1

الفصل الأول : مقدمة

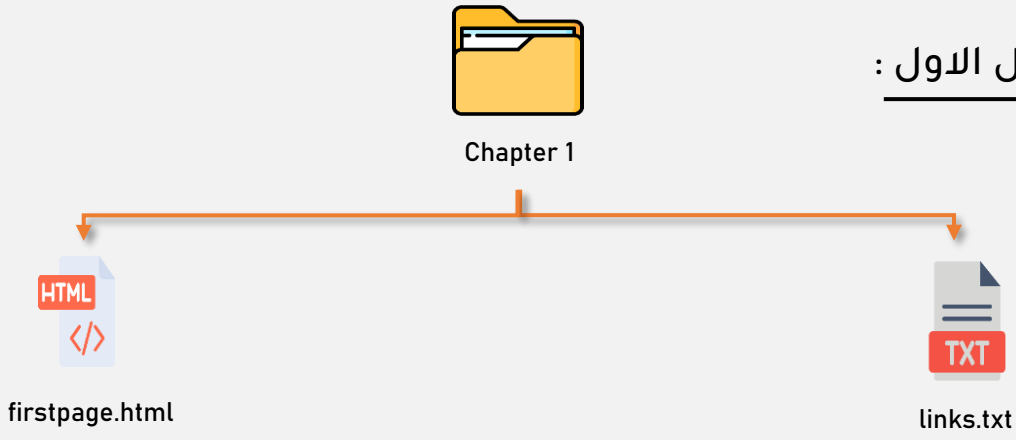
نبذة عن المحتوى :

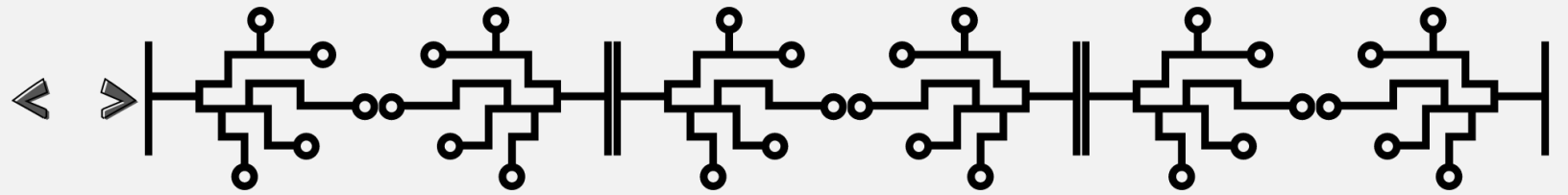
في هذا الفصل سنتعرف على خارطة الطريق التي ستساعدك خلال رحلة تعلمك لـ تصميم المواقع الالكترونية, وسنتعرف على بعض المفاهيم البرمجية اثناء عرضنا لهذه الخطة. بعدها سيتم التعرف على الادوات اللازمة لإنشاء صفحات الويب وتطبيق ماتم تعلمه من خلال انشاء صفحة ويب.

اهداف الفصل :

1. التعرف على خارطة الطريق.
2. التعرف على بعض المفاهيم البرمجية.
3. إعداد بيئة العمل والتعرف على VS CODE .
4. إنشاء اول صفحة ويب خاصة بك.

ملفات الفصل الاول :





لايخفى على احد التطور الذي حصل في عالم الانترنت في السنوات الاخيرة , وكيف أصبح الانترنت وسيلة لبناء وادارة المشاريع الخاصة, لذالك فإن تعّلم كيفية بناء صفحات الويب والمواقع الالكترونية اصبح وسيلة للدخول في هذا المجال. سوف نستعرض في هذا الفصل المخطط الذي يمكنك إتباعه لـ تصبح مطور ويب محترف وقادر على بناء مشاريع معقدة " هذا الكتاب لن يستعرض جميع التقنيات المذكورة في هذا المخطط ".

لكن وقبل أن نستعرض هذا المخطط دعونا نتعرف على بعض مصطلحات ومفاهيم عالم الانترنت :

الانترنت (Internet) : هو عبارة عن شبكة عالمية من أجهزة الكمبيوتر المتصلة ببعضها البعض والتي تتواصل من خلال مجموعة موحدة من البروتوكولات.

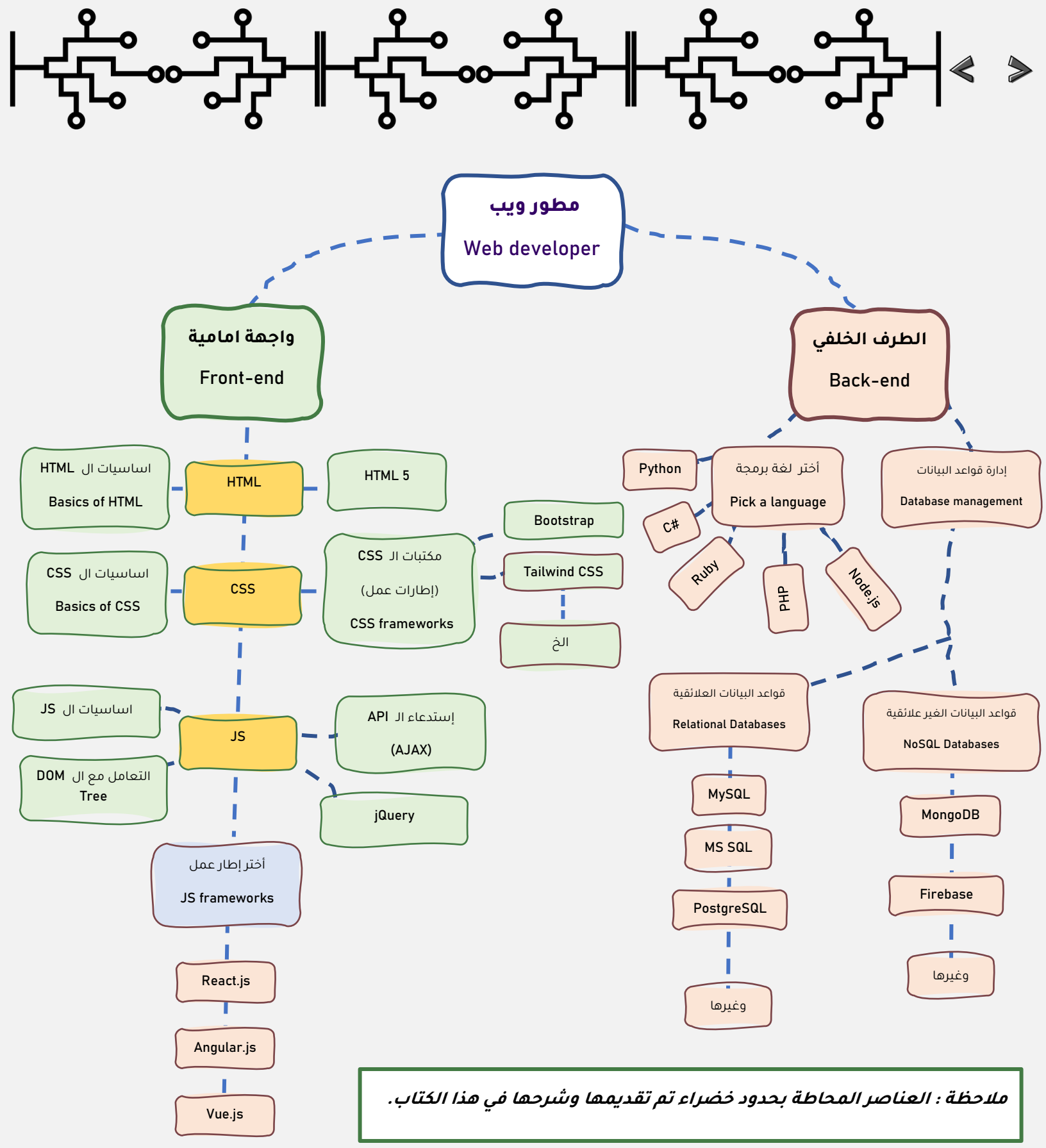
صفحة ويب (web page) : هي عبارة عن صفحات يتم تنسيقها بإستخدام لغة الـ HTML ويتم عرض محتواها عن طريق متصفحات الويب (web browsers) مثل : كروم , فايرفوكس , مايكروسوفت ايدج , برايف .... وغيرها.

متصفحات الويب (web browsers) : هي برامج تمكّن المستخدم من الوصول الي مواقع الويب على الانترنت , حيث تقوم هذه المتصفحات بترجمة الصفحات المكتوبة بـ لغة الـ HTML وعرض محتوياتها للمستخدم.

موقع الكتروني (website) : هو عبارة عن مجموعة من الصفحات يتم ربطها بطريقة معينة من خلال استخدام لغة الـ html.

خادم الويب (server) : هو جهاز كومبيوتر يتم فيه تخزين الملفات التي يمكن الوصول إليها عبر الإنترنت باستخدام HTTP.

بروتوكول نقل النص التشعبي (HTTP) : هو مجموعة قواعد تسمح بطلب مستندات الـ HTML ونقلها بين المتصفحات و خوادم الويب عبر الانترنت.



ملاحظة : العناصر المحاطة بحدود خضراء تم تقديمها وشرحها في هذا الكتاب.

وهناك العديد من التقنيات التي للاسف لم اكن قادراً على ذكرها في هذا المخطط ولكني انصح وبشدة ان تتعلمها مثل : Git , Package mangers (npm & yarn).

تطوير الواجهات الامامية (Front-end development) :

تطوير الواجهات الامامية والذي يعرف ايضاً بـ (**client-side development**) وتعني تطوير الجزء الذي يراه ويتفاعل معه المستخدم. يعمل مطورو الواجهات على كتابة الكود الخاص بالمظهر الخارجي لمواقع الويب او لتطبيقات الويب.

ماهي لغات البرمجة المستخدمة في تطوير الواجهات الامامية Front-end ؟

يستخدم مطورو الواجهة الأمامية ثلاث لغات برمجية أساسية :

1. HTML
2. CSS
3. JavaScript

1) لغة HTML :

(Hypertext Markup Language (HTML : هي لغة أو طريقة لوصف محتويات صفحة الويب وهي المكون الأساسي لصفحات الموقع فهي من يقوم بوصف العناصر (نصوص , صور , جداول , قوائم , نماذج) وغيرها العديد من العناصر الاخرى, لذلك ستكون نقطة الانطلاق ولن تاخذ وقتاً طويلاً لتعلمها.

2) لغة CSS :

تعرف باللغة العربية باسم صفحات الطرز المتراصة، وهي واحدةٌ من التقنيات الرئيسية المستخدمة لبناء صفحات الويب جنبًا إلى جنب مع لغة بناء صفحات الويب HTML. وتمتاز بإضفاء السمات والخصائص على البنية الخاصة بصفحات الويب كالألوان والخطوط والأنماط التي تتخذها صفحة الويب.

3) لغة JavaScript :

هي لغة برمجة عالية المستوى تستخدم أساسا في متصفحات الويب لإنشاء صفحات أكثر تفاعلية.

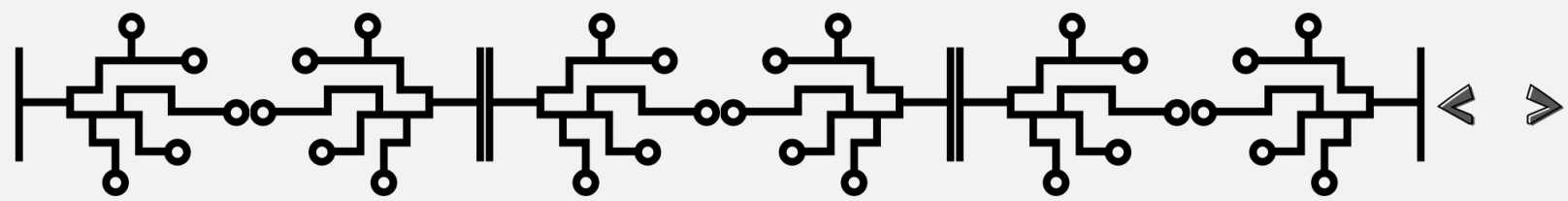
دعني اوضح العلاقة بين هذه اللغات باستخدام مثال بسيط :

1) HTML: البنية الاساسية للموقع (الاساس).
2) CSS: طريقة عرض الموقع (المظهر).
3) JavaScript: التفاعل والوظائف (الوعي).

HTML

CSS

JavaScript



من أين ابدأ ؟

كما يقال : " رحلة الألف ميل تبدأ بخطوة " وأول خطوات تعّلم تصميم المواقع تبدأ بتعّلم اساسيات الـ HTML , ولا اعتقد انك ستقضي وقتاً طويلاً في فهم الـ HTML وستصبح في وقتٍ قصير قادر على استخدام وسوم و عناصر ال HTML .

بعدها يجب ان تبدأ بتعلم لغة الـ CSS لإضافة بعض السمات والخصائص على هذه الوسوم وتجعل صفحات الويب اكثر جمالاً. وبعد ذالك دعنا نتعلم كيف نبعث الحياة على هذه الصفحات ونجعلها تفاعلية من خلال تعلم لغة ال JavaScript. بالتاكيد ستواجه العديد من المشاكل خلال فترة تعلمك لهذه اللغات الثلاثة , قد تجد حلولاً لبعض هذه المشاكل وقد لاتجد حلولاً ! لكن من المؤكد إنك ستتعلم الكثير من الاشياء في هذا المجال , وستصبح قادراً على قراءة وتحليل الاكواد المكتوبة بهذه اللغات ..... لذا اكمل رحلتك.

يمكنك اخذ استراحة قصيرة بعدها , لكن احرص على تطبيق ماتعلمته بشكل دوري ومستمر. ينصح خلال هذه الفترة أن تبحث عن بعض المشاريع الجاهزة والعمل على تحليلها وفهمها ومحاولة القيام بمشاريع مشابهه , يمكنك البحث عن بعض المشاريع الجاهزة في موقع الـ GitHub الذي اعتبره كنزاً حقيقياً لمن اراد تعلم البرمجة خصوصاً , ستجد فيه العديد من المشاريع المفتوحة المصدر والذي يمكنك رؤيتها والتعديل عليها.

بعدها توجه الي تعلم إطارات ومكتبات ال CSS والتي ستساعدك اكثر في تجميل صفحات الويب.

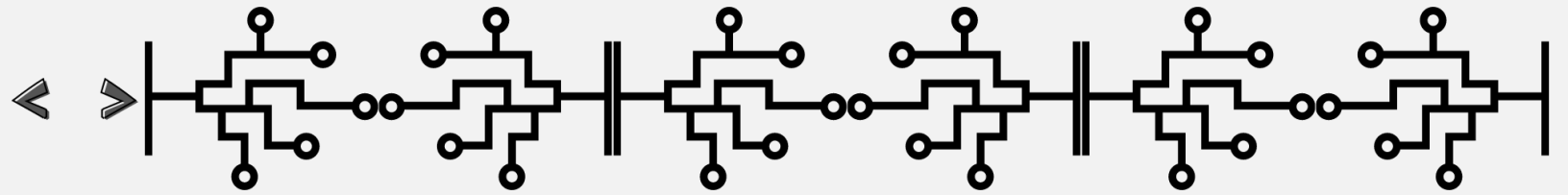
ماهي إطارات ومكتبات ال CSS التي يمكن استخدامها لتحسين عملية تصميم صفحات الويب ؟

هناك العديد من المكتبات المتاحة سوف اكتفي بذكر ثلاثٍ منها :

1. Bootstrap
2. Tailwind CSS
3. Foundation

بعدها ستتوجه الي تعلم مكتبات ال JavaScript والتي ستساعدك في جعل صفحات الويب التي تقوم بتطويرها أكثر حيوية وفاعلية وستمكنك من إضافة العديد من المزايا الي صفحات الويب.

وبعد هذه الرحلة الطويلة ستكون مطور ويب قادر على انشاء العديد من المشاريع الجميلة لكنها مجرد البداية , ستعمل بعدها على إختيار احدى إطارات تطوير الوجهات الامامية.



ماهي الإطارات وماهي الإطارات المستخدمة في تطوير الواجهات الامامية Front-end ؟

إطار (Framework) : قوالب عمل مجانية ومفتوحة المصدر يستخدمها المطورين عادةً لتسريع عملية إنشاء وبناء المواقع الالكترونية وتطبيقات الويب. أشهر هذه الاطارات :

1. React framework
2. Angular framework
3. Vue.js framework

بعد الانتهاء من إطارات الواجهات الامامية يمكنك البدء بالعمل كمطور واجهات امامية.

