال Function يقوم على إرجاع ال cookie الذي أرغب بالحصول عليها من خلال ال name، مثل `getCookie('name')`

# JS Cookies



ملاحظة: طريقة عمل ال **Function** السابق بسيطة إذا تتبعتها، فهي تقوم على أخذ الإسم الذي قمت بكتابته (للبحث عنه) ثم عملية **concat** مع إشارة ال **=** وهذا يعني أنني أخذت المقتطف الأول من ال **Cookie** مثل **website=** بعد ذلك أقوم بتحويل النص الى مصفوفة من خلال ال **;**، فكما تذكر هي الفاصل بين كل من القيم التي يمكننا إستخدامها، ثم قمنا بعمل جملة دوران، بطول المصفوفة الناتجة، ثم أردنا بالتأكد أنه لا يوجد فراغ قبل ال **..=website** لذلك قمنا بوضع شرط، اذا وجدت فراغات قم بحذفها...وبعد ذلك قم بقطع النص من الموقع الذي تنتهي عنده ال **=** الى نهاية القيمة الخاصة بهذا الإسم

```
name=Anees Hikmat Abu Hmaid; username=anees ^_^; website=2nees.com ^_^; yourname=saed ^_^
```

... ^\_^

Elements Resources Network Sources Timeline Profiles Audits Console

```
> console.log(getCookie('name'));  
Anees Hikmat Abu Hmaid
```

مثال



# JS Cookies



ثالث ال Function يقوم على التحقق من قيمة معينة لتنفيذ شرط معين بعدها، وعلينا نحن ما قمنا بكتابة ال Cookie الى لنستخدمها في أمر ما، وعملية جلب ال Cookie التي تعلمناها في ال Function رقم 2 يجب أن لا تكون عبثية، وهنا يظهر دورها `^_^`، في هذا ال Function لو قمت بارسال ال name مثلا، فسيتم أولا البحث عنه لجلبه من خلال ال get، إن لم يكن موجودا، ستظهر لك القيمة `theres no name`، أما اذا كانت موجودة

فسيطبع هنا `anees`

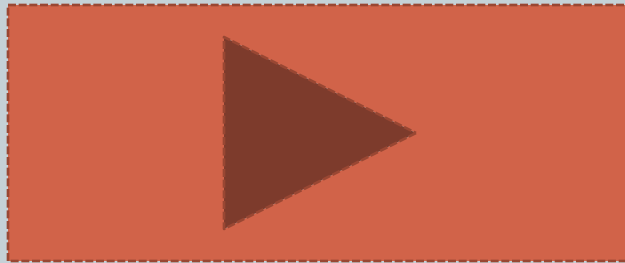
```
function checkCookie(cname) {  
    var name=getCookie(cname);  
    if (name!="") {  
        alert("Welcome ^_^ " + name);  
    }else{  
        alert('theres no name like this !');  
    }  
}
```

# JS Cookies



والآن، لنشاهد مجموع الأمثلة السابقة معا حول هذا الموضوع، ولا تنسى أن تقوم بتشغيل هذا المثال على `local server`...

ملاحظة: هذه ال `3 Function` تمثل 3 أفكار رئيسية، لكن يمكنك البحث أو تطوير `Function` خاصة بك للقيام بمثل هذه العمليات، فمثلا لدي 10 فانكشن قمت بكتابتها لتخدمني بالشكل المناسب أو الذي أرغبه.. الأمر ليس حفظا، وإنما “ منطق “ تفكير `^_^`  
شاهد الأمثلة:



# خاطرة



اللهم أنت الحليم فلا تعجل، وأنت الجواد فلا تبخل، وأنت العزيز فلا تذلل، وأنت المجير  
فلا تضام، وأنت المنيع فلا ترام، وأنت على كل شيء قدير.

اللهم لا تحرمني سعة رحمتك، وسبوغ نعمتك، وشمول عافيتك، وجزيل عطائك، ولا  
تجازني بقبيح عملي، ولا تصرف وجهك الكريم عني برحمتك يا أرحم الراحمين .

اللهم إليك مُدَّت يدي، وفيها عندك عظمت رغبتي، فاقبل توبتي، وارحم ضعف قوتني،  
واغفر خطيئتي، واقبل معذرتي، واجعل لي من كل خير نصيباً وإلى كل خير سبيلاً.

اللهم آمين

أدعية رائعة منقولة من موقع [mawdoo3](http://mawdoo3.com)

# الخاتمة



الحمد لله رب العالمين، أن من علينا و عليكم بواسع منه وفضله، ووفقنا لأن نكمل ما بدأناه في أول هذه الدورة المتواضعة، والحمد لله رب العالمين أن وفقنا لأن نتعلم ونكتسب معا الكثير والجديد، ورزقنا الهمة لأن نكتب و ندرس هذه الصفحات المعدودة، فالحمد لله رب العالمين...

إخواني الأعزاء، هذا العمل المتواضع ما هو الى عمل بشري، والعمل البشري لا يصل الى الكمال والتمام، وإنما بذلنا ما بوسعنا بإذن الله تعالى بأن نتحرى أدق معاني الدقة وانتقاء الكلمات المناسبة في هذه الشروحات، لذلك، فإن أخطأنا فهذا تقصير من أنفسنا، وإن أصبنا، فبفضل من الله تعالى ورضوانه، والحمد لله رب العالمين صاحب المن والفضل.

# الخاتمة



بخصوص هذه الدورة فهي الجزء الثاني من دورة الجافا سكربت، والذي أفترض أنك اطلعت عليها لأهميتها، والآن سننتقل بإذن الله تعالى الى مستوى آخر في الجافا سكربت، لكن لحين ذلك الوقت، أنصحك بالإستمرار والمتابعة والقراءة من المصادر المتنوعة والمختلفة، سائلا المولى عز وجل أن يوفقنا لكل خير...  
دورات أخرى قد تهتمك:

**HTML CSS HTML5 CSS3 JSON AngularJS Bootstrap  
SASS PHP-Advance-101 Drupal 7 jQuery jQuery UI**

جميع هذه الدورات تجدونها بإذن الله في موقع :

[www.2nees.com/courses](http://www.2nees.com/courses)



وأخر دعوانا أن الحمد لله رب العالمين