بسم الله الرحمن الرحيم

السلام عليكم و رحمة الله تعالى و بركاته

نبدأ في هذا الجزء الجديد بإذن الله تعالى من سلسلة المعلومات المهمة لكل مبرمج حيث نتطرق لمعلومات جديدة و مواضيع شيقة ، فهي معلومات لا غنى عنها لأي مبرمج فاتمنى لكم قراءة ممتعة و نبدأ على بركة الله .

1 – البرمجة الكائنية

في وقتنا الحالي اي شخص يريد الدخول لعالم البرمجة الواسع لابد و ان مرت عليه هذه العبارة التي لا يفهم منها المبتدأ شيء في بداية طريقه ، فما هي البرمجة الكائنية ؟؟؟

البرمجة الكائنية هي نمط من انماط البرمجة العديدة تعتبر هي الافضل و اغلب لغات العصر تدعمها لما لها من فوائد كثيرة في عملية البرمجة ، ربما نسمع عن بقية الانماط مجرد سماع مثل البرمجة الهيكلية و الاجرائية و المنطقية .... الخ و الكثير من الانماط التي اختفت و لم تعد تستعمل لكن ما زالت هناك لغات ربما تستعملها ، و كل لغة برمجة تدعم نمط معين او عدة انمط فمثلا لغة C++ الشهيرة تعتبر اجرائية و كائنية في نفس الوقت .

طيب دعونا نركز حديثنا على البرمجة الكائنية طبعا نحن لن نتطرق للاكواد و الشرح لكن سنكتفي بتعرفها و معرفة اهم خصائصها .

تعتمد البرمجة الكائنية على اسلوب تقسيم البرنامج لاجزاء تسمى كل جزئية (صنف - فئة- Class) و يعتبر هذا الصنف هو اساس البرنامج مثلا عندما ننظر لواجهة اي برنامج نجد العديد من العناصر مربعات نصوص و ازرار و قوائم و حتى النافذة نفسها كل ما في البرنامج يسمى كائن يتم اشتقاقه من الصنف بغض النظر عن طبيعة عمله ، طيب لما هذا النظام ما الفائدة منه ؟ تكمن فائدة البرمجة الكائنية في سرعة الانتاجية فانت لا تطر في كل مرة لاعادة كتابة نفس الاكواد بالاضافة للتنظيم و الحماية .

مكونات الفئة

اي فئة برمجية هو عبارة عن هيكل من المتغيرات (حاويات البيانات) و الدوال (الوظائف) بحيث نستطيع اشتقاق عدة كائنات من نفس الصنف ، لناخذ مثلا الازرار يعني لو برنامجك فيه عدة ازرار ، انت في البداية تصنع فئة اساسية تسميها (الزر) طبعا بها مجموعة من الوظائف و البيانات مثل الالوان و دوال الرسم و غيره من الدوال المهمة و عند الحاجة لعمل زر جديد نشتق من الصنف كائن و نقوم بالتعديل في خصائصه لونه ،اسمه، حجمه ....الخ ، و لو لاحظت في بيئة التطوير للواجهات الرسومية تجد ان بيئة التطوير تقوم بعمل شكل واحد فقط للازرار و هي الصفات الاساسية التي اخذها الكائن من الصنف ، و مهما صنعت من ازرار فهي تعطيك نفس النتيجة تماما مثل مصنع السيارات فهو في كل عام يستخدم فئة اساسية موجودة عنده يقوم باشتقاق كائن جديد منه و يعدل في خصائصه و ينتج لك الموديل الجديد ، اهم نقطة في الموضوع هو ان كل ما تراه امامك في البرامج هو كائنات .

اهم خصائص البرمجة الكائنية

AGASHE

- الوراثة : يعتبر مفهوم الوراثة جزء لا يتجزء من البرمجة الكائنية فكل صنف جديد يمكنه ان يرث مجموعة من الصفات من صنف آخر مثلا في حديثنا عن السيارة نستطيع ان ننشئ صنف جديد اسمه الشاحنة فهو يرث الصنف الاساسي (السيارة) و يضيف عليه ببعض الخصائص الجديدة مثل القدرة على الجر او اقصى حمولة .... الخ .