جهة الخادم (Back-end) :

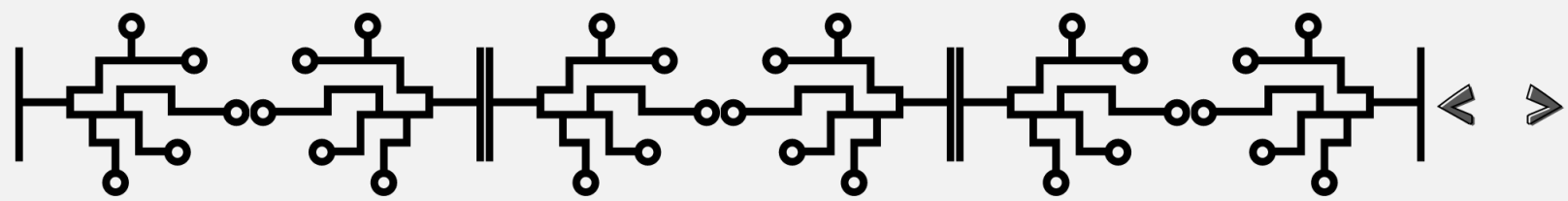
يمكنك بعدها البدء بـ **تطوير البنية التحتية أو جهة الخادم** (back-end) وهنا ستتعلم كيف تجعل الموقع الالكتروني يتفاعل مع المستخدم في الخلفية وكيف يتم التعامل مع **قواعد البيانات** والتحقق من البيانات المدخلة والكثير من العمليات التي لاتظهر للمستخدم. وبعد هذه العمليات كلها يرسل الخادم استجابة (Response) فيها معلومات عن نجاح أو فشل العملية.

بعض إطارات ال Back-end :

1. ExpressJS
2. ASP.NET
3. Django
4. Laravel

في هذه الاثناء من المهم ايضاً تعّلم قواعد البيانات وكيف يمكن استخدامها في تطوير المواقع الالكترونية.
بعدها قم بإنشاء مشاريع تستخدم فيها كل ماتعلمته خلال رحلة تعلمك , واحرص على انشاء ملف تعريفي خاص بك واعرض به جميع مشاريعك التي قمت بها.

من المؤكد انها ستكون رحلة طويلة وشاقة لكنها ايضاً تستحق , اتمنى ان تستمتع برحلتك ...



ماهي متطلبات هذا الكورس وما الذي تحتاجه اثناء قرائتك لهذا الكتاب ؟

1. سابدأ هذه القائمة بالشغف والرغبة على ان تصبح مطور مواقع , لانها ستكون دافعك للاستمرار حتى النهاية.
2. جهاز حاسوب ومعرفة اساسيات استخدام الحاسوب (انشاء المجلدات والملفات وتصفح الانترنت).

في اخر فقرات هذا الفصل دعنا نعمل على إعداد بيئة العمل التي ستساعدنا كثيراً في رحلتنا هذه :

متصفح الويب (**Web browser**): توجد العديد من متصفحات الويب التي يمكنك استخدامها لكني افضل شخصياً **Microsoft edge** , وذلك لما يحتويه من ادوات ستساعدنا اثناء تطوير صفحات الويب.
اما إذا كنت مستخدماً لنظام ال **mac** او ال **Linux** ف **google chrome** بديل مناسب.

محرر الكود (**code editor**) : برنامج تحرير التعليمات البرمجية يحتوي على العديد من الميزات التي تجعل كتابة وتحليل الاكواد اكثر سهولة وسرعه.

يوجد العديد من برامج تحرير الاكواد واختيار البرنامج المناسب لك يعتمد على تفضيلاتك الشخصية.

بعض برامج تحرير النصوص التي يمكنك إستخدامها :

1. Visual Studio Code (vs code)
2. Sublime Text
3. ATOM
4. Brackets
5. ++Notepad

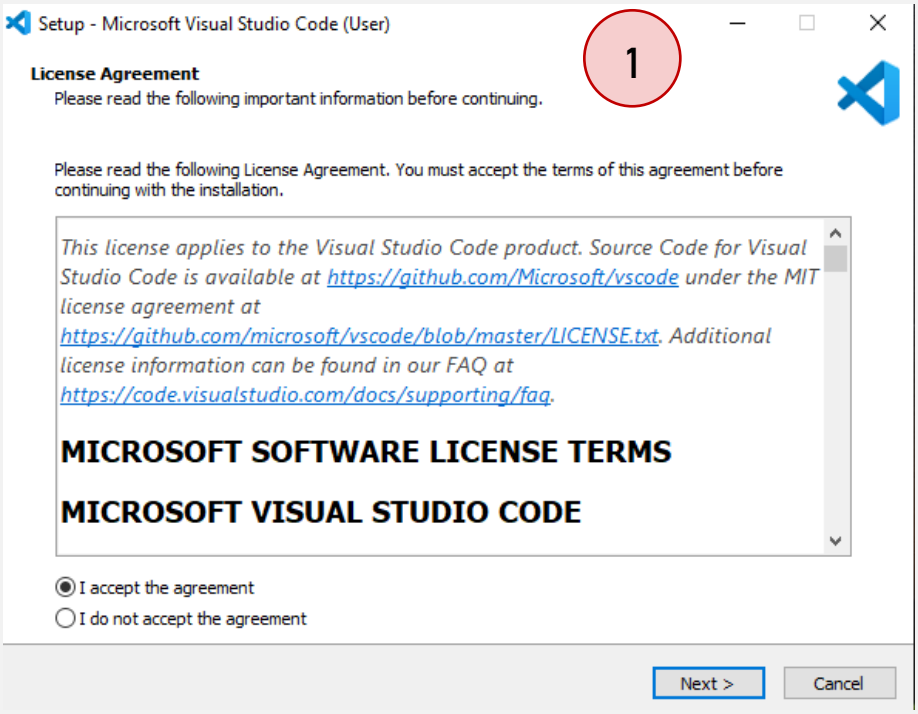
لكني شخصياً افضل استخدام الـ **visual studio code** , وذلك لانه يحتوي على ادوات وإضافات جعلتني افضله عن غيره.

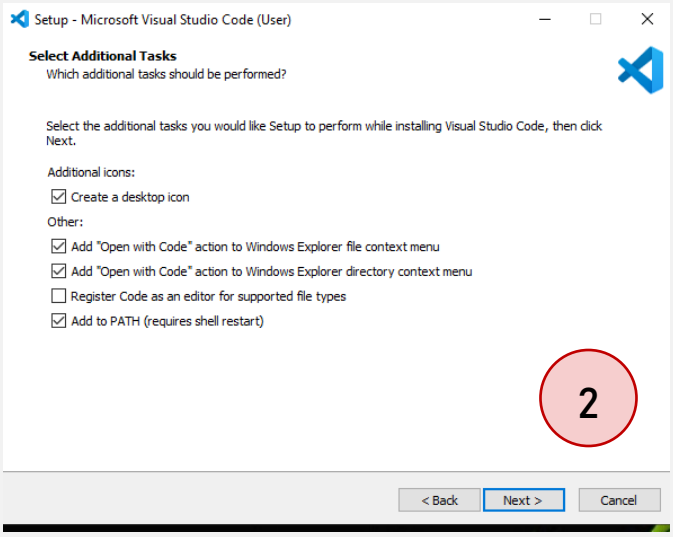
خطوات تنزيل وتثبيت VS CODE :

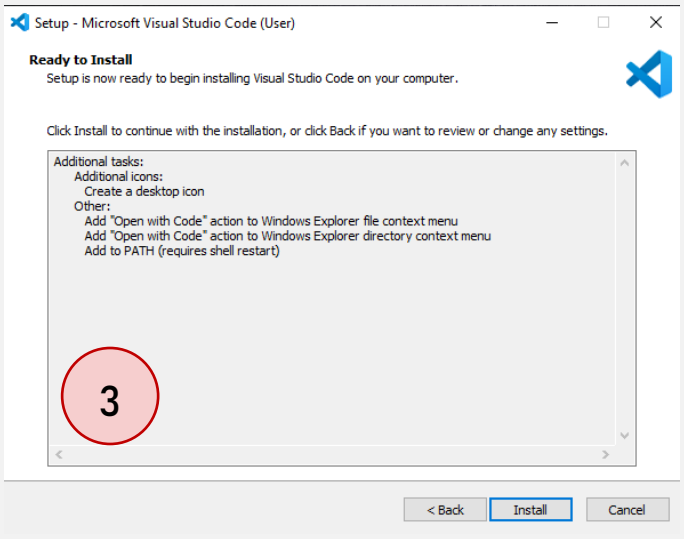
1. اذهب الي الموقع الرسمي و أختر النسخة التي تناسب نظام تشغيل جهازك :

   https://code.visualstudio.com/download

2. إبدء بعملية تنصيب البرنامج كما هو موضح في الصور:



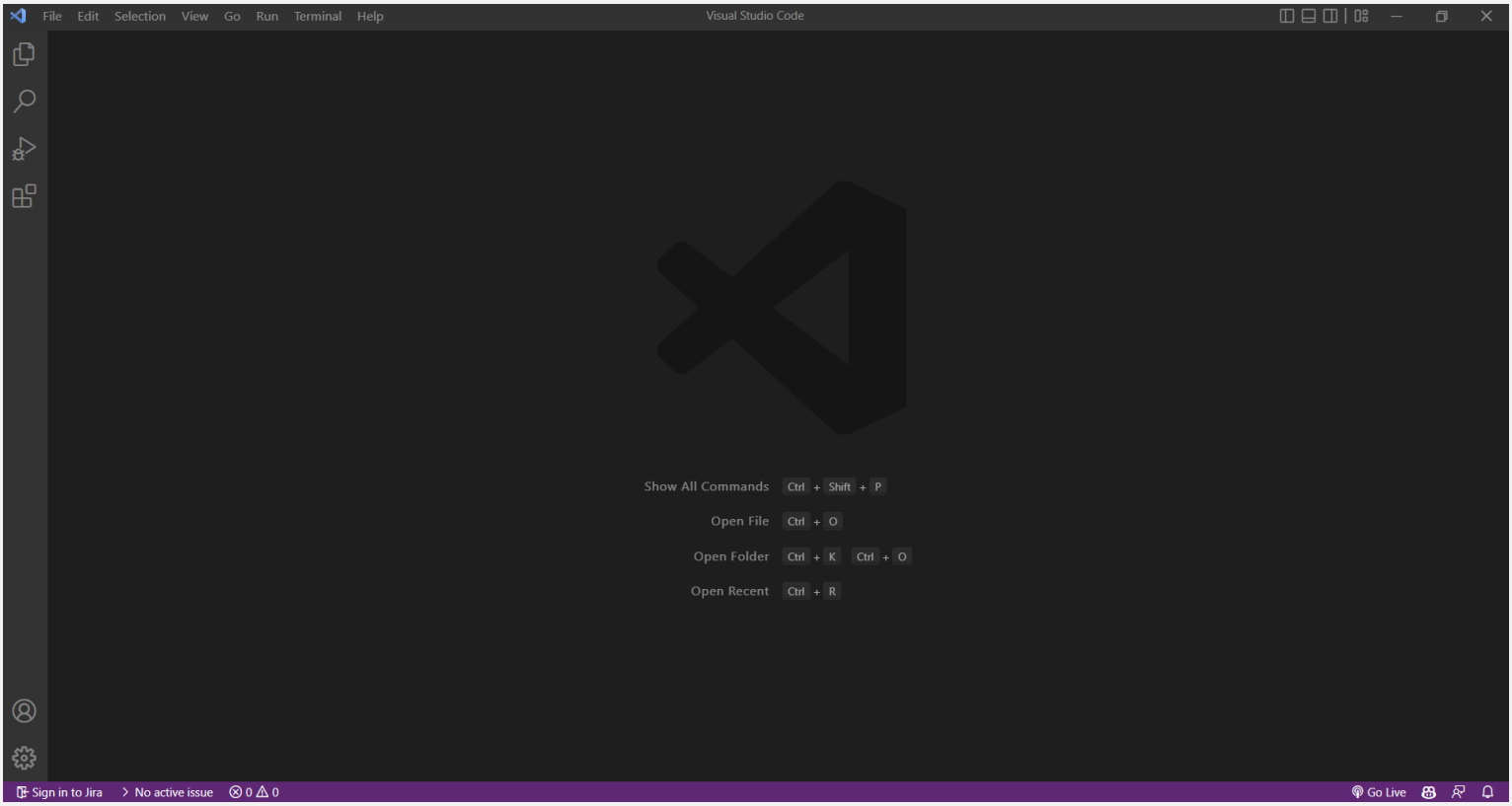




بعد الانتهاء من عملية التثبيت دعنا نتعرف على واجهة البرنامج :

- في البداية ستختار النمط الذي تراه مناسباً لك.
- بعدها ستظهر لك **welcome screen** , يمكنك القيام بجولة لتتعرف على اقسام البرنامج.

الواجهة الرئيسية :



1. **Activity bar**: شريط المهام والذي يتيح لك التبديل بين المهام.

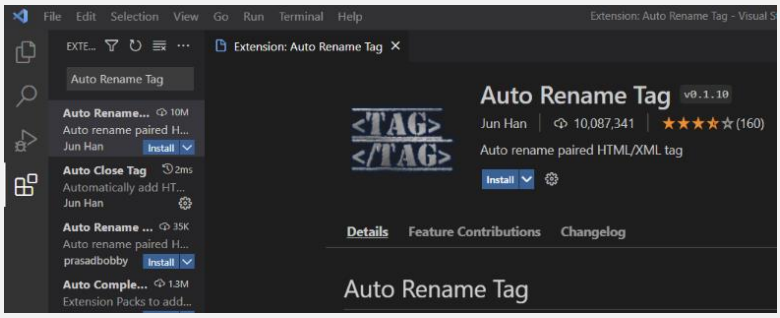
a. **Explorer**: يستخدم المستكشف لاستعراض كافة الملفات والمجلدات الموجودة في مشروعك وفتحها وإدارتها.

b. **Search.**

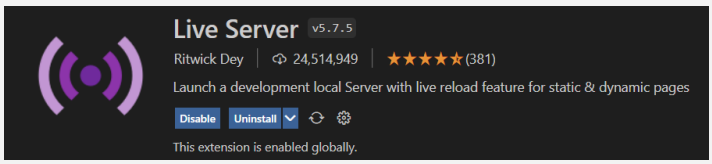
c. **Run & debug**.

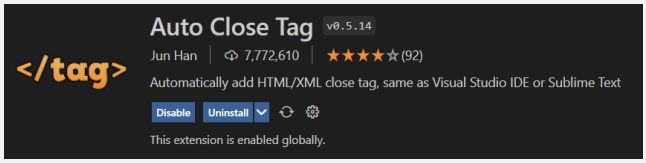
d. **Extensions**: ومن هنا يمكنك تحميل وإدارة الاضافات.

