

HTML

This Book brought to you by Ayoub Ayaou about HTML

? Html لغة ما هي

أصدرت لغة الـ HTML سنة 1993 حيث تعتبر من أهم اللغات في تطوير المواقع الالكترونية حيث هي أول شيء ستتقابل معه في مسارك كمطور موقع وجميع محركات البحث مثل جوجل يبنج تتعامل مع هذه اللغة، و هدف لغة html هو هيكلة الموقع الإلكتروني أو لبناء هيكل الموقع إعتبرها مثل هيكل البيت يعني تعتبر لغة html هي هيكل جميع المواقع على الإنترنت ولكن تحتاج معها إلى ثلاثة لغة لتكون مطور ويب محترف وهي css و JavaScript **هنا مقال يتحدث عن لغة css وهذا آخر يتحدث عن الـ JavaScript** لكي تكتشفهما أكثر وتصبح محترف و لغة الـ HTML مرت من عدة إصدارات وصولا إلى الإصدار الخامس

- +HTML
- HTML2.0
- HTML3.2
- HTML4.01
- HTML5
- HTML5.1
- HTML5.2

من اللغة هو الذي يجب تعلمه في بدأ مشوارك في مجال الويب مع بعض الأوصمة من الإصدار الرابع من اللغة و تعتبر لغة html لغة تصويفية ليس لغة برمجة مما يحيل على سهولة هذه اللغة حيث يعتبر Syntax الخاص باللغة سهل جدا كله عبارة عن علامة أو أوصمة باللغة الإنجليزية tags تتكون من البداية والنهاية نفتح الوسم بعلامة الأصغر من و نكتب إسم الوسم و عند الإنتهاء من الإسم نكتب علامة الأكبر من و نكتب المحتوى في الداخل و نكتب مرة أخرى علامة الأصغر من ونحدد نهايته بعلامة السلاش / و نكتب نفس الإسم و ننتهي بعلامة الأصغر من كما يظهر

لكم في الكود هنا:

```
<name-tag>content</name-tag>
```

ولدينا طريقة أخرى وهي بدون قفلة الوسم نبدأ بعلامة الأصغر من و نكتب إسم الوسم ونغلق مباشرة بعلامة السلاش / ومن ثم علامة الأكبر من كما ظاهر معكم في الكود طبعا:

```
<name-tag/>
```

يمكننا أيضا كتابة شيء إسمه السمات أو attributes تمكينا من التحكم بالوسم بطريقة سهل وتعطيه المزيد من الإضافات هذه هي طريقة كتابة السمة عند الأوسمة التي لها بداية ونهاية و التي ليست لها نهاية:

```
<name-tag attribute-name="value">content</name-tag>
```

```
<name-tag attribute-name="value"/>
```

هنا مثال لكود بلغة HTML عند تشغيله يعطيك النص بطريقة كبيرة و براز هو خاص بالعناوين:

```
<h1>big title</h1>
```

فكم ترى داخل إسم الوسم نطبع وصف للمحتوى والمحتوى سيتغير معنا عند التشغيل الأمر بسيط جدا نفس طريقة الكتابة كما ترى فقط يختلف الإسم ودوره هذا يحتاج إلى فيديو لهذا لدينا في قناتنا دورة لغةHtml يمكنك تفقدها وكلمة HTML اختصار لكلمة hyper text Markup language هناك العديد من القنوات التي تعلم لغة html يمكنك تفقد قسم **الأسئلة** سوف تجد سؤال حول قنوات تعلم البرمجة و شكرا لأن قرأت إلى هنا و وفقك الله.

ما هو syntax الخاص بلغة html

HTML هي لغة ترميز تستخدمن في تصميم صفحات الويب. وت تكون HTML من عناصر وعناصر فرعية (sub-elements) وأدوات وعناصر تحكم. في هذه المقالة، سنناقش بعض من syntax الخاصة بلغة HTML وسنوضح الفروق الرئيسية بين هذه العناصر. ترميز العناصر: تعتمد HTML على ترميز العناصر بواسطة الزوج الافتتاحي والزوج الختامي، على سبيل المثال:

< p > هذه فقرة نصية < /p >

في هذا المثال، يتم تعريف العنصر `<p>` باستخدام الزوج الافتتاحي `<p>` ويتم إغلاقه باستخدام الزوج الختامي `</p>`. وكل محتوى العنصر يتم تضمينه بين الزوجين الافتتاحي والختامي.