- التغليف : أيضا من الامور المهمة في البرمجة الكائنية هي التغليف فانت تتمكن من حماية بياناتك و عدم اظهارها للمستخدم بصورة مباشرة عن طريق استعمال التغليف .

- تعدد الاوجه : تشبه الوراثة في كيفية عملها الا ان تعدد الاوجه تعمل على جعل الدوال تعمل مع عدة اصناف و هذا يختلف عن الوراثة العادية قليلا .

2- محرك الالعاب

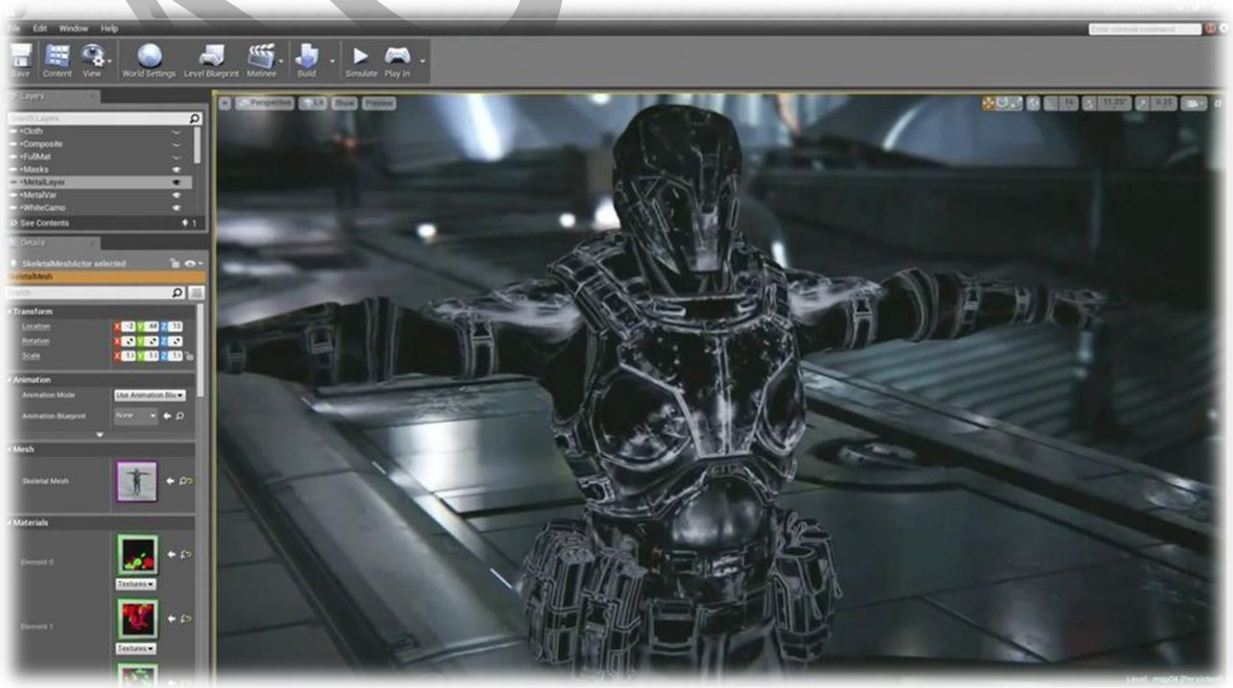
عبارة ربما تمر عليك عزيزي القارئ اذا ما بدأت الدخول في عالم تطوير الالعاب و هو شيء وارد لذا سوف نعطي نبذة مختصرة