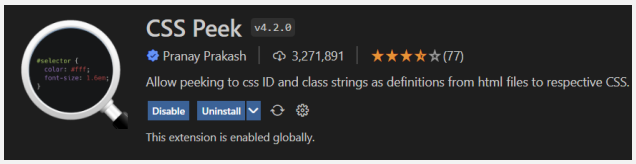
دعنا نبدأ بتحميل بعض الإضافات التي ستساعدك اثناء تعلم برمجة المواقع , وهنا اهم هذه الإضافات :

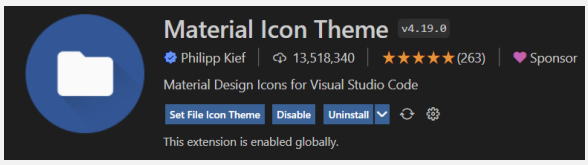


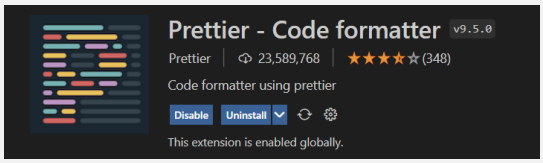
1. Auto Rename Tag
2. Auto Close Tag
3. Live Server
4. Material Icon Theme
5. Prettier
6. CSS Peek



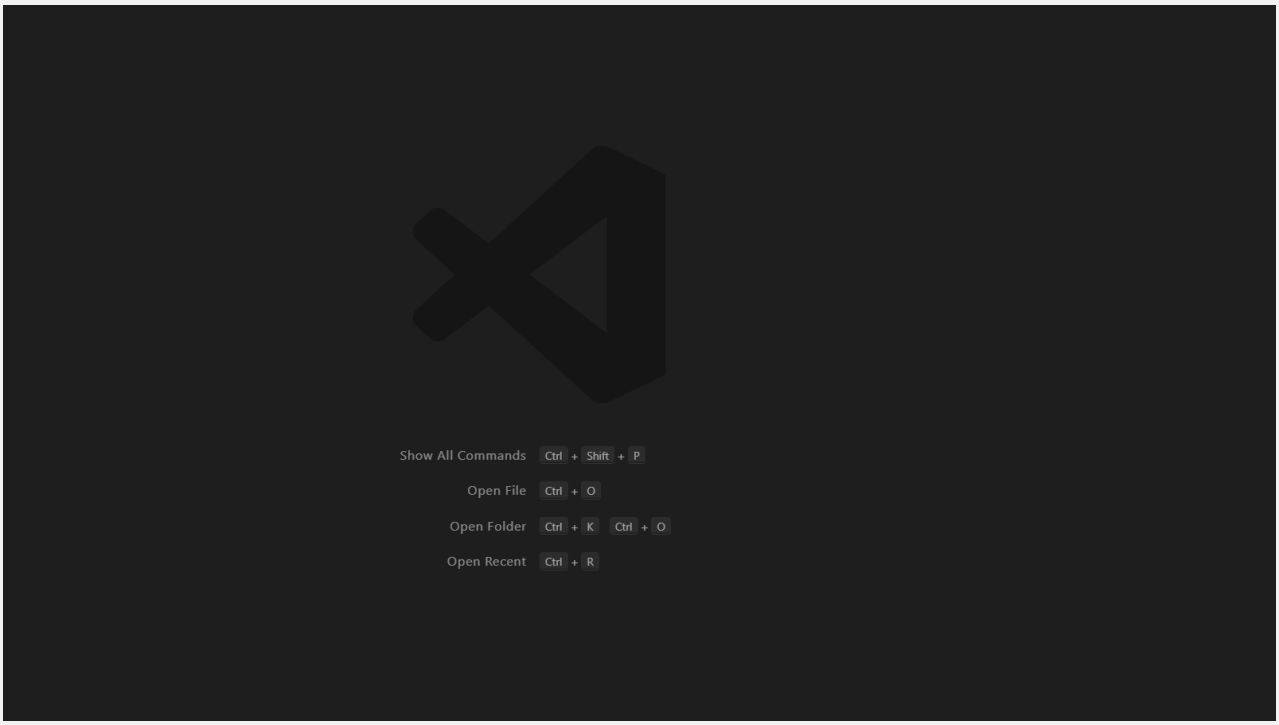








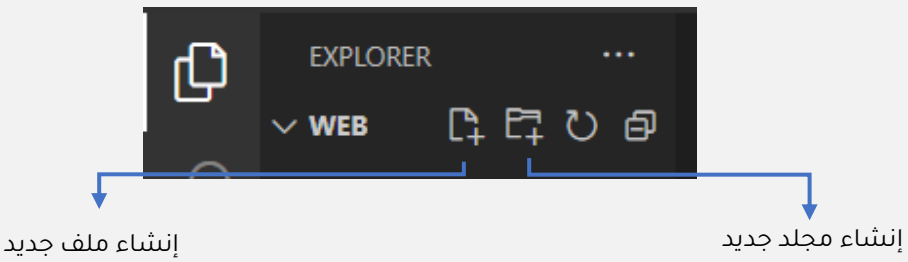
2. **Editor**: منطقة كتابة وتحرير الكود. ستعرض محتويات الملفات في هذا المساحة حيث يمكنك فتح العديد من الملفات.

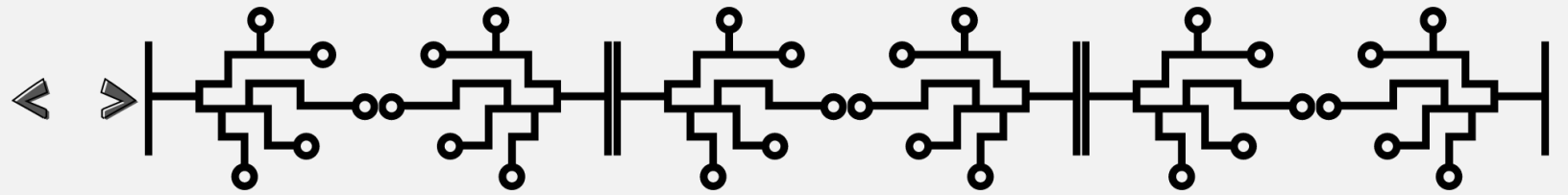


ماهي خطوات إنشاء وفتح مجلد جديد بإستخدام الـ VS CODE ؟

1. انقر على كلمة File الموجودة في الاعلى.
2. Open Folder ...
3. أختر المجلد الذي تريد وضع مشروعك فيه (يمكنك انشاء مجلد جديد بالنقر على كلمة New folder).
4. بعد تحديد المجلد انقر على Select Folder .

ستظهر بعدها جميع الملفات الموجودة داخل المجلد في المستكشف وستظهر ايضاً هذه الايقونات , يمكنك استخدام هذه الايقونات لـ إنشاء ملفات ومجلدات جديدة داخل المشروع.





الان اصبحت جاهزاً للبدء في تطوير صفحات الويب. لذلك دعنا الان نقوم بعرض صفحتك الاولى في المتصفح !

1. قم بإنشاء ملف جديد وتسميته (firstpage.html) يمكنك اختيار اسم الملف كما تحب لكن إحرص على ان يكون امتداد الملف (**.html**).
2. انقر على الملف الذي قمت بـ إنشاءه و أضف هذه الاسطر ←
3. احفظ الملف.
4. انقر الماوس من جهة اليمين وابحث عن خيار (Open with live server) وانقر عليه.
5. بهذه الطريقة تكون قد انشأت اول صفحة ويب.

```
<!DOCTYPE html>
<html>
<head>
    <title>Document</title>
</head>
<body>
    <h1> Hello World! This is my first page</h1>
</body>
</html>
```

لن يتم شرح هذا الكود في هذا الفصل , سيتم شرح هذه الاسطر بالتفصيل في الفصل الثاني.

ملخص الفصل الأول:

1) ينقسم تصميم المواقع الرئيسية الي قسميين رئيسيين :
   a) تصميم الواجهات الامامية Front-end development .
   b) برمجة جهة الخادم Back-end development .
2) محرر الكود Code Editor : هو برنامج يحتوي على العديد من الادوات والاضافات التي تساعد المبرمج في كتابة وتحليل الاكواد البرمجية.
3) يجب ان تنتهي ملفات ال HTML بالامتداد (**.html**).
4) يمكنك رؤية محتوى صفحة الويب عن طريق عرضها في متصفح الويب , ويمكن استخدام اداءة الـ Live server لجعل عرض المحتوى بشكل مباشر بعد حفظ التغييرات دون الحاجة الي إعادة فتح الملف من جديد.

2

الفصل الثاني : أساسيات الـ HTML

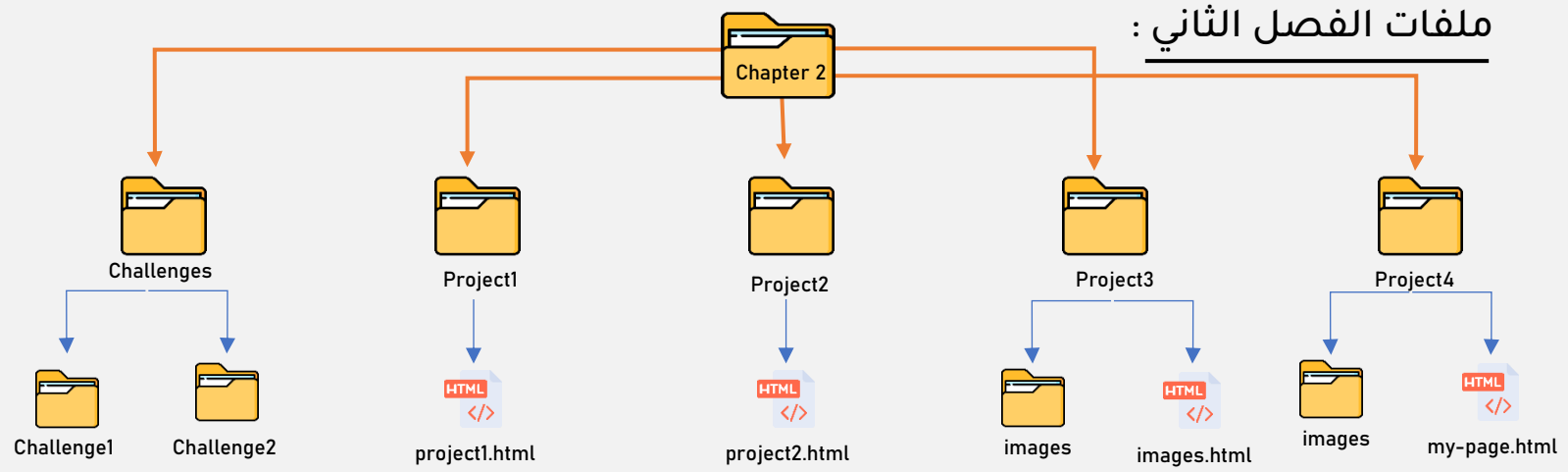
نبذة عن المحتوى :

في هذا الفصل سنتعرف أكثر على لغة الـ HTML وكيف تتم عملية كتابة و عرض صفحات الويب في متصفح الويب. سننظر ايضاً في كيفية كتابة عناصر الـ HTML وماهي مكونات هذه العناصر. بعدها سنبدأ بكتابة صفحات ويب تحتوي على بعض النصوص والصور والقوائم. ونختتم هذه الفصل بتصميم صفحة شخصية.

اهداف الفصل :

1. التعرف على البنية الاساسية لملفات الـ HTML .
2. التعرف على بعض الوسوم وخصائصها.
3. إضافة عناصر النصوص الي صفحة الويب.
4. إضافة صور الي صفحة الويب.
5. إضافة قوائم الي صفحة الويب.
6. تطبيق ماتم تعلمه : بناء صفحة شخصية.

ملفات الفصل الثاني :



مشروع نهاية الفصل الثاني :

Name : Panda Manga

I am a web developer

I am a web developer and a web designer. I am currently working as a ***freelancer*** web developer.

Work experinces :

i. 6 years as a web developer
ii. 4 years as a mobile application Designer
iii. 3 year as a teacher

Languages :

- English
- Arabic
- Turkish
- Spanish

Skills :

1. Web Design
2. Web Development
3. Mobile Application Design
4. Mobile Application Development
5. Teaching

interests :

- Reading
- Watching Movies
- Watching TV Shows
- Watching Sports
- Watching Anime

بعد الانتهاء من التعرف على مكونات الويب و على أقسام تطوير المواقع الالكترونية (الفصل الأول) والانتهاء ايضاً من إعداد الحاسوب فانت الان اصبحت جاهزاً لتعلم لغة الـ HTML , لذا دعنا نبدأ بسؤال : **ماهي الـ HTML ؟**

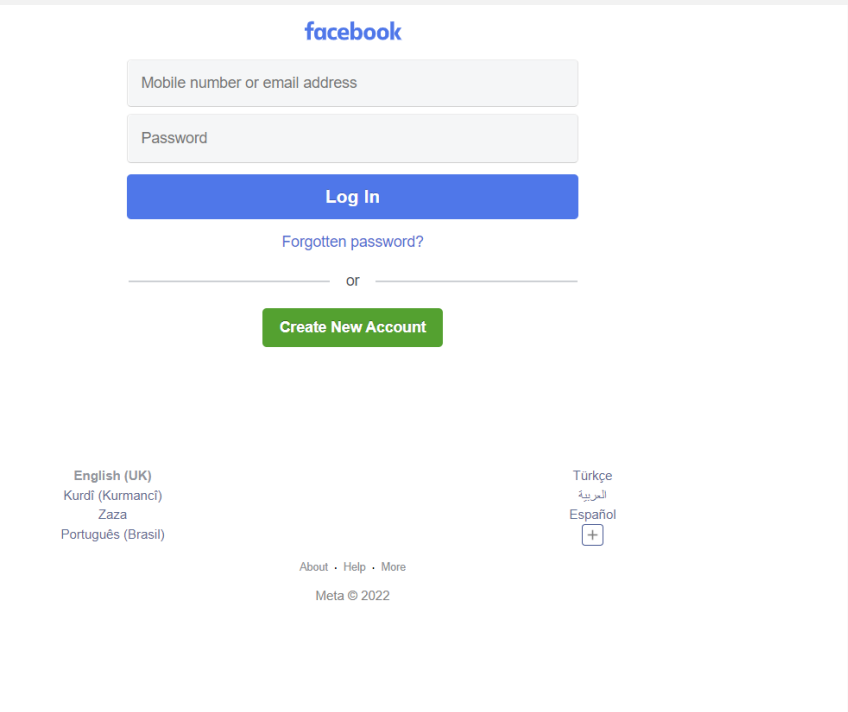
HyperText Markup Language (HTML) : هي لغة ترميز تستخدم لوصف مكونات صفحة ويب. (***لغة وصف وليست لغة برمجية***).

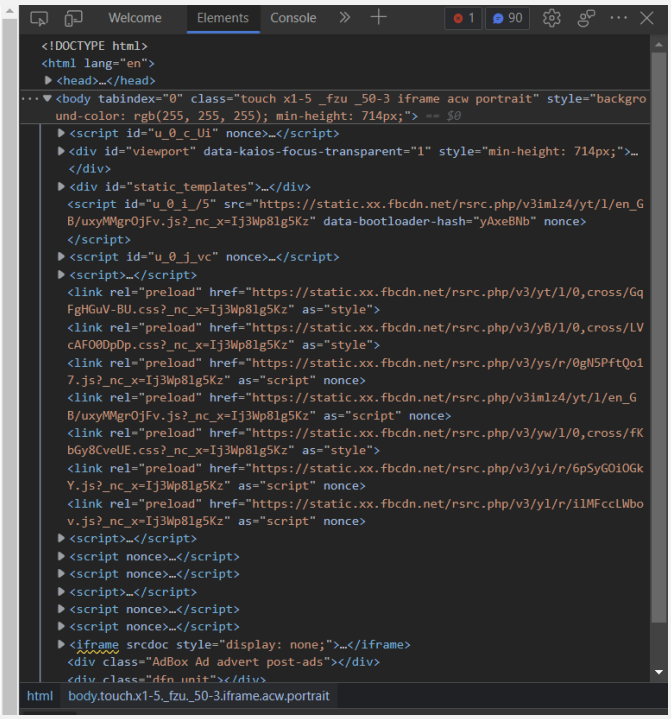
تعتبر الـ **HTML** البنية الاساسية (الهيكل) لكل صفحات الويب التي تراها في متصفحات الويب , حيث يمكنك ان تتأكد من هذا الامر من خلال زيارة اي صفحة ويب او موقع الكتروني والاطلاع على مكوناته.

نعم يمكنك الاطلاع على مكونات اي موقع الكتروني بكل بساطة :

1. إذهب الي الموقع الذي تريد الاطلاع على مكوناته.
2. انقر الماوس من جهة اليمين وابحث عن خيار الـ **inspect** (فحص) ثم انقر عليه.
3. يمكنك الاطلاع على مكونات الصفحة من خلال النافذة التي ستظهر.

على سبيل المثال : دعنا نجرب هذه العملية على موقع فيسبوك (صفحة تسجيل الدخول).





يمكنك رؤية مكونات الصفحة من خلال النافذة الجديدة الموجودة في جهة اليمين.

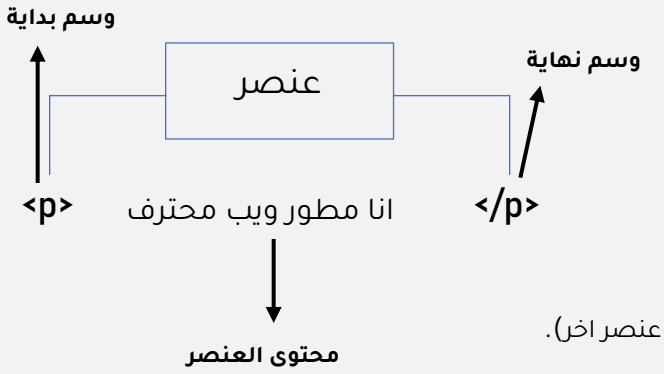
(*لاتقلق , إنشاء المواقع ليس معقداً كما قد يتضح من الصورة , الامر اسهل بكثير!*)

تتكون صفحة الويب من مجموعة من العناصر (**Elements**) او ماتسمى بالوسوم (**Tags**) تكتب هذه الوسوم بين علامتي أكبر و أصغر <> , ولا يقوم المتصفح بعرض هذه الوسوم، ولكن يقوم بعرض محتوياتها.