العناصر الداخلية: يمكن لعناصر HTML أن تحتوي على عناصر فرعية، والتي تسمى العناصر الداخلية. في المثال التالي، يتم تضمين النص في العنصر `` داخل العنصر `<p>`:

< p > < strong > فقرة مهمة < /strong > هذا النص في < /p >

في هذا المثال، نعرف العنصر `<p>` باستخدام الزوج الافتتاحي `<p>` ويتم إغلاقه باستخدام الزوج الختامي `</p>` ، ويتم تضمين العنصر `` داخله بواسطة الزوجين `` و ``.

الصور: تستخدم العناصر `` لإدراج الصور في صفحات الويب، على سبيل المثال:

< img src="image.jpg" alt="your text" >

يتم تعريف العنصر `` باستخدام الزوج الافتتاحي `` والذي يحتوي على معرف `src` الذي يحدد مسار موقع الصورة، ومعرف `alt` الذي يحدد النص البديل للصورة.

وهكذا إنتهينا من مقالتنا المتواضعة وتعلمنا على طرق الكتابة بلغة HTML، أتمنى أن تعجبكم المقالة وتحياتي لكم.

أساسيات لغة html

HTML هي لغة ترميز تستخدمن في تصميم صفحات الويب والتي تعتبر الأساس الأساسي لأي موقع وت تكون هذه اللغة من مجموعة من العناصر التي تستخدمن لتحديد محتوى الصفحة وترتيبه وتنظيمه. في هذه المقالة، سوف نتحدث عن بعض الأساسيات الضرورية لفهم لغة **HTML** ومعرفة كيفية استخدامها.

تعريف الصفحة:

يتم تعريف صفحة **HTML** عن طريق البدء بعلامة **HTML** وإغلاقها في نهاية الصفحة. هذا هو الشكل العام لصفحة **HTML**:

```
<!DOCTYPE html>
<html>
<head>
    <title>/<title>عنوان الصفحة
</head>
<body>
    محتوى الصفحة هنا.
</body>
</html>
```

تبدأ الصفحة بتعريف **DOCTYPE** والذي يحدد نوع المستند، وبعد ذلك يتم فتح علامة **<html>** التي تحدد بداية الصفحة. يتم تعريف رأس الصفحة باستخدام العنصر **<head>** ويتم تعريف العنوان الخاص بالصفحة باستخدام العنصر **<title>**. يتم وضع محتوى الصفحة كله في عنصر **<body>**.

العناصر الأساسية:

تحدد العناصر الأساسية في **HTML** محتوى الصفحة وتنظيمه. هذه بعض العناصر الأساسية:

:**<h1>**

يتم استخدام هذا العنصر لتحديد عنوان كبير للصفحة.

<h1>/>عنوان كبير أو العنوان الرئيسي

: يتم استخدام هذا العنصر لتحديد فقرة من النص.

< p > هنا فقرة نصية < /p >

: يتم استخدام هذا العنصر لإنشاء رابط إلى صفحة أخرى.

< a href="https://test.com" > هنا رابط < /a >

: يتم استخدام هذا العنصر لإضافة صورة إلى الصفحة.

< img src="imag.jpg" >

وهكذا إنتهينا من مقالتنا لم نذكر كل العناصر ولكن كل عنصر له قسمه الخاص سنذكره بالتفصيل هنا فقط تعرفنا على الأساسية شكرًا لكم ووفقكم الله.

ما هي العناصر في لغة html؟

HTML هي لغة ترميز تستخدمن في تصميم صفحات الويب. تتالف صفحة HTML من مجموعة من العناصر التي تستخدم لتحديد محتوى الصفحة وتنظيمه. في هذه المقالة، سوف نتحدث عن بعض العناصر الشائعة في لغة HTML وكيفية استخدامها. العناوين < h1 > إلى < h6 > يتم استخدام هذه العناوين لتحديد عنوان المقالة أو القسم. تبدأ بـ < h1 > كأكبر حجم وتنتهي بـ < h6 > كأصغر حجم. هذا هو مثال عن استخدام عنوان < h1 >:

< h1 > عنوان المقالة < /h1 >

الفقرات < p > يتم استخدام هذا العنصر لتحديد فقرة من النص. يمكنك استخدامه

لكتابة المقالة أو تحديد نص في الصفحة. هذا هو مثال عن استخدام `<p>`:

```
<p>/<!-- النصية -->
```

الصور `` يتم استخدام هذا العنصر لإضافة صورة إلى الصفحة. يجب عليك تحديد موقع الصورة عن طريق الخاصية `src`. يمكنك أيضًا إضافة النص البديل عند عدم تمكن المتصفح من عرض الصورة عن طريق الخاصية `alt`. هذا هو مثال عن استخدام

```
:<img>
```

```

```

الروابط `<a>` يتم استخدام هذا العنصر لإنشاء رابط إلى صفحة أخرى. يتم تحديد الرابط الذي يربط به العنصر عن طريق الخاصية `href`.
هذا هو مثال عن استخدام `<a>`:

```
<a href="https://test.com">/<!-- زيارت موقعنا -->
```

وهكذا إنتهينا تعرفنا على العناصر في لغة `html` نلتقي في دروس السمات وشكرا لكم لدعمنا وفقكم الله.

ما هي السمات في لغة `html`؟

تُستخدم لغة `HTML` لتصميم صفحات الويب، وتتضمن عناصر `HTML` وسموم العرض والتنسيق والإدخال. ولكل عنصر في `HTML` مجموعة من السمات أو `Attributes` التي تحدد خصائص العنصر. في هذه المقالة، سنتحدث عن أهمية السمات في لغة `HTML` وكيفية استخدامها.

الشكل العام للسمات في لغة `HTML`:

ت تكون السمات في لغة `HTML` من اسم السمة وقيمة السمة وتكون السمة داخل علامات العنصر. ويتم تحديد اسم السمة بعد اسم العنصر مباشرة ثم يليه قيمة السمة بين علامة الاقتباس.

```
<element-name attribute_name="attribute_value">content</element-name>
```

أمثلة على السمات:

.CSS: تستخدم لتحديد العنصر وتنسيقه من خلال **class**.

```
<p class="paragraph">This is p</p>
```

.CSS: يستخدم لتحديد العنصر بشكل فريد في الصفحة وتنسيقه باستخدام **id**

```
<p id="paragraph">This is p</p>
```

.**href**: تستخدم لتحديد الرابط للعنصر **a**

```
<a href="https://test.com">visit test.com</a>
```

src: تستخدم لتحديد المسار لملف الصورة في عنصر **img**

```

```

style: تستخدم لتنسيق العنصر مباشرةً باستخدام **CSS** كما ترون، إستبدلنا اللون الخاص بها لللون الأحمر.

```
<p style="color: red"></p>
```

السمات المخصصة:

تتيح لنا لغة HTML إمكانية إنشاء سمات مخصصة لتخزين بيانات إضافية في العناصر، وذلك عن طريق استخدام "-data" وبعد ذلك اسم السمة المخصصة. سنتحدث عن السمات المخصصة في لغة HTML وكيفية استخدامها.

تعريف السمات المخصصة:

السمات المخصصة هي سمات يتم إنشاؤها من قبل المطورين لتخزين بيانات إضافية في العناصر، ويتم تحديدها باستخدام "-data" متبوعة بعد ذلك بالاسم الذي تريد إضافته. وعندما تستخدم هذه السمات في صفحة HTML، يمكن للمطورين استخدامها لتخزين أي بيانات يريدونها دون أي تأثير على عمل العنصر نفسه.

استخدام السمات المخصصة:

يمكن استخدام السمات المخصصة بشكل مشابه للسمات القياسية في لغة HTML، ولكن بدلاً من استخدام اسم السمة القياسي، يجب عليك استخدام اسم السمة

المخصصة المتبوعة "-data"؛ مثل:

HTML

```
<div data-custom="value"></div>
```

ويمكنك أيضًا تحديد السمة المخصصة في الكائن DOM باستخدام الوظيفة `getAttribute` على النحو التالي:

javascript

```
const myDiv = document.querySelector('div');
const myAttributeValue = myDiv.getAttribute('data-custom');
console.log(myAttributeValue); // output = value
```

الفوائد الرئيسية للاستخدام السمات المخصصة:

1. توفير معلومات إضافية للمطورين حول العناصر في الصفحة.
2. يمكن استخدامها لتبسيط الأكواد وتجنب الحاجة إلى إضافة بيانات إضافية في الصفحة باستخدام بيانات خارجية مثل قاعدة البيانات وغيرها...