كلنا يعرف ان الالعاب تتكون من رسومات ، اصوات ، ذكاء صناعي ، فيزياء ، بعض البرمجة

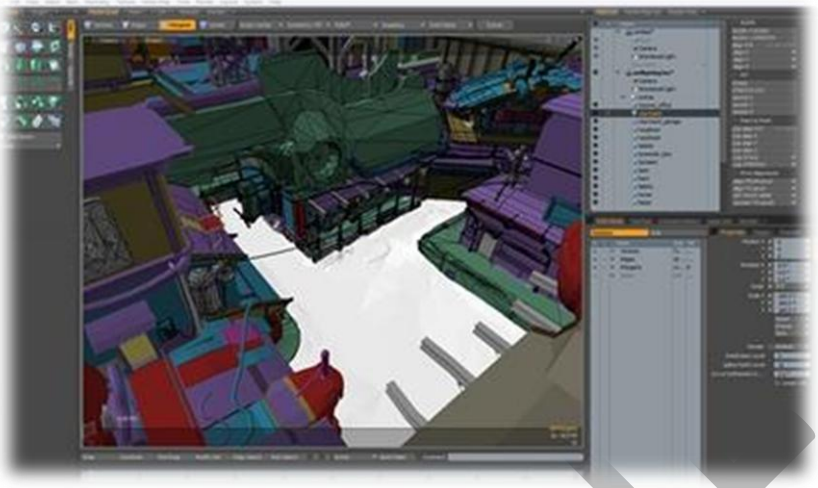
و تبدا هنا المعضلة فكل جزء يحتاج لعمل و مجهود كبير حتى تحترفه و هنا جاء دور محرك الالعاب فهو اطار عمل يسهل على المطور القيام بالاجزاء المختلفة للعبة و تقريبا كل شركات الالعاب تمتلك محرك خاص بها يختلفوا من حيث سرعتهم و دعمهم للالعاب ثنائية و ثلاثية الابعاد و اللغات المستخدمة في برمجتهم يعني تماما مثل بيئة التطوير لكن بشكل اوسع قليلا و تعتبر صناعة محركات الالعاب موضوع كبير و خطير و يحتاج لخبرة كبيرة جدا في البرمجة الكائنية و خاصة موضوع الفنات فالالعاب لا تخلو منها ، تكمن مشكلة المحركات في شيء واحد و هو الثمن الباهظ الذي يجعل المطورين الهواة يطرون الى استخدام محركات اقل جودة و لكنها مجانية و هذه الأخيرة ربما تساعدهم في اطلاق العنان لابداعتهم لكن اذا ما حاولوا التربح منها فان الشركات ربما تطالبهم بالدفع او البعض ربما يتشارك الربح مع تلك الشركات بحيث يكون لها نسبة في الالعاب التي تنتجها .