ما هي مكونات هذه الوسوم (Tags) ؟

- وسم بداية العنصر : ** **
- وسم نهاية العنصر : ** **
- محتوى العنصر : ويكون المحتوى بين وسم البداية و وسم النهاية.

```
<p>  انا مطور ويب محترف  </p>
```



قد تحتوي هذه العناصر على بعض الخصائص والتي سيتم وضعها داخل وسم البداية كما سنلاحظ لاحقاً.

قد تحتوي ايضاً على عناصر بداخلها(محتوى العنصر يمكن ان يكون عنصر اخر).

أغلب عناصر الـ **HTML** تحتوي وسم البداية و وسم النهاية , لكن توجد بعض العناصر التي تتكون من وسم واحد فقط ويتم عرض محتوى هذه العناصر مباشرةً , فمثلاً طريقة عرض الصور في صفحة الويب :

```
<img src="img.png" alt="My image">
```

كما سنلاحظ في عنصر الصورة فهو يتكون من وسم واحد فقط يحتوي على بعض الخصائص (**src , alt**) سنتعرف عليها بالتفصيل لاحقاً .

دعنا نتعرف أكثر على هيكلة ملف الـ HTML : سبق و أن رأينا هذه

```
<!DOCTYPE html>
<html>
<head>
   <title>Salah Alkmali</title>
</head>
<body>
   <h1> Hello World! This is my first page</h1>
</body>
</html>
```

الاسطر في الفصل الأول , لكننا لم نتعرف على هذه العناصر .

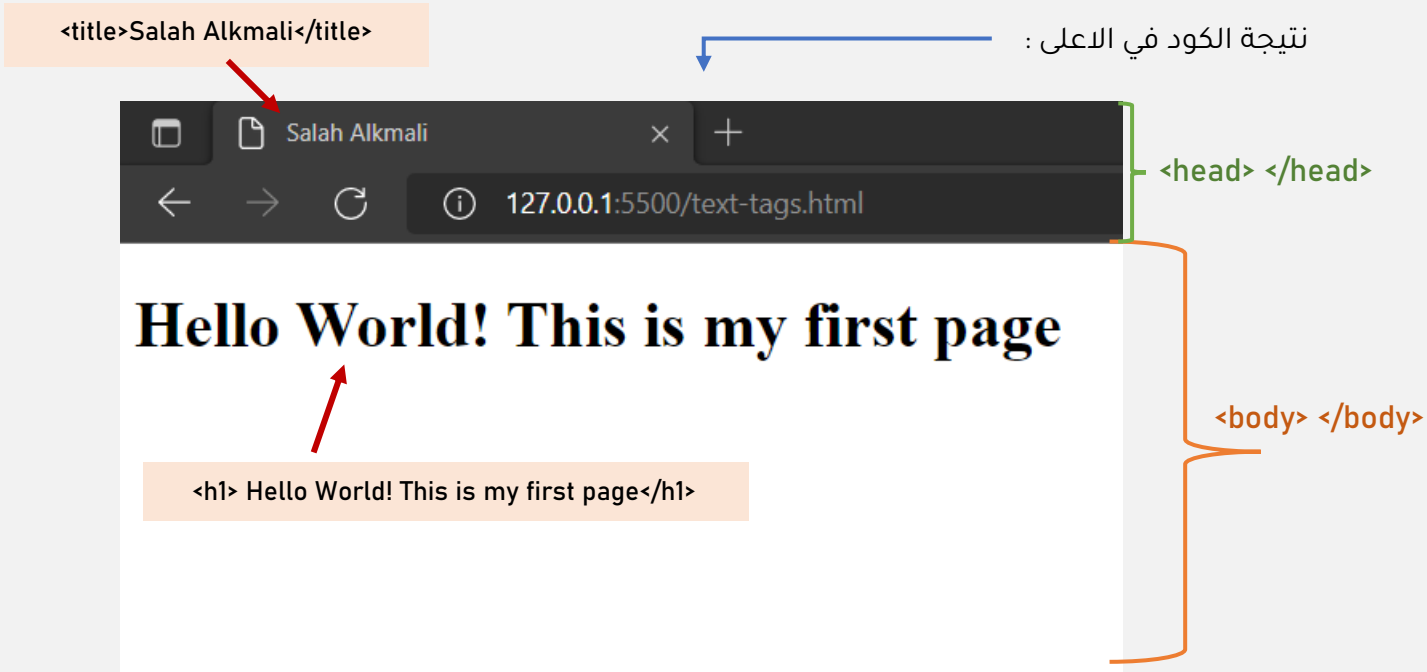
1. **<!DOCTYPE html>** : إعلان نوع الصفحة.

يوجد هذا الاعلان في اول سطر من جميع ملفات الـ **HTML** وهو يستخدم ل إعلام المتصفح ان هذا الملف هو عبارة عن صفحة ويب.

2. **<html> </html>** : يعتبر هذا العنصر هو اساس صفحة الويب , يقسم هذا العنصر صفحة الويب الي قسمين رئيسيين :

a) **<head> </head>** : يحتوي هذا العنصر على معلومات عن صفحة الويب , لن تظهر محتويات هذا العنصر في صفحة الويب (باستثناء عنوان صفحة الويب).

b) **<body> … </body>** : والذي يحتوي على جميع العناصر التي ستظهر في صفحة الويب.



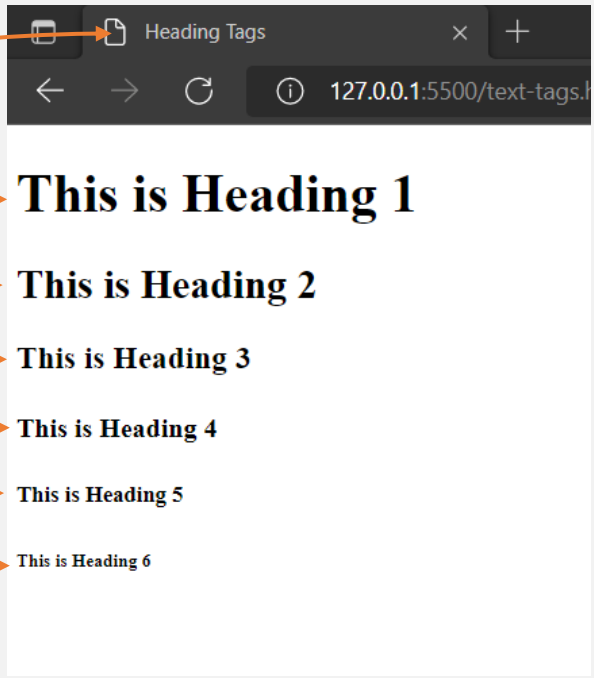
جميع النصوص والصور والنماذج وغيرها من العناصر التي تراها في صفحات الويب موجودة في داخل ال body .

عناصر النصوص (Text tags) : عناصر تستخدم ل اضافة نصوص الي صفحة الويب , وهناك العديد من هذه العناصر.

- **Heading tags** : تستخدم هذه العناصر لكتابة العناوين الرئيسية وتنقسم الي **6** عناصر تختلف جميعها بحجم الخط : <h1> </h1> , <h2> </h2> ... <h6> </h6> , حيث يعتبر **h1** اكبرها خطاً و **h6** اصغرها خطاً.

```
<!DOCTYPE html>
<html>
<head>
  <title> Heading Tags </title>
</head>
<body>
  <h1> This is Heading 1 </h1>
  <h2> This is Heading 2 </h2>
  <h3> This is Heading 3 </h3>
  <h4> This is Heading 4 </h4>
  <h5> This is Heading 5 </h5>
  <h6> This is Heading 6 </h6>
</body>
</html>
```

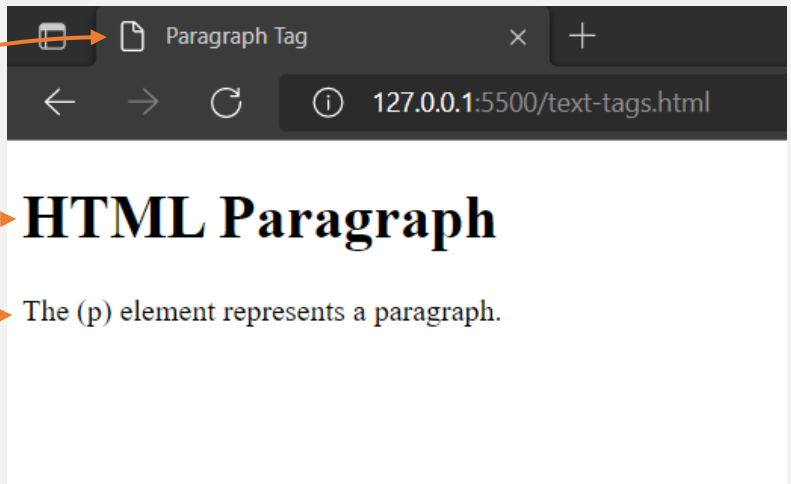
الكود

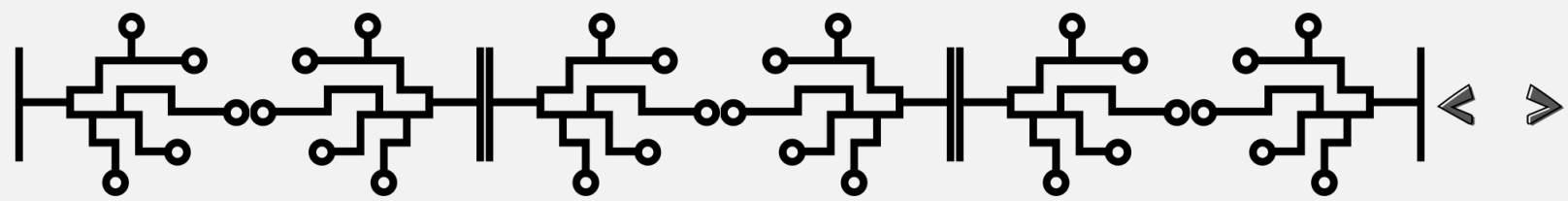


النتيجة

- **Paragraph Tag ( )** : فقرة نصية , ويستخدم هذا العنصر لأنشاء نصوص.

```
<!DOCTYPE html>
<html>
<head>
  <title> Paragraph Tag </title>
</head>
<body>
  <h1> HTML Paragraph </h1>
<p> The (p) element represents a paragraph. </p>
</body>
</html>
```





يتميز عنصر الـ **<p>** بانه يقوم بتجاهل كل الفراغات والاسطر التي تم كتابتها في ملف الـ **HTML** لذلك فإنك سترى الصفحة بدون الفراغات والاسطر الجديدة إن كنت اضفتها الي محتوى الـ **<p>**.

- **<pre> </pre>** : يمكنك إستخدام هذا العنصر لعرض النص كما هو مكتوب في ملف ال **HTML** , حيث يحافظ هذا العنصر على كلٍ من المسافات والأسطر الجديدة.

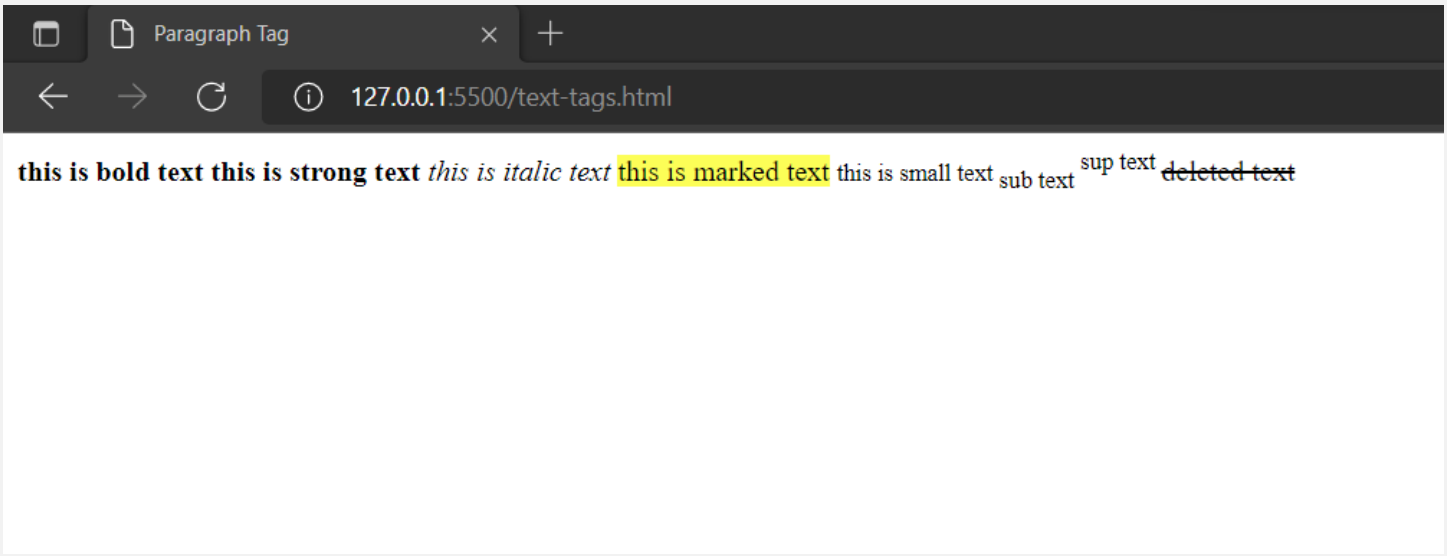
عناصر الـ **heading** و عنصر الـ **p** وايضاً عنصر **pre** , جميها تعتبر **block-level** وهذا يعني انها تبدأ دائماً بسطر جديد وتاخذ عرض الشاشة المتاح كاملاً (تلقائياً) , توجد بعض العناصر التي لا تبدأ في سطر جديد، فقط تأخذ المساحة التي تحتاجها فقط بدون التمدد لباقي المساحة ويطلق عليها عناصر **inline-level**.

الجدول التالي يحتوي على بعض عناصر النصوص التي تعتبر **inline-level** :

العنصر	شرح
** **	نص عريض : **bold text**
** **	نص عريض : **bold text**
** **	نص مائل : **italic text**
<mark> </mark>	نص مميز بخلفية ملونة : **Highlighted text**
** **	نص صغير
****	نص صغير اسفل خط الكتابة
****	نص صغير اعلى خط الكتابة
<del> </del>	نص محذوف

```
<!DOCTYPE html>
<html>
<head>
    <title> Paragraph Tag </title>
</head>
<body>
        <b> this is bold text</b>
        <strong> this is strong text</strong>
        <i> this is italic text</i>
        <mark> this is marked text</mark>
        <small> this is small text </small>
        <sub> sub text </sub>
        <sup> sup text </sup>
        <del> deleted text</del>
</body>
</html>
```

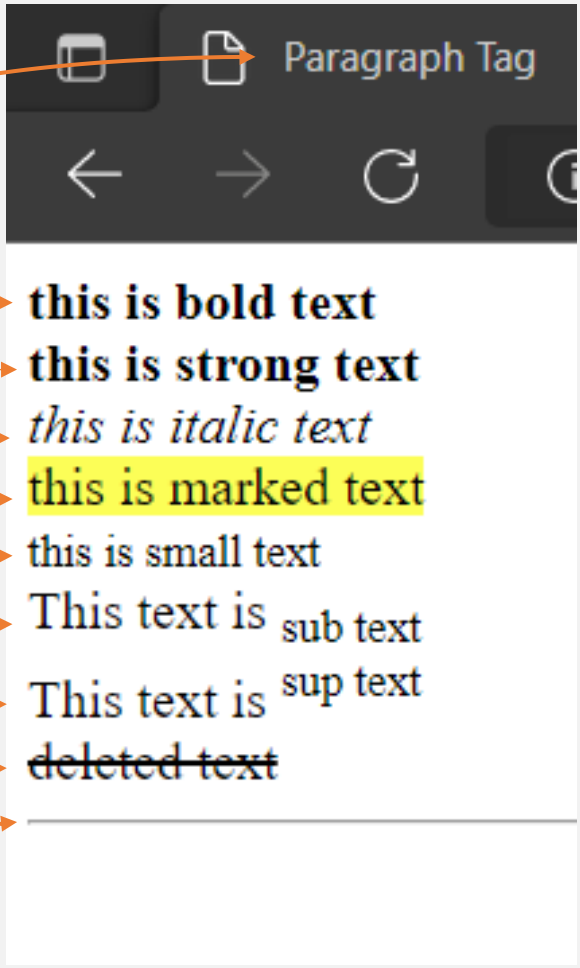
سوف تلاحظ في نتيجة هذا الكود إن جميع النصوص ستكتب في سطر واحد , وذلك لإن جميع هذه العناصر تعتبر **inline-level** اي انها تحجز مكان مايكفي لعرض محتواها.