هناك العديد من الفوائد الأخرى عند تحديث الدرس سنذكرها كلها وهكذا إنتهينا تعرفنا على العناصر في لغة `html` نلتقي في دروس أخرى وفقكم الله.

ما هي العناصر الدلالية في لغة `html`؟

تعد العناصر النصية الدلالية Semantic Elements من الأدوات الرئيسية في لغة `HTML` حيث تساعد في تحسين هيكل المستند وترتيبه، وتحسين تجربة المستخدم وسهولة فهم المحتوى. تعتبر هذه العناصر مهمة للغاية في بناء صفحات الويب السليمة والمتوافقة مع معايير الويب.

العناصر الدلالية Semantic Elements توفر معلومات حول المحتوى في الصفحة وعلاقتها بالصفحة بأكملها. وتتيح هذه العناصر للمطورين إمكانية تحديد أهمية الأجزاء المختلفة في صفحة الويب. ومن أمثلة العناصر الدلالية Semantic Elements الشائعة في `HTML`:

1) عنصر `:header`

هو العنصر الذي يستخدم لتحديد المقطع العلوي من صفحة الويب، وعادةً ما يحتوي على شعار الموقع والقائمة الرئيسية والعنوان الرئيسي للصفحة. يستخدم كود العنصر **header** بالشكل التالي:

HTML

```
<header>
  <h1>عنوان الصفحة</h1>
  <nav>
    <ul>
      <li><a href="#">الصفحة الرئيسية</a></li>
      <li><a href="#">من نحن</a></li>
      <li><a href="#">اتصل بنا</a></li>
    </ul>
  </nav>
</header>
```

(2) عنصر **:nav**:

يستخدم هذا العنصر لتحديد القائمة الرئيسية في صفحة الويب، ويتم استخدامه داخل العنصر **header** بشكل شائع. يمكن استخدام عنصر **nav** بالشكل التالي:

HTML

```
<header>
  <nav>
    <ul>
      <li><a href="#">الصفحة الرئيسية</a></li>
      <li><a href="#">من نحن</a></li>
      <li><a href="#">اتصل بنا</a></li>
    </ul>
  </nav>
</header>
```

(3) عنصر **:section**:

يستخدم عنصر **section** لتقسيم المحتوى إلى أقسام مختلفة، وعادةً ما يتم استخدامه مع العناصر الداخلية مثل عنصر **article**. يمكن استخدام عنصر **section** بالشكل التالي:

```

<section>
    <h2>عنوان المقطع الأول</h2>
    <p>نص المقطع الأول</p>
</section>

<section>
    <h2>عنوان المقطع الثاني</h2>
    <p>نص المقطع الثاني</p>
</section>

<section>
    <h2>عنوان المقطع الثالث</h2>
    <p>نص المقطع الثالث</p>
</section>

```

4) عنصر **article**: يستخدم هذا العنصر لتحديد مقالة مستقلة في صفحة الويب، ويمكن استخدامه لتحديد المحتوى الرئيسي في صفحة الويب مثل الأخبار والمقالات والمدونات. يمكن استخدام عنصر **article** بالشكل التالي:

```

<article>
    <h2>عنوان المقالة</h2>
    <p>نص المقالة</p>
</article>

```

5) عنصر **footer**: هو العنصر الذي يستخدم لتحديد المقطع السفلي من صفحة الويب، وعادةً ما يحتوي على معلومات الاتصال وحقوق النشر وروابط الشبكات الاجتماعية. يستخدم العنصر **footer** بالشكل التالي:

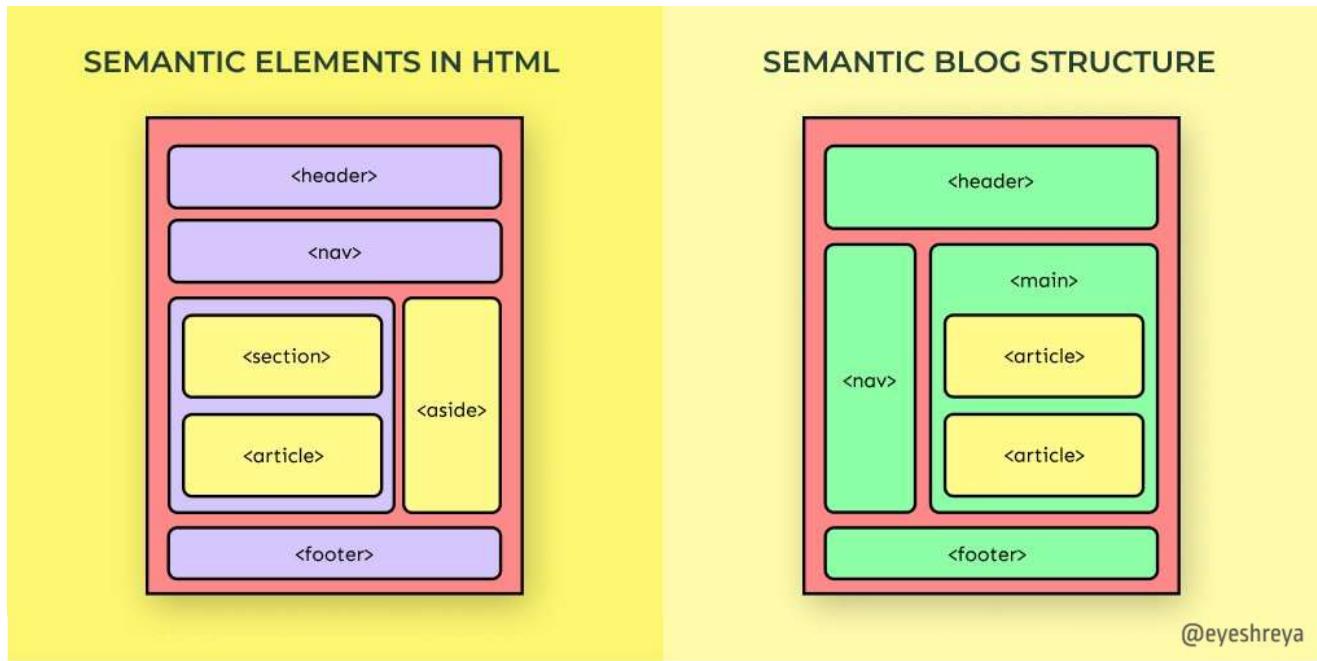
```

<footer>
    <p>جميع الحقوق محفوظة © 2023</p>
</footer>

```

وهكذا إنتهينا من هذا الدرس سنضيف المزيد من العناصر الدلالية في المستقبل
وهكذا مع جميع الدروس شكرًا لكم ووفقكم الله.

هذه صورة معبرة لكي تفهم أكثر:



ما هي العناصر غير الدلالية

تُعد لغة **HTML** أحد أهم لغات الويب، حيث تُستخدم لتحديد هيكل ومحفوظ صفحات الويب. وعند إنشاء صفحة ويب باستخدام **HTML** فإن هناك بعض العناصر التي لا تحمل أية دلالة على المحتوى الذي يتم عرضه، ولكن تكون لها أهمية كبيرة في تحديد تنسيق الصفحة وتحسين تجربة المستخدم. وتُعرف هذه العناصر باسم "العناصر الغير دلالية"، وفي هذه المقالة سنتحدث عن بعضها.

1) العنصر `:div`

يُستخدم العنصر `div` لتجميع العناصر المختلفة معاً داخل صفحة الويب، حيث يتم تطبيق نمط `CSS` عليها مباشرة دون أن يكون للعنصر `div` أي تأثير على المحتوى. ويمكن استخدام العنصر `div` لتقسيم الصفحة إلى مجموعات مختلفة من العناصر وتطبيق أنماط `CSS` مختلفة عليها مثال لكود على عنصر `div`:

HTML

```
<div>
  <p>Hello how are you</p>
```

</div>

2) العنصر :span

يُستخدم العنصر `span` لتحديد نص معين داخل صفحة الويب، حيث يمكن تطبيق نمط `CSS` على هذا النص مباشرة دون أن يؤثر العنصر `span` على المحتوى الذي يتم عرضه. ويمكن استخدام العنصر `span` لتطبيق أنماط مختلفة على جزء من النص داخل صفحة الويب مثل على كود لعنصر `span`:

HTML