اشهر محركات الالعاب

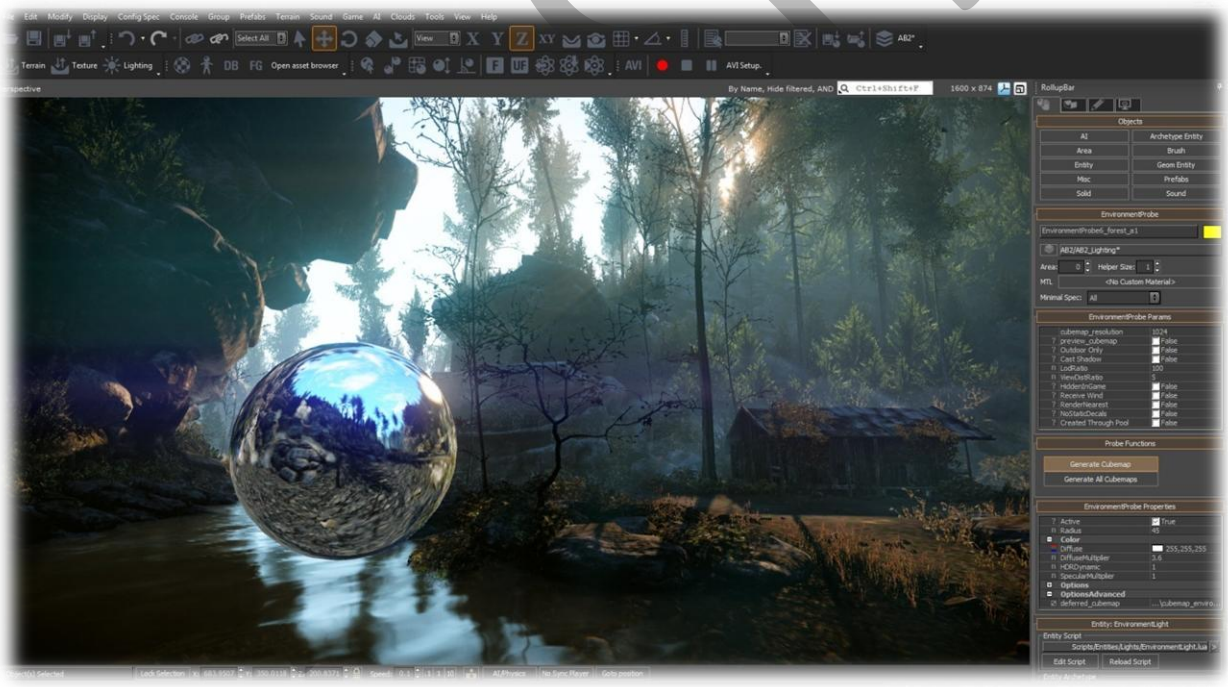
- Unreal engine : يعتبر واحد من افضل محركات الالعاب في العالم و يوجد عدد كبير من الالعاب الحديثة التي انتجت به



- **Rage**



- **cryEngine**



طبعا هذه بعض اقوى و اشهر المحركات التي تصنع به العاب لمنصات مشهورة مثلPS4

لكن عندنا ايضا

XNA + C# - Game maker - Unity

3- اللغات السكربتية

في عالم البرمجة الواسع يجب ان تمر على كلمة (سكربت– Script) و اللغات السكربتية فهي اساس لا غنى عنه لدى المبرمجين و نرى هذه العبارة خاصة في بداية المشوار عندما تنوي الدخول في عالم البرمجة و لذا يتحتم علينا ذكر بعد التفاصيل المهمة عن هذه النقطة .

1 - ما معنى انها لغة سكربتية(تعتمد على كتابة الاسكربتات) ؟

الاسكربت هو مقطع برمجي ، كود يتم كتابته لغرض معين ، في اللغات العادية هذا المقطع يجب ان يترجم بالكامل قبل ان ينفذ اما في اللغات السكربتية فلا يشطرت ذلك هذا يعني ان السطر الذي يمر يترجم و ينفذ ، هذه اللغات في الغالب مبنية على لغات اخرى مثل C او java .

2 - ما المميز فيها ؟؟؟

اهم ما يميز هذه اللغات هو التضمين فانت تستطيع استعمال عدة اسكربتات من لغات مختلفة في نفس الوقت اكبر مثال على ذلك صفحات html بحر واسع من الاسكربتات من عدة لغات مختلفة تقوم انت بتضمينها ، و هذا يعطي هذه اللغات مرونة اكبر في التعامل .

3 - مجالات عملها :

يعتبر مجال تطوير المواقع هو المجال الافضل للغات الاسكربتية لما تتميز به من سرعة و خفة و مميزات اخرى عديدة

و لا ننسى تطبيقات الويب فهذا المجال الذي اصبح لا غنى عنه يعتمد اعتمادا كليا على لغات اسكربتية . بالنسبة لتطبيقات الويب فايضا اللغات الاسكربتية لديها القدرة على بناء تطبيقات سطح مكتب تماما مثل اللغات الاخرى .