دعنا نتعرف على عنصر ال **
** وهو عنصر يستخدم لـ انشاء سطر فارغ حيث يمكننا استخدامه للفصل بين العناصر المذكورة في المثال السابق.

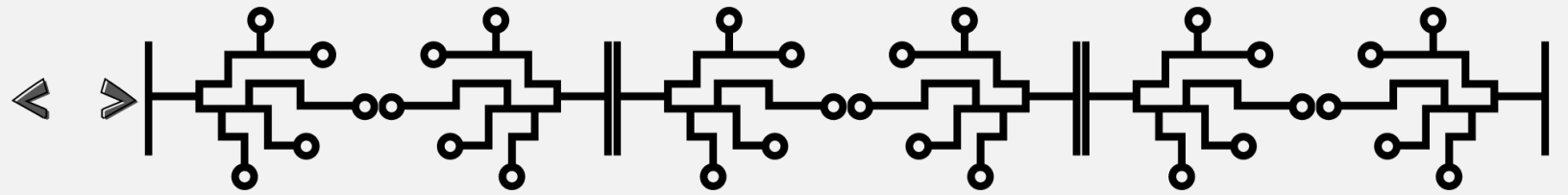
يوجد ايضاً عنصر **<hr>** ويستخدم هذا العنصر لـ إنشاء خط افقي (مرئي).

```
<!DOCTYPE html>
<html>
<head>
    <title> Paragraph Tag </title>
</head>
<body>
        <b> this is bold text</b>
        <br>
        <strong> this is strong text</strong>
        <br>
        <i> this is italic text</i>
        <br>
        <mark> this is marked text</mark>
        <br>
        <small> this is small text </small>
        <br>
        <sub> sub text </sub>
        <br>
        <sup> sup text </sup>
        <br>
        <del> deleted text</del>
        <hr>
</body>
</html>
```



عناصر الـ **
** و **<hr>** هي من العناصر التي تتكون من وسم واحد فقط (ليس لها وسم بداية ونهاية) وذلك لانها لايمكن ان تحتوي على عناصر اخرى.

دعنا نطبق ماتعلمناه عن عناصر النصوص بـ إنشاء صفحة تحتوي على مقالة بسيطة عن الـ HTML.



مشروع 2.1 : project1.html

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Web Development</title>
</head>

<body>

    <h1>HTML</h1>

    <p>
        <strong> HyperText Markup Language (HTML)</strong>
        is a markup language used to <br> describe the
        structure of a web page.
    </p>

    <hr>

    <h2>Tools Needed to Learn HTML</h2>

    <h4>1. Web Browser</h4>

    <p>Web browser is a program that allows you to view a website.</p>

    <h4><mark>2. HTML Editor</mark></h4>

    <p>HTML editor is a program that allows you to edit the HTML code.</p>

</body>

</html>
```

خاصية تسمح لنا بتحديد لغة صفحة الويب : en → English

عنوان الصفحة

عنصر نصي يحتوي بداخله على نص
وبعض العناصر الاخرى :

br للقفز الي سطر جديد ,

strong لجعل محتواها بخط عريض

مشروع 2.2 (مقالة باللغة العربية)

project2.html

```html
<!DOCTYPE html>
<html lang="ar" dir="rtl">

<head>
    <title> مطور ويب </title>
</head>

<body>
    <h1> ماهي لغة الـ HTML ؟</h1>

    <p> HTML ، من أمتع ما يمكن أن تتعلمه في عالم الويب، هي لغة
        نعم إنها بسيطة جداً وشيقة، يمكنك تطبيق ما ستعرفه وترى
        نتيجته في الحال.
        <br>
        فقط HTML لا تحتاج الكثير من الأدوات كي تبدأ في تعلم وتطبيق
        <strong>محرر نصوص</strong> <strong>ومتصفح
            ويب</strong>، في البداية ربما تجد بعض
        الغرابة
        لكن سرعان ما تألفها وتتعود عليها HTML في التعامل أكواد.
    </p>

    <hr>

    <h2> الأدوات المطلوبة لتعلم HTML </h2>

    <h4>1. متصفح ويب</h4>

    <p>متصفح ويب هو أداة تستخدم لعرض الموقع على الشاشة والتحكم به</p>

    <h4><mark>2. محرر نصوص</mark></h4>

    <p> محرر النصوص هو أداة <i> تستخدم لتحرير المحتوى </i> المرتبط بالموقع</p>

</body>

</html>
```

لاحظ ان عنصر ال HTML في هذا المشروع يحتوي على خاصية ال **lang** وهي خاصية لتحديد لغة صفحة الويب , حيث اننا جعلنا لغة الصفحة هي العربية "**ar**" , بعد تحديد اللغة , اضفنا خاصية جديدة وهي **dir** وهذه الخاصية هي من تحدد جهة الكتابة , حيث جعلنا جهة الكتابة في هذا المشروع من اليمين الي اليسار (**rtl**) اختصاراً لـ :

right to left.

عنوان الصفحة : مطور ويب.

بدأنا الصفحة بعنوان رئيسي **<h1>**.

بعدها اضفنا عنصر **<p>** واضفنا بداخله بعض العناصر : سطر جديد , خط عريض.

وبعدها اضفنا بعض العناصر , بعض هذه العناصر كانت تحتوي على عناصر اخرى كما ترى في صورة النتيجة ←

عنصر الصور img : يمكننا إضافة الصور الي صفحة الويب بإستخدام هذا العنصر , وهو من العناصر ذات الوسم الواحد فقط حيث يمكننا تحديد الصورة المراد عرضها بـ إستخدام خاصية **src** .

```
<body>
    <img src="https://see.news/wp-content/uploads/2020/12/UK_wildbirds-01-robin.jpg" alt="Bird">
</body>
```

كما نرى هنا , فإن عنصر الصور يتكون من وسم واحد فقط ويحتوي على بعض الخصائص :

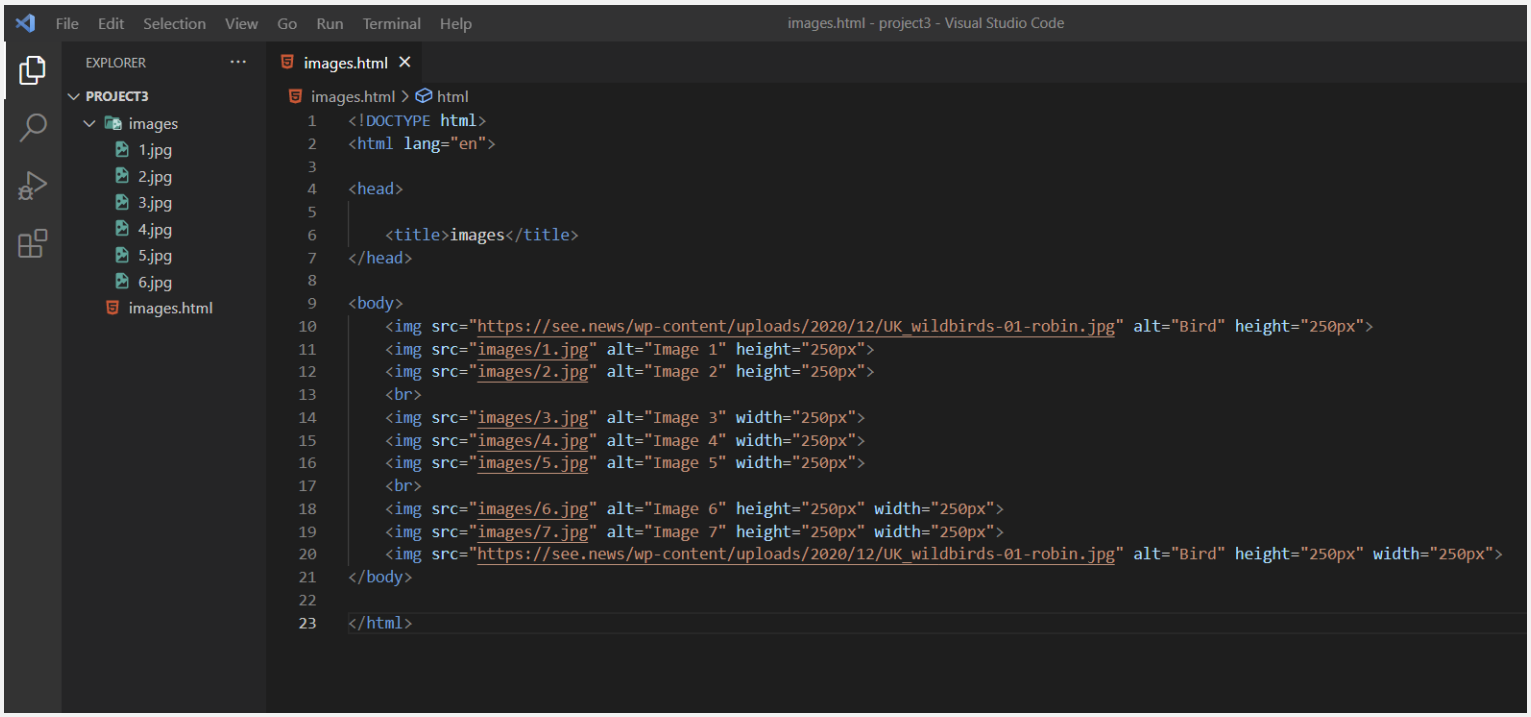
src : رابط **URL** للصورة، وهذه الخاصية إجبارية لجميع عناصر <**img**>.

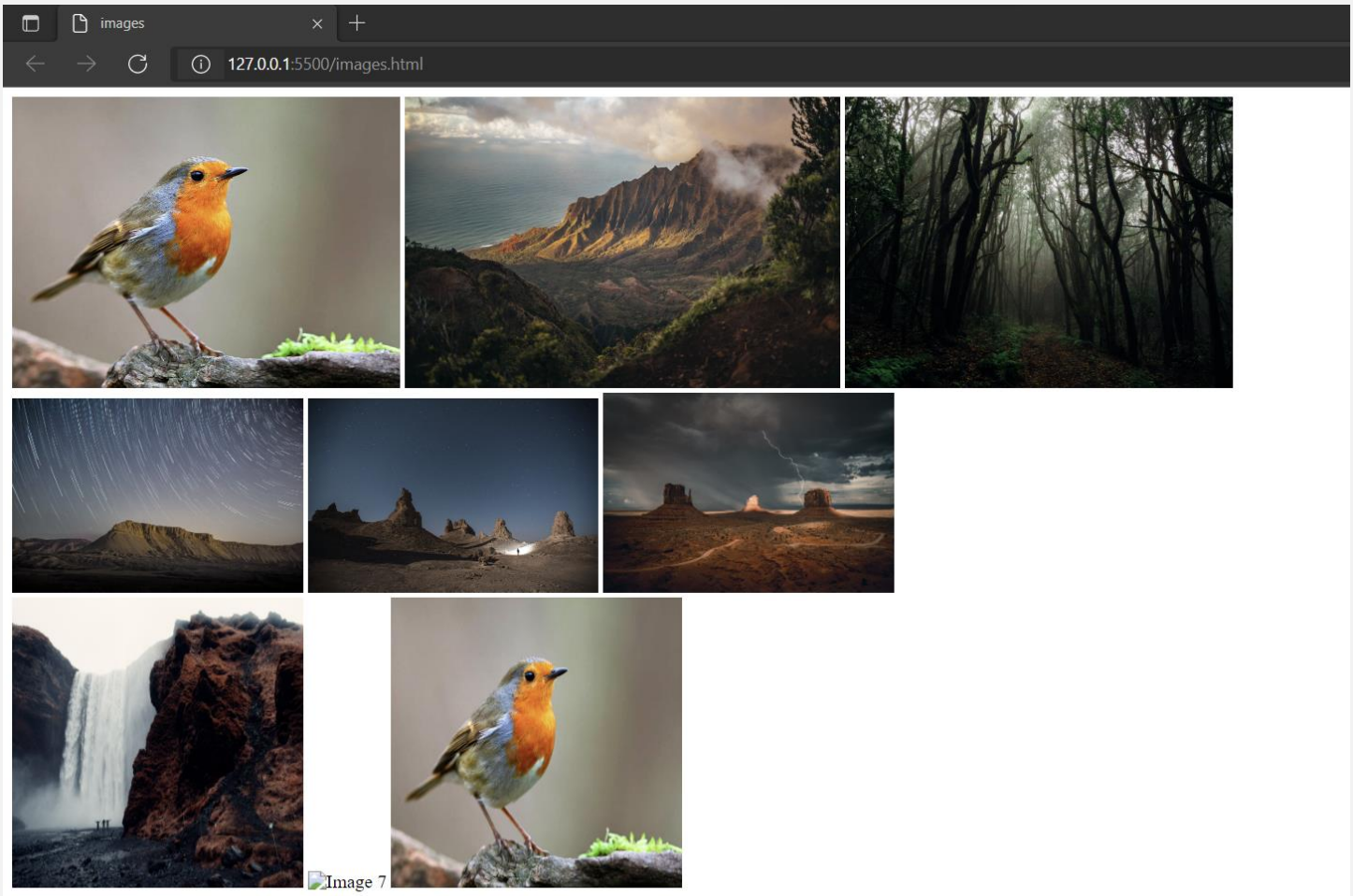
alt : تضيف هذه الخاصية نصاً سيتم عرضه في حالة عدم ظهور الصورة والذي قد يحدث بسبب خطأ في رابط الصورة.

width : خاصية تمكنك من التحكم ب عرض الصورة (يجب أن تكون القيمة ب البكسل).

height : خاصية تمكنك من التحكم ب إرتفاع الصورة (يمكن ان تكون القيمة بالبكسل او بالنسبة المئوية).

مشروع 2.3 : images.html





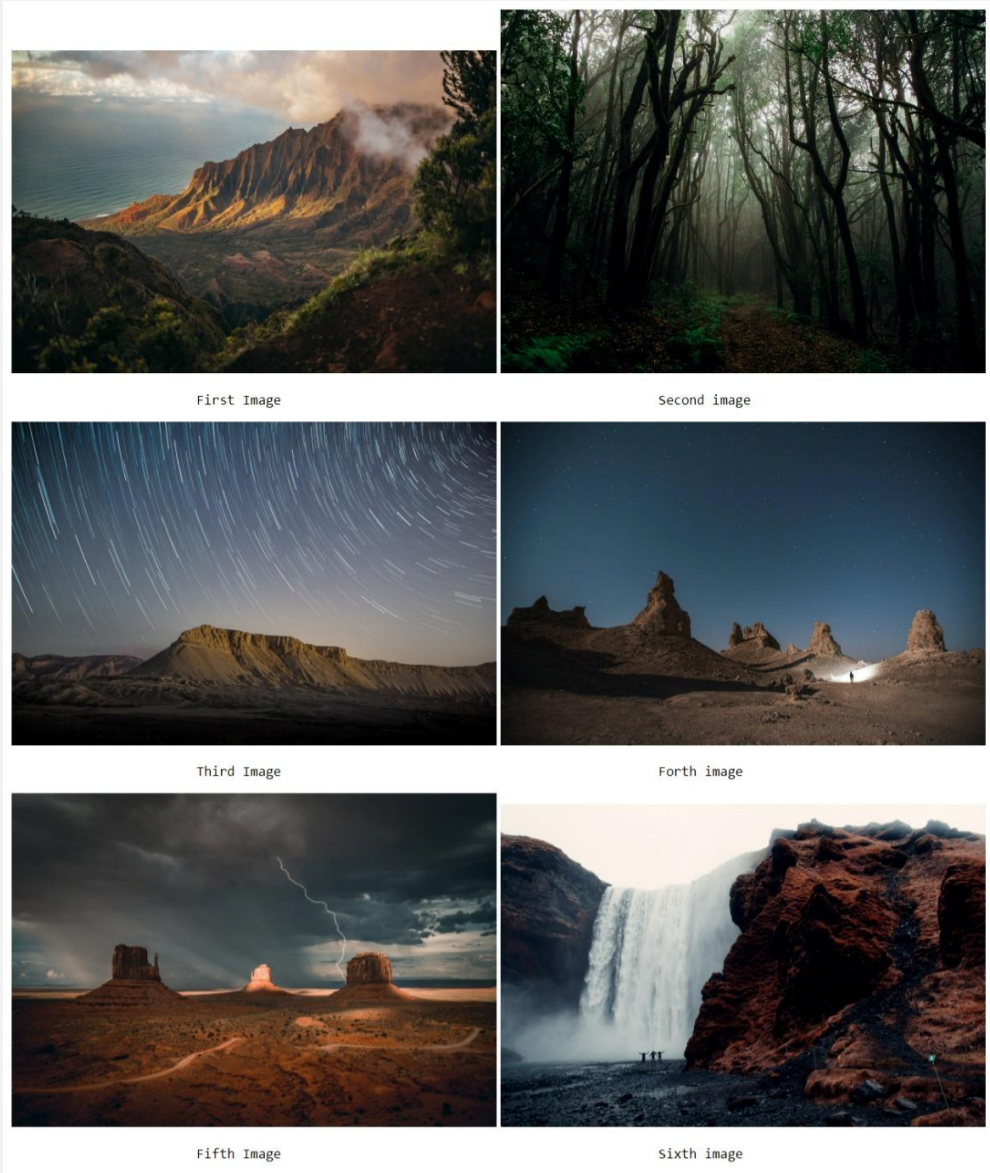
ستجد في هذا المشروع (**project 3**) مجلد **images** يحتوي على بعض الصور حيث سيتم عرضها في صفحة الويب.

image 7 ليست ظاهرة في الصفحة وذلك لإن رابط الصورة **imaegs/7.jpg** غير صحيح (ليست موجودة) , لذا سيظهر النص المكتوب بداخل خاصية ال **alt** .

أهم صيغ الصورة المدعومة :

- PNG
- JPEG
- WebP
- GIF
- SVG

تحدي 1 : صمم صفحة الويب التي سيظهر محتواها بهذه الطريقة :

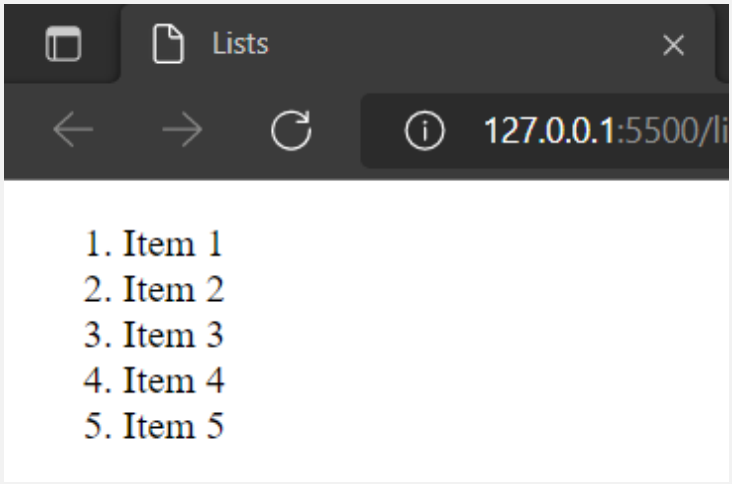


يمكنك رؤية الكود الخاص بهذا التحدي في مجلد **Challenges** بداخل مجلد **Challenge 1**

عناصر القوائم Lists : تستخدم هذه العناصر لـ إنشاء قائمة ويوجد نوعان :

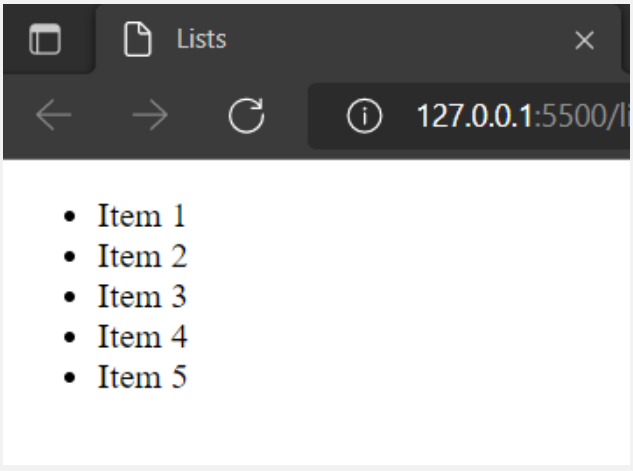
1. قائمة مرتبة **ordered list** : **<ol>** : تعرض مرقمة.