4 - اشهر اللغات الاسكربتية :

Vbscript

Javascript

Python

Pearl

Ruby

ملحوظة : كلمة (اسكربيتة) هي مجرد صفة وضعتها للتوضيح اما الترجمة الصحيحة فهي لغات نصية

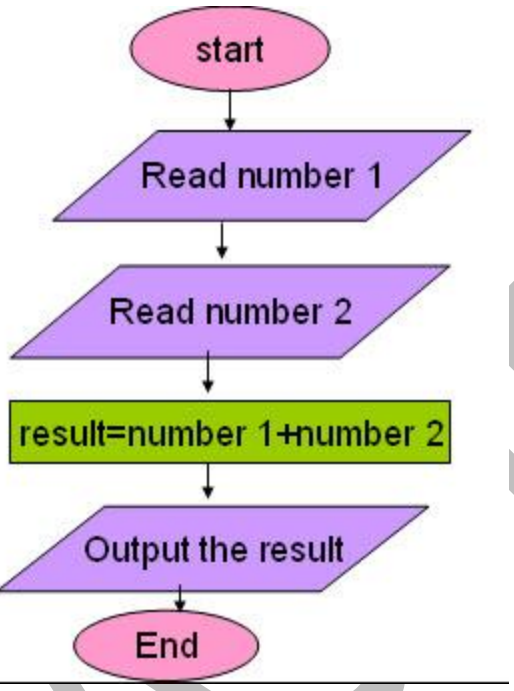
4 - لغات النمذجة

نحن جميعا مبرمجين نفكر في البرامج ثم نبدأ بالعمل لكن مع الاسف فهندسة البرمجيات لا تنظر للتنفيذ مباشرة كما ننظر نحن فالجميع يحب التطبيق لكن هندسة البرمجيات الحقيقية تملي علينا التخطيط بدقة كبيرة لجميع مشاريعنا و كما هناك لغات للبرمجة فهناك ايضا لغات للتخطيط و هذه اللغات تستخدم بشكل اساسي في المشاريع الكبيرة و المعقدة و لكن هذا لا يعني ان ليس باستطاعتك اساتخدامها في برنامج صغير ، تضيف لغات النمذجة توثيق اكبر للبرنامج قدرة كبيرة على التعديل بالاضافة تسهيل للمبرمجين الاخرين اذا ما تابعوا المشروع من بعدك .

AGASHE

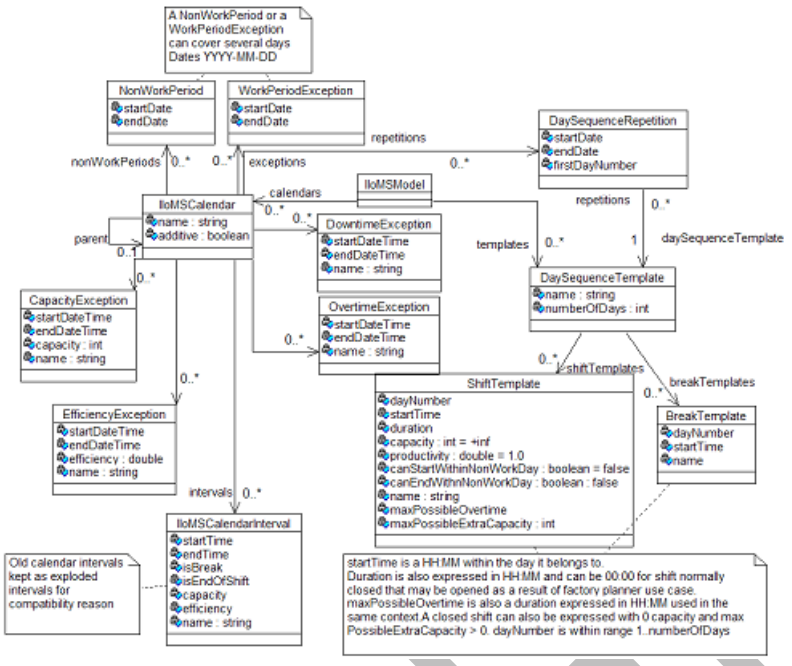
- **بعض لغات النمذجة :**

1 - خرائط التدفق : كلنا ربما درسنا هذا الدرس في النمدرسة في مادة الحاسب الالي ، حيث كان المطلوب هو عمل مجموعة من الاشكال نقوم فيها بوضع آلية عمل البرنامج و الخطوات المتبعة و تعتبر خرائط التدفق هي الاكثر انتشار والابسط بين لغات النمذجة .

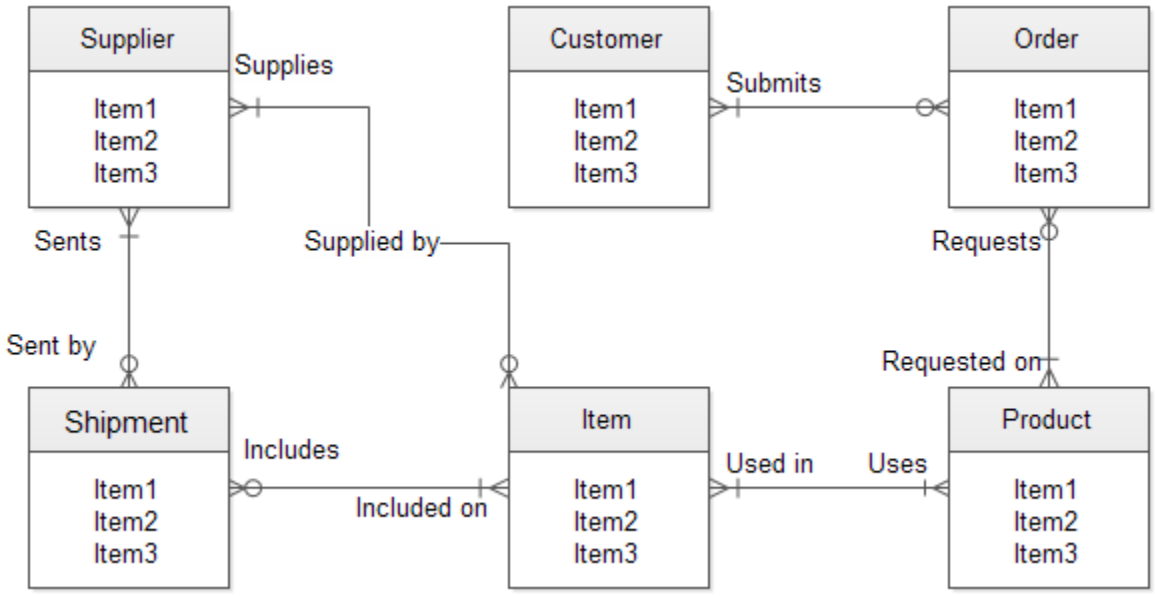


و من البرامج التي تستخدم في عمل خرائط التدفق : flowchart edraw - algodraw

2 - لغة النمذجة الموحدة (uml) : تعتبر الافضل في مجال النمذجة كبرى الشركات تعتمد عليها ، تعطيك قدرة اكبر على فهو و فهم البرنامج بالاضافة الى المساعدة على تقسيم البرنامج الى وحدات حتى يتمكن كل عضو في الفريق للبدء بالعمل .تعتبر هذه اللغة من اللغات القديمة فهي ترجع للستينيات حيث كان مفهوم البرمجةالكائنية في بداياته و كان لابد من وجود آلية للتخطيط تتلائم مع امكانيات مبدأ البرمجة الكائنية .



3 - لغة العلاقات بالكائنات (ERD) : تعتبر هذه الطريقة احد افضل الطرق لتنظيم قواعد البيانات حيث تفرض ان اي عنصر داخل قاعدة البيانات هو كائن و تربط بين هذه الكائنات ببعضها البعض عن طريق علاقات .



5 - الاخطاء البرمجية

احد اهم الجزيئات التي لا تخلو منها حيات المبرمجين سواء المحترفين او المبتدئين ، فنحن جميعا معرضين للخطأ و خاصة في هذا المجال الكبير فاخطاءه لا تنتهي . فالرموز كثيرة و المكتبات اكثر و كل لغة و لها قواعدها و لو كنت تجيد اكثر من لغة تصبح العملية في منتهى الروعة ☺ . و انواع الاخطاء عامة لجميع اللغات !!!