```
<ol>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
    <li>Item 5</li>
</ol>
```



2. قائمة غير مرتبة **unordered list** : **<ul>** : تعرض على شكل قائمة منقطة

```
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
    <li>Item 5</li>
</ul>
```



يمكن استخدام بعض الخصائص لتعديل طريقة عرض القوائم, فمثلاً :

خاصية **type** تمكننا من التعديل على نوع النقاط (دائرية او مربعة) في الـ **<ul>** , ويمكن ايضاً ان تستخدم في القائمة المرقمة والتي تحدد نوع الترقيم (احرف أو ارقام لاتينية).

سيتم عرض هذه الخاصية بالتفصيل في الفصل الرابع.

مشروع **2.4** :تطبيق كل ماتم تعلمه في هذا الفصل.

Name : Panda Manga

I am a web developer

I am a web developer and a web designer. I am currently working as a ***freelancer*** web developer.

Work experinces :

i. 6 years as a web developer
ii. 4 years as a mobile application Designer
iii. 3 year as a teacher

Languages :

- English
- Arabic
- Turkish
- Spanish

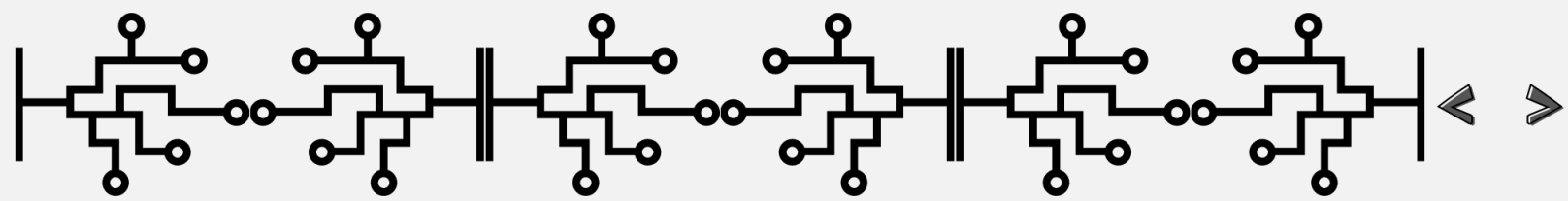
Skills :

1. Web Design
2. Web Development
3. Mobile Application Design
4. Mobile Application Development
5. Teaching

interests :

- Reading
- Watching Movies
- Watching TV Shows
- Watching Sports
- Watching Anime

ستجد هذا المشروع في مجلد **project4** .



إليك ايضاً بعض العناصر التي يمكنك استخدامها الان (يمكنك تجربتها ومعرفة خصائصها) :

العنصر	شرح
<address > </address>	استخدام العنصر <address> لعرض معلومات التواصل مع العنوان الفيزيائي
<blockquote> </blockquote>	يُشير العنصر <blockquote> إلى أنَّ النص الموجود داخله هو نص اقتباس طويل.
<abbr> </abbr>	يستخدم هذا العنصر للاختصارات ويوفِّر شرحًا لها، وعند إضافة الشرح يجب أن تحتوي الخاصية title على الشرح الكامل لهذا الاختصار .
<code> </code>	يمثل هذا العنصر جزءًا من شيفرة برمجية.
<q> </q>	إقتباس قصير
<details> </details>	يُستخدم العنصر <details> لعرض معلومات إضافية للمستخدم يمكنه إظهارها وإخفاؤها متى شاء. يمكن وضع أيّ نوع من المحتوى ضمن العنصر <details>، وتكون محتوياته مخفيةً افتراضيةً (إلا إذا ضُبِطَت الخاصية open).
	يستخدم هذا العنصر لـ إضافة عناصر في المنتصف.

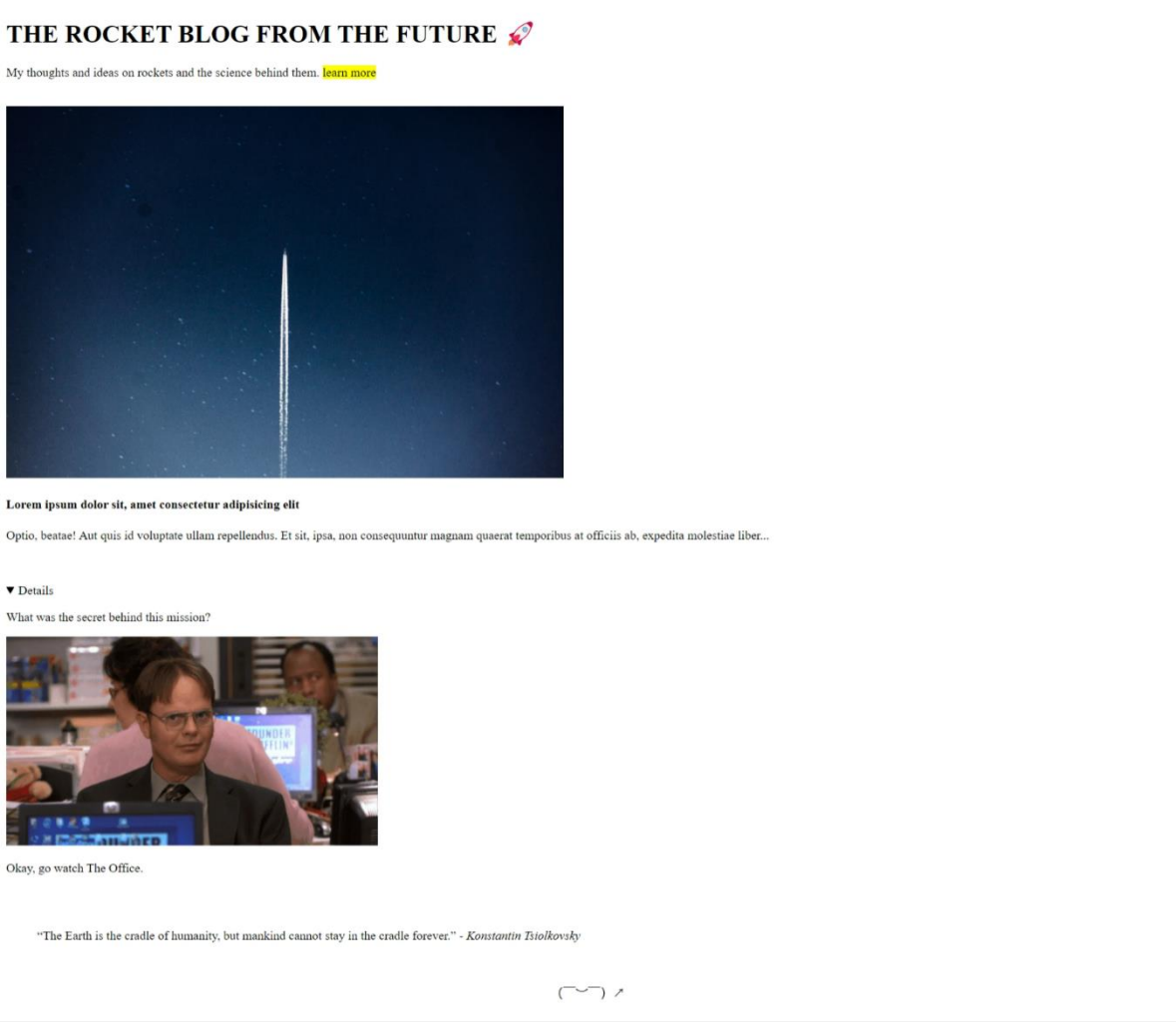
ملخص الفصل الثاني :

- تُستخدَم العناصر النصية لإضافة فقرات نصية , ويمكن الاستعانه ببعض العناصر لجعل طريقة عرضها أكثر جمالاً.
  - <h1> ... <h6>
  - <p> , <pre>
  - <small> , <u> , , , <del> , <mark> .... Etc
- يمكن إضافة الصور بإستخدام عنصر **img** من خلال إضافة رابط الصورة الي خاصية الـ **src** .
- يمكن إضافة قوائم الي صفحة الويب عن طريق إستخدام عناصر القوائم:
  - قائمة مرتبة
  - قائمة غير مرتبة

اسئلة الفصل الثاني:

1. ماهي الهيكلة الاساسية لجميع ملفات ال **HTML** ؟
2. ماهي الفروقات بين عناصر العناوين (**heading**) **<h6> ..... <h1>** ؟
3. ماهي انواع القوائم في ال **HTML** وماهي الخاصية التي يمكنك إستخدامها لتغيير نوع عرض القائمة ؟
4. ماهي وظيفة خاصية ال **alt** في عنصر الصور **img** ؟

تحدي الفصل الثاني :



يمكنك رؤية الصورة بوضوح في مجلد ال **Challenges** بداخل مجلد **challenge2** , والذي يحتوي ايضاً على الحل داخل مجلد ال **answer** .

3

الفصل الثالث : أساسيات الـ CSS

نبذة عن المحتوى :

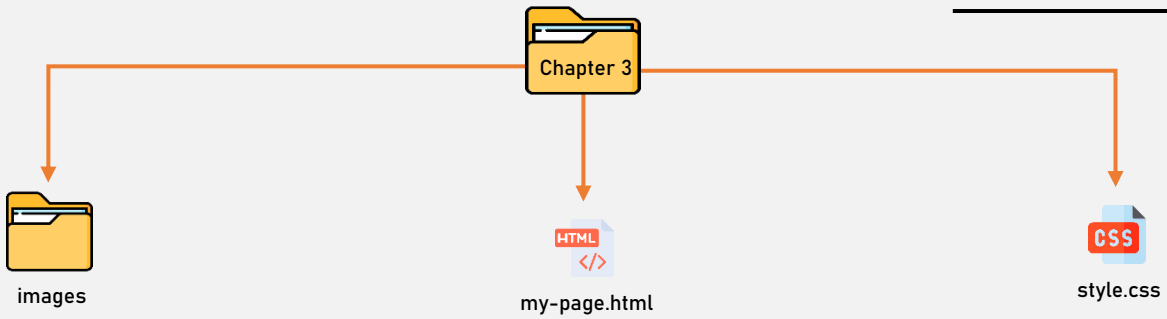
في هذا الفصل سنتعرف على واحدة من التقنيات الرئيسية المستخدمة لبناء صفحات الويب جنبًا إلى جنب مع لغة بناء صفحات الويب HTML.

والتي تمتاز بإضفاء السمات والخصائص على البنية الخاصة بصفحات الويب كالألوان والخطوط والأنماط التي تتخذها صفحة الويب.

اهداف الفصل :

1. التعرف على البنية الاساسية لملفات الـ CSS .
2. التعرف على الطرق الممكنة لإضافة خصائص الـ CSS الي عناصر الـ HTML .
3. التعرف على المحددات وطريقة إستخدامها.
4. تطبيق ماتم تعلمه : تحسين الصفحة شخصية.

ملفات الفصل الثالث :



مشروع نهاية الفصل الثالث :

Name : Panda Manga

I am a web developer

I am a web developer and a web designer. I am currently working as a *freelancer* web developer.

WORK EXPERINCES :

i. 6 years as a web developer
ii. 4 years as a mobile application Designer
iii. 3 year as a teacher

LANGUAGES :

- English
- Arabic
- Turkish
- Spanish

SKILLS :

1. Web Design
2. Web Development
3. Mobile Application Design
4. Mobile Application Development
5. Teaching

INTERESTS :

- Reading
- Watching Movies
- Watching TV Shows
- Watching Sports
- Watching Anime

بعد التعرف على طريقة إضافة عناصر صفحة الويب بإستخدام الـ HTML سننتقل الي تعّلم كيفية التحكم بطريقة عرض هذه العناصر , فمثلاً :
اريد ان تكون الصورة على شكل دائري او ان أجعل حجم ولون الكتابة مختلف والكثير من الخصائص التي سنتعرف عليها إثناء القيام ببعض المشاريع .

ماهي الـ CSS ؟

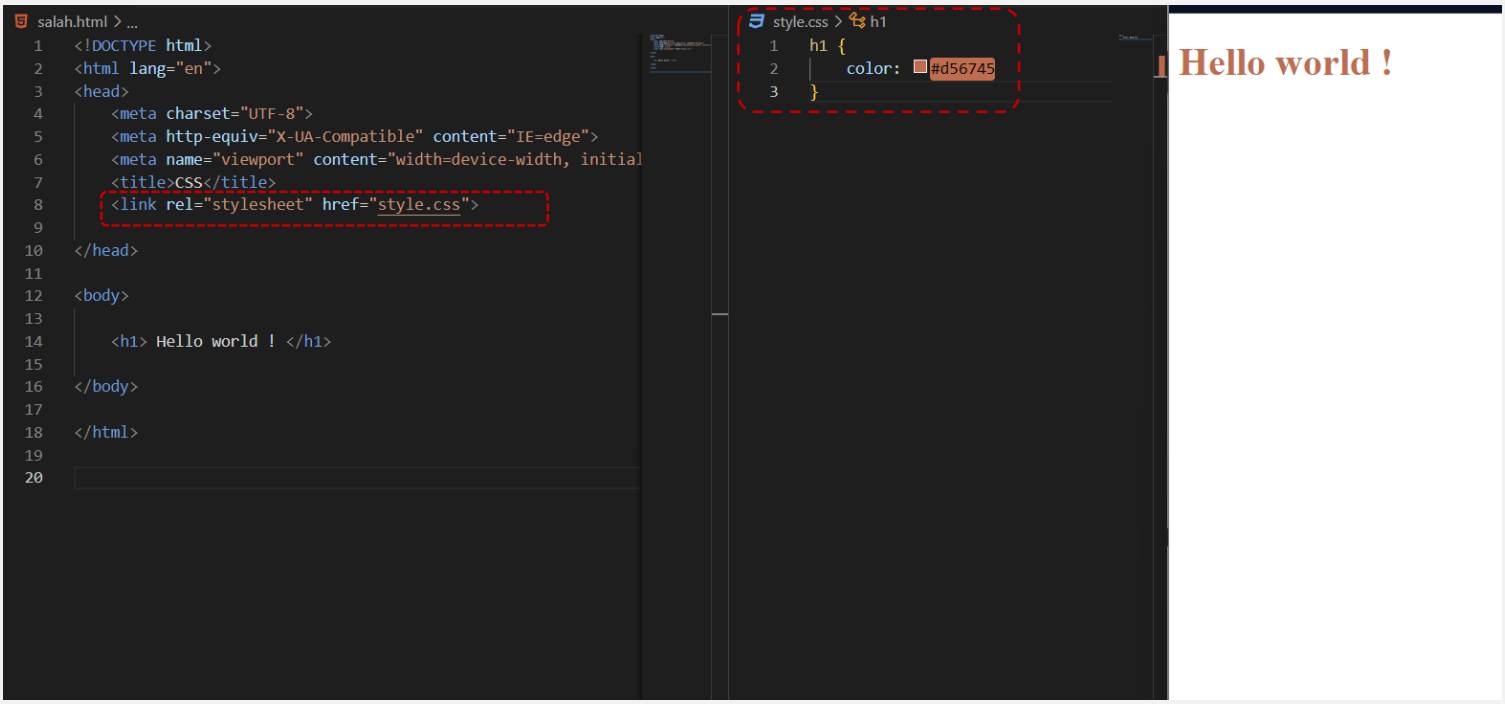
هي لغة تنسيقية مسؤولة عن تصميم شكل صفحات الويب وعن إضافة السمات والخصائص لعناصر الـ HTML.

ويتم ذلك عن طريق تحديد العنصر وإضافة بعض التعليمات التي يمكن من خلالها تحديد الخاصية (حجم العنصر , لون الخلفية , حجم ولون الخط لنصوص العنصر ... الخ).

كيف يمكن أضافة الـ CSS الي ملف الـ HTML ؟

الطريقة الاولى : يمكن إضافة تعليمات ال CSS بداخل عنصر الـ <style> </style> بداخل ملف الـ HTML

الطريقة الثانية : إنشاء ملف CSS منفصل ثم إستخدام عنصر الـ <link> بداخل الـ HTML ليتم ربطهم ببعض:



نلاحظ في الصورة وجود ملف salah.html خاص بعناصر الصفحة , وملف اخر style.css خاص بتصميم الصفحة.

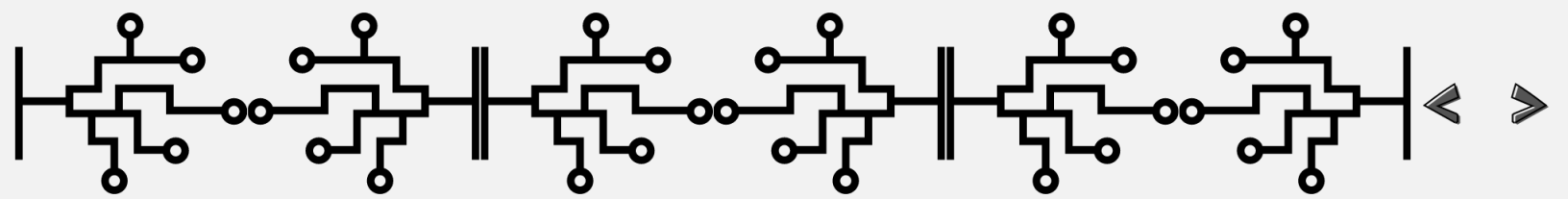
تم ربطهما ببعضهما بإستخدام عنصر : <link rel = “stylesheet” href=”style.css”>

تستخدم خاصية ال href بتحديد إسم الملف المراد ربطه بملف ال HTML . . .

الطريقة الاخيرة : إستخدام خاصية ال style بداخل كل عنصر وتحديد الخصائص لهذا العنصر:

```
<h1 style="color: #f9a059"> Hello world </h1>
```

سيتم في هذا الكتاب الاعتماد على الطريقة الثانية (ملف منفصل).



ماهي قوانين كتابة تعليمات الـ CSS ؟

```
selector {
    property-name: value;
    property-name: value;
    property-name: value;
    property-name: value;
}
```

- قبل كتابة تعليمات ال **CSS** يتم اختيار **selector** المحدد (العناصر) المراد التعديل على سماتها.
- كتابة التعليمات يتم بداخل الاقواس المنحنية **{ }** .
- تبدأ التعليمة بإختيار الخاصية (**property-name**), ثم نقطتين : وبعدها نحدد قيمة هذه الخاصية (**value**) وبعدها ننهي التعليمة بالفاصلة المنقطوة **;** حتى نستطيع إضافة خصائص اخرى...

مثلاً :

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>

    <title>HTML & CSS</title>

    <link rel="stylesheet" href="style.css">

</head>
<body>

    <h1> Hello world </h1>
    <h1> This is Heading </h1>
    <p> This is Paragraph </p>
    <h2> This is h2</h2>

</body>
</html>
```

style.css

```
1
```

Hello world

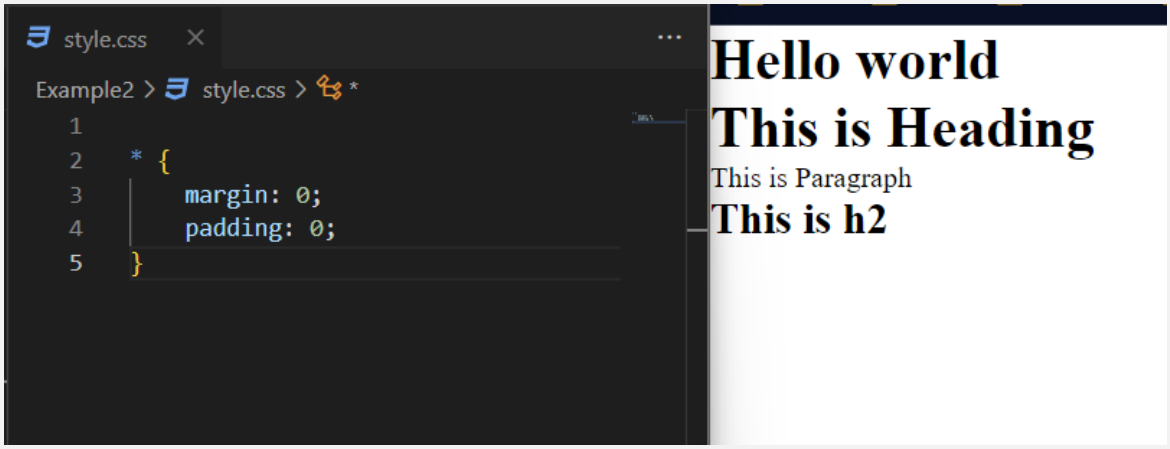
This is Heading

This is Paragraph

This is h2

هكذا ستكون شكل الصفحة دون إضافة اي خصائص عليها .

دعنا نعمل على إضافة بعض خصائص ال **CSS** على هذه العناصر ونرى كيف يمكننا تعديل طريقة ظهور هذه العناصر .



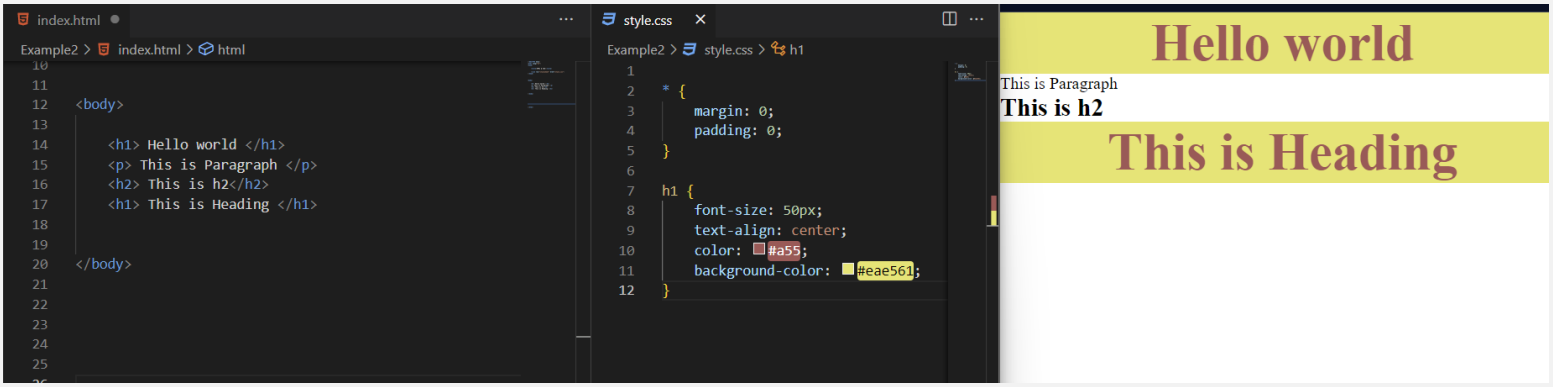
سيصبح شكل الصفحة بهذا الشكل بعد إضافة هذه الخصائص , اليك ماحصل :

اولاً : تم أختيار اسم المحدد * والذي يعني جميع العناصر , لذا فان الخصائص الموجودة بداخل هذا المحدد ستنطبق على جميع العناصر.

ثانياً : تلقائياً تحتوي عناصر الـ HTML على هوامش (مسافة) من جهاتها الاربع , لذلك عند إستخدام خاصية الـ margin: 0; سيتم الغاء كل هذه الهوامش التلقائية.

ثالثاً: ايضاً تلقائياً تحتوي هذه العناصر على بعض الحواشي (مساحة داخلية) , لذالك تم إستخدام padding: 0; لـ الغاء الحواشي التلقائية لجميع العناصر.

دعنا الان نضيف بعض الخصائص على عناصر الـ h1 :



اما الان ستصبح الصفحة بهذا الشكل بعد إضافة بعض الخصائص في الـ CSS , لذا دعنا مالذي حصل هنا:

اولاً : تم اختيار المحدد (h1) ويعني انه جميع عناصر ال **h1** سيتم التأثير عليها بهذه الخصائص .

ثانياً : خاصية الـ **font-size** : تتحكم هذه الخاصية بحجم الخط .

ثالثاً : خاصية الـ **text-align** : تحدد هذه الخاصية اتجاه عرض النص , وتم اختيار الاتجاه بالمنتصف **center** .

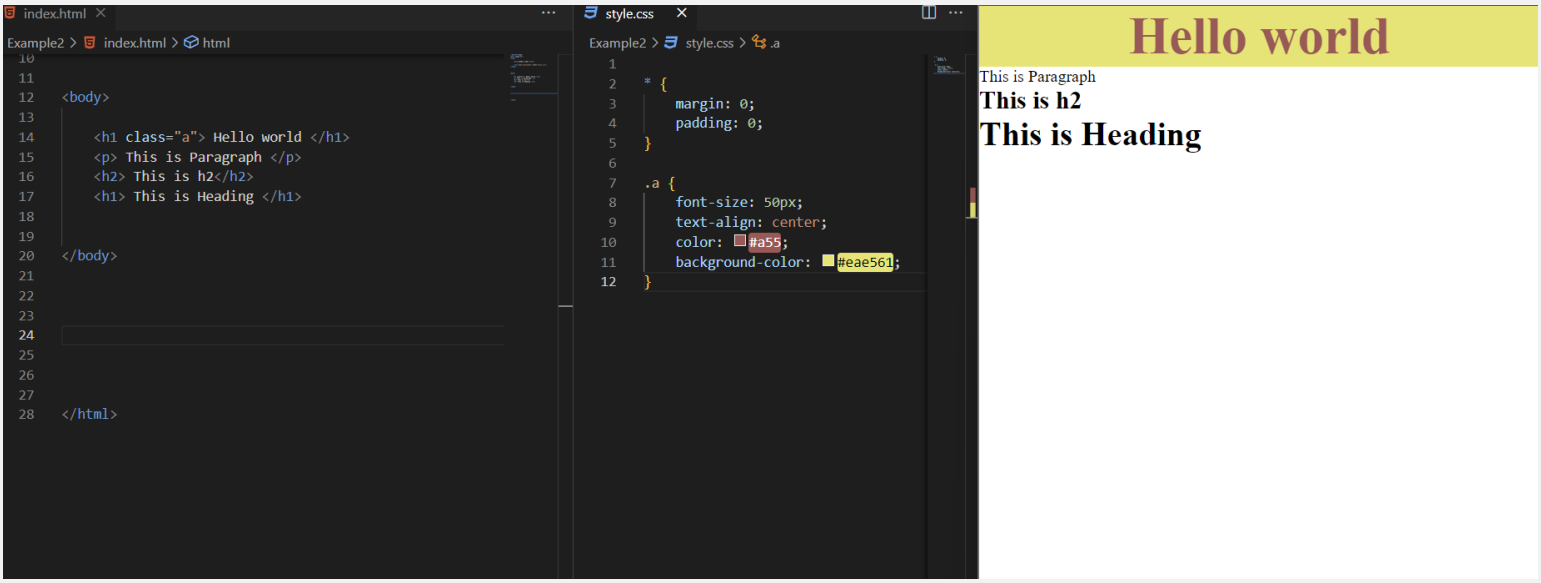
رابعاً : خاصية الـ **color** : تحدد هذه الخاصية لو خط الكتابة.

خامساً : خاصية الـ **background-color** : تستخدم هذه الخاصية لتغيير لون خلفية العنصر.

ماذا لو اردت تطبيق هذه الخصائص على واحد من عناصر ال h1 وليس على كل هذه العناصر ؟

يمكن القيام بذالك بأكثر من طريقة , لكن اسهل هذه الطرق هي ب تصنيف العناصر , ويمكن القيام بذالك بإستخدام خاصية الـ **class** داخل العناصر ثم إستخدام اسم هذا الصنف **(class)** لتعديل خصائص الـ **CSS** .

دعنا نجرب تطبيق الخصائص السابقة فقط على عنصر **h1** الذي يحتوي على " **Hello World** " :



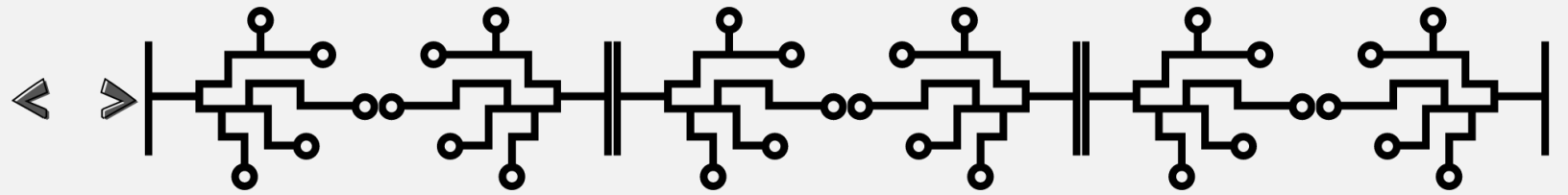
اولاً : تم اضافة خاصية ال **class** الي عنصر ال **h1** المراد تغيير خصائصه , وإعطائه اسم (**a**).