طيب عموما عندما يأتي وقت تجربة برنامجك و تفاجئت بان المترجم يعطيك خطأ

ابدأ بالتفكير بمنطقية و هدوء و لا تتوتر

1 - اخطاء لغوية (synatex error)

هي ابسط نوع من الاخطاء و يكون بسبب السهو او الاستعجال و غالبا تكون في آخر جزء تكتبه من من البرنامج حيث الفرح الشديد و التركيز القليل ☺ ، امثلته عديدة اهمها نسيان الفاصلة المنقوطة (;) و اغلب اللغات تعتمد عليها ايضا الاقواس فعبارات الشرط و التكرار المتداخلة هي تحدي كبير للمبرمج لترتيب كوده ، رمز التساوي في عبارة الشرط في بعض اللغات يكون مضاعف (==) ممكن تنسى و تكتبه مفرد ، اسماء جميع الدوال و المتغيرات و يجب ان تعرف ماذا كانت لغتك تفرق بين الحروف الكبيرة و الصغيرة ام لا ، الاشارات المنطقية مثلand و or .

كل ما سبق هو خطأ بسيط و شائع و في الغالب البيئة التي عمل عليها سوف تنبهك لهذا الخطأ بتلوينه

راجعت الكود جيدا لم تجد اخطاء لغوية بدأ توترك يزيد و تفقد اعصابك

لا مشكلة ننظر للنوع الثاني و نحافظ على تركيزنا

2 - اخطاء وقت التنفيذ (run time error)

يعتبر هذا النوع مكمل لما سبق ، فهنا البرنامج لن يعمل ايضا بسبب عدم تضمين الملفات المطلوبة بشكل صحيح

بفرض ان لدينا ملف جافا سكربت اسمه main.js موضوع في المجلد script عند تضمينك للملف في صفحة html

لابد ان تكتب script/main.js قد ينسى البعض و يكتب الاسم مباشرة او قد يكون هناك خطا في الكتابة مثل نسيان حرف

طبعا بيئة التطوير سوف تطبع لك رسالة مفادها ان احد الملفات غير مربوط بالبرنامج

تاكدت من الملفات و لم تجد مشاكل☹ !!!

تفقد اعصابك و تبدأ تضرب في الكمبيوتر المسكين و تلعن البرمجة

مع ان الخطأ كامن في الكود الذي كتبته

AGASHE

3 - اخطاء منطقية (logic error)

تعتبر اسوء نوع لا بيئة تنبهك و لا ملفات ناقصة و انما خطأ في الفكرة نفسها اشهر مثال القسمة على الصفر فانت لو صنعت آلة حاسبة و تعرضت لهذا الموقف سوف تتوقف تلقائيا نظرا لانه لا يوجد قسمة على الصفر في الكمبيوتر ،ترجع ايضا المشاكل البرمجية الى تكدس الكود و تداخل الحلقات و الشرط مما يفقدك تركيز و يشتت ذهنك فقد تبدل قيم في حلقات التكرار و لا تحصل على ناتجك ، و الحل الامثل لهذه المشكلة في رأيي هو بتشغيل كل جزء من الكود على حده إلى ان تصل الى السطر الذي لا يعمل او يسبب المشكلة و عندها نتعامل معه .

بعد كل هذا ارجوا ان تعتذر للكمبيوتر المسكين فهو مهما حدث لا يخطئ دائما اكوادك هي التي تعاني من العلل .

هكذا نكون قد وصلنا الى نهاية رحلتنا في هذا الجزء و الى اللقاء في اجزاء و كتب اخرى

ملحوظة : جميع الصور في هذا الكتاب من مصادر خارجية و حقوقها محفوظة لناشريها

كان معكم أخوكم

م / محمد يوسف محمد

للتواصل معي و في حالة وجود استفسارات او تعليقات

Modi401@hotmail.com

أو

https://www.facebook.com/mohamed.yossef.583

AGASHE

2010 - 2014