ثانياً : في ملف الـ **CSS** تم تغيير اسم المحدد الي (**.a**) والذي يعني العناصر التي تنتمي الي هذا الصنف.

ثالثاً : فقط العنصر الذي يحتوي على هذا الصنف **a** سيتم تطبيق هذه الخصائص عليه.

قم بتجربة إضافة خاصية ال **class** الي عنصر اخر واجعل اسم الصنف **a** , **مالذي حدث ؟**

*ملاحظة : اسم المحددات الخاصة ب الاصناف **class** تحتوي على نقطة في البداية .*

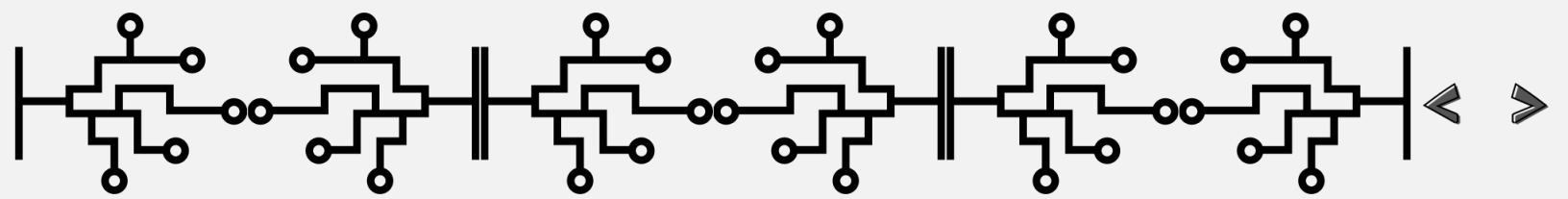


انواع المحدادت في ال CSS :

1. محدد عام (*) : ويعني جميع العناصر. `* { . . . }`
2. محدد النوع (**type selector**) اي نوع العنصر الذي تم إستخدامه في مستند ال HTML . `h1 { . . . }`
3. محدد الصنف (**class selector**) اي العناصر التي تنتمي لهذا الصنف. `.a { . . . }`
4. محدد المُعرِّف (**ID selector**) اي العنصر الذي يحتوي على ال ID . `#b { . . . }`

وتوجد محدادت اخرى سيتم التعرف عليها لاحقاً.

لنتعرف على بعض خصائص ال **CSS** من خلال انجاز مشروع **3.1** (**project1**) .



لاحظ ان مستند الـ نفسه الموجود في مشروع **2.4** لكن بوجود بعض الإضافات :

- لربط المستند بملف ال **CSS** <link rel="stylesheet" href="style.css”>
- تم إضافة خاصية الـ **class** لبعض العناصر حتى يتم إستدعاء هذا الصنف وإضافة بعض الخصائص لها.

```html
<!DOCTYPE html>
<html lang="en">

<head>

    <title>Personal Page</title>
    <link rel="stylesheet" href="style.css">
</head>

<body>
    <img src="images/me.png" alt="Salah Alkmali" width="275px">

    <h1 class="name">Name : Panda Manga</h1>

    <h3 class="career">I am a web developer</h3>

    <p class="career box">
        I am a web developer and a web designer. I am currently working as a
        <strong> <i> freelancer</i></strong> web developer.
    </p>

    <hr>
    <br>

    <h4 class="heading-text">Work experinces : </h4>

    <ol type="i" class="red-box">
        <li> 6 years as a web developer</li>
        <li> 4 years as a mobile application Designer </li>
        <li> 3 year as a teacher</li>
    </ol>

    <br>

    <h4 class="heading-text">Languages : </h4>

    <ul type="square" class="white-box">
        <li>English</li>
        <li>Arabic</li>
        <li>Turkish</li>
        <li>Spanish</li>
    </ul>

    <h4 class="heading-text">Skills : </h4>

    <ol class="black-box">
        <li>Web Design</li>
        <li>Web Development</li>
        <li>Mobile Application Design</li>
        <li>Mobile Application Development</li>
        <li>Teaching</li>
    </ol>

    <br>

    <h4 class="heading-text">interests : </h4>

    <ul>
        <li>Reading</li>
        <li>Watching Movies</li>
        <li>Watching TV Shows</li>
        <li>Watching Sports</li>
        <li>Watching Anime</li>
    </ul>

</body>

</html>
```

النتيجة بعد ربط ملف ال HTML بملف ال CSS :

Name : Panda Manga

I am a web developer

I am a web developer and a web designer.
I am currently working as a *freelancer* web developer.

WORK EXPERINCES :

i. 6 years as a web developer
ii. 4 years as a mobile application Designer
iii. 3 year as a teacher

LANGUAGES :

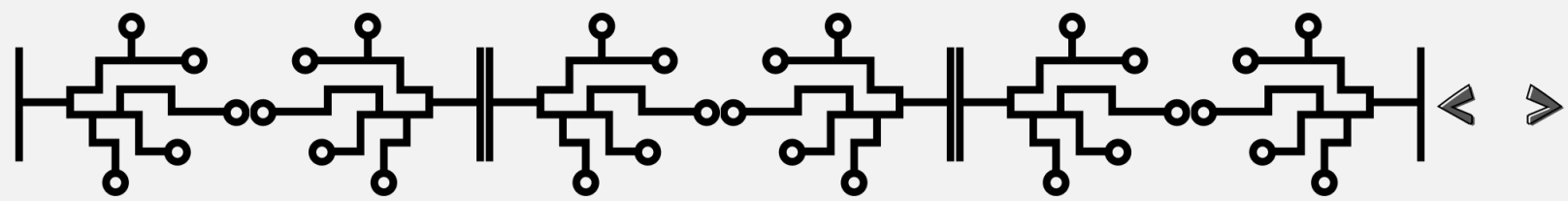
- English
- Arabic
- Turkish
- Spanish

SKILLS :

1. Web Design
2. Web Development
3. Mobile Application Design
4. Mobile Application Development
5. Teaching

INTERESTS :

- Reading
- Watching Movies
- Watching TV Shows
- Watching Sports
- Watching Anime



يمكنك التعديل على هذه الخصائص ستجد هذا المشروع في مجلد **project1** الموجود داخل مجلد **chapter 3**.

دعنا نرى ونحلل هذه الخصائص التي تم إستخدامها في هذا المشروع :

محدد الـ **body** وهو العنصر الذي يحتوي على كل عناصر صفحة الويب.

```css
body {
    font-family: 'Open Sans', sans-serif;
    line-height: 1.5;
    color: #333;
    background: #e9dfc8;
}
```

font-family : الخاصية التي تتحكم بنوع خط الكتابة.

line-height : لضبط المسافة المستخدمة للاسطر.

color : لتغيير لون خط الكتابة.

background : لتغيير الخلفية , وهنا تم تغييرها الي لون.

محدد ال **img** (الصورة).

```css
img {
    width: 450px;
}
```

width : تحديد عرض الصورة.

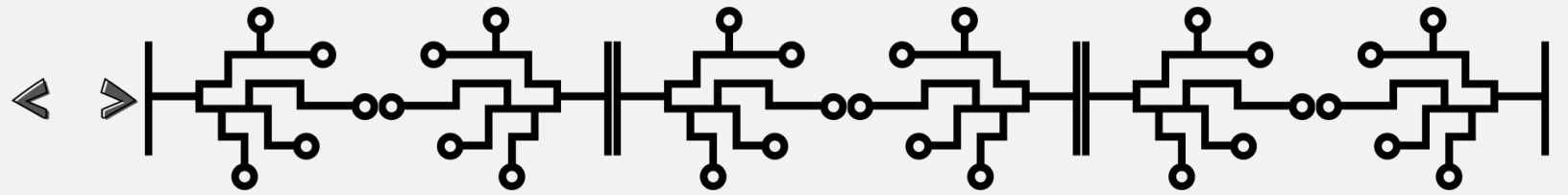
محدد الـ **.name** والذي سيؤثر على جميع العناصر التي تحتوي على هذا الاسم في خاصية الـ **class** :

```css
.name {
    font-size: 32px;
    margin-left: 50px;
    font-weight: bold;
    color: #2b2828;
}
```

font-size : التحكم بحجم خط الكتابة.

margin-left : التحكم بالهامش من جهة اليسار ,
سيبتعد العنصر بمقدار **50px** عند حدود الصفحة من جهة اليسار

font-weight : تحدد ثخانة خط الكتابة.



محدد الـ career. (صنف) :

```css
.career {
    width: 320px;
    font-size: 16px;
    margin-left: 50px;
    color: #2b2828;
    padding: 10px;
}
```

padding : تتحكم بمقدار الحواشي في العنصر , اي مدى إبتعاد محتويات هذا العنصر عن حدوده.

محدد الـ box. (صنف) :

```css
.box {
    border: 1px solid #2b2828;
}

```

border : التحكم بإطار العنصر حيث ان :

- 1px هي حجم إطار العنصر (من جميع الجهات).
- solid : نوع الإطار (خط مستقيم).
- #2b2828 : لون الإطار .

ملاحظة : تم إضافة أكثر من صنف لعنصر p كما موجود بالصورة في الاسفل , وهذا يعني انه يمكننا إضافة أكثر من صنف واحد لكل عنصر , يجب الفصل بين هذه الإصناف بمسافة.

```html
<p class="career box">
        I am a web developer and a web designer. I am currently working as a
        <strong> <i> freelancer</i></strong> web developer.
    </p>
```

محدد ال hr (خط افقي) :

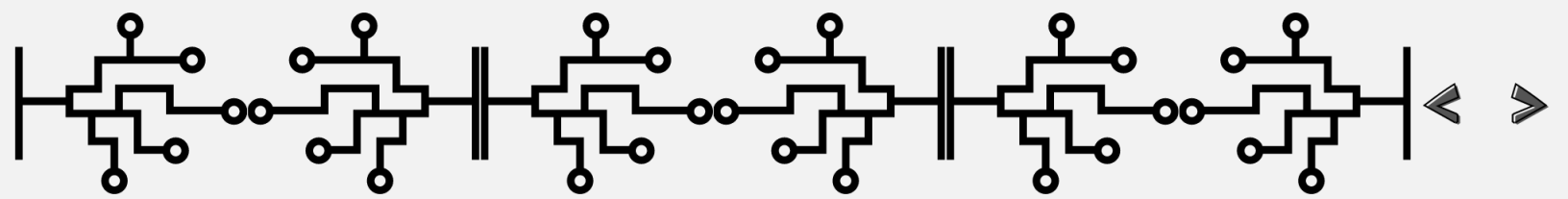
```css
hr {
    width: 80%;
    height: 1.5px;
    background-color: #2b2828;
    margin: 20px auto;
}
```

width: 80%; : تم تحديد عرض هذا الخط بنسبة 80% من حجم العنصر الأب (body) .

height : طول هذا العنصر.

margin: 20px auto; : تم تحديد الهوامش هنا :

- 20px من الاعلى والاسفل.
- auto من اليسار واليمين , ويعني انها سيوزع المساحة المتاحة بالتساوي بين اليمين واليسار.



محدد الـ .heading-text (صنف) :

```
.heading-text {
    font-size: 20px;
    color: #267940;
    margin-left: 30px;
    text-decoration: underline;
    text-transform: uppercase;
}
```

text-decoration : تحدد شكل الزخرفة :
underline خط اسفل النص.

text-transform : تحدد طريقة عرض أحرف النص,

uppercase , سيتم عرض النص بأحرف كبيرة.

```
.red-box {
    width: 350px;
    border: 1px solid #000000;
    border-radius: 4px;
    background-color: #e17070;
    padding-top: 15px;
    padding-bottom: 15px;
    margin-left: 20px;
}
```

```
.white-box {
    width: 350px;
    border: 1px solid #000000;
    border-radius: 12.5%;
    background-color: #efefef;
    padding-top: 15px;
    padding-bottom: 15px;
    margin-left: 20px;

}

```

```
.black-box {
    width: 350px;
    border: 1px solid #ffffff;
    background-color: #2b2b2b;
    padding-top: 15px;
    padding-bottom: 15px;
    margin-left: 20px;

}

```

تشترك هذه الاصناف الثلاثة بكثير من الخصائص , لذا دعنا في البداية نتحدث عن الخصائص التي نراها لأول مرة , ثم دعنا نكتب هذه التعليمات بطريقة اخرى تجنبنا تكرار بعض هذه التعليمات :

border-radius : تسمح لنا بتحديد شكل زوايا العنصر (الاصل فيها ان تكون حاده , كما تراه في الصندوق الاسود).

padding-top : اضافة حواشي من الاعلى مساحة بين محتويات العنصر وإطار العنصر من الاعلى.

padding-bottom : اضافة حواشي من الاسفل مساحة بين محتويات العنصر وإطار العنصر من الاسفل.

يمكن كتابة التعليمات الموجودة بالاعلى بطريقة اخرى تجنباً لتكرار بعض التعليمات بهذه الطريقة :

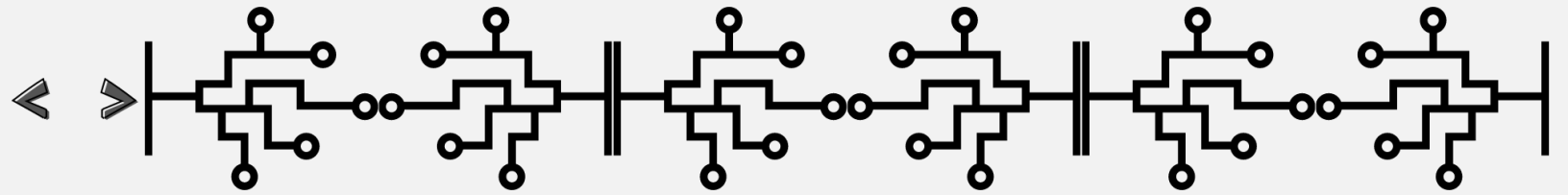
```
.red-box , .white-box , .black-box {
    width: 350px;
    border: 1px solid #000000;
    padding-top: 15px;
    padding-bottom: 15px;
    margin-left: 20px;
}
```

```
.red-box {
    background-color: #e17070;
    border-radius: 4px;
}

.white-box {
    border-radius: 12.5%;
    background-color: #efefef;

}

.black-box {
    background-color: #2b2b2b;
}
```



.black-box , .white-box , .red-box : هذا المحدد يعني تطبيق التعليمات على العناصر التي تحتوي على إحدى هذه الإصناف. لذالك ستنطبق كل هذه الخصائص على الصناديق الثلاثة.

بعدها تم إنشاء محدد خاص بكل صنف , لإضافة السمات الخاصة بهذا الصنف.

ملاحظة : يتم الفصل بين المحددات بإستخدام الفاصلة , .

المحددات .black-box li , .red-box li. يعني ان التعليمات الموجودة في الاسفل ستنطبق على عناصر الـ li الموجودة داخل العنصر الذي يحتوي على صنف بإسم الـ red-box أو black-box.

```
.red-box li ,
.black-box li {
    color: #efefef;
    margin-bottom: 5px;
}
```

ملخص الفصل الثالث :

- تستخدم الـ CSS لإضافة الخصائص التي ستظهر بها عناصر الصفحة , ويمكن إضافة هذه الخصائص بـ ثلاث طرق :
  1. إستخدام عنصر ال <style> بداخل ملف الـ html.
  2. إستخدام ملف خارجي (.css) وربطه مع ملف الـ html بإستخدام عنصر ال <link> .
  3. إستخدام خاصية ال style داخل العنصر.

```
h1 {
font-size: 12px;
}
```

- في حالة إستخدام الطريقة الاولى او الثانية فإنه قبل إضافة الخصائص يتم الاعلان عن المحددات (selectors) وتعني العناصر المراد التحكم بخصائصها ويتم وضع الخصائص بين اقواس مربعة { }

```
h1 , p {
font-size: 12px;
font-color: red;
}
```

- يمكن أختيار أكثر من محدد وإضافة الخصائص لهذه المحددات , يتم الفصل بين هذه المحددات بإستخدام الفاصلة , .

- يجب أن تنتهي كل خاصية في ال css ب فاصلة منقوطة ; حتى تكون قادر على إضافة خصائص اخرى .

4

الفصل الرابع : سمات عناصر الـ HTML (Attributes)

نبذة عن المحتوى :

اهداف الفصل :

ملفات الفصل الرابع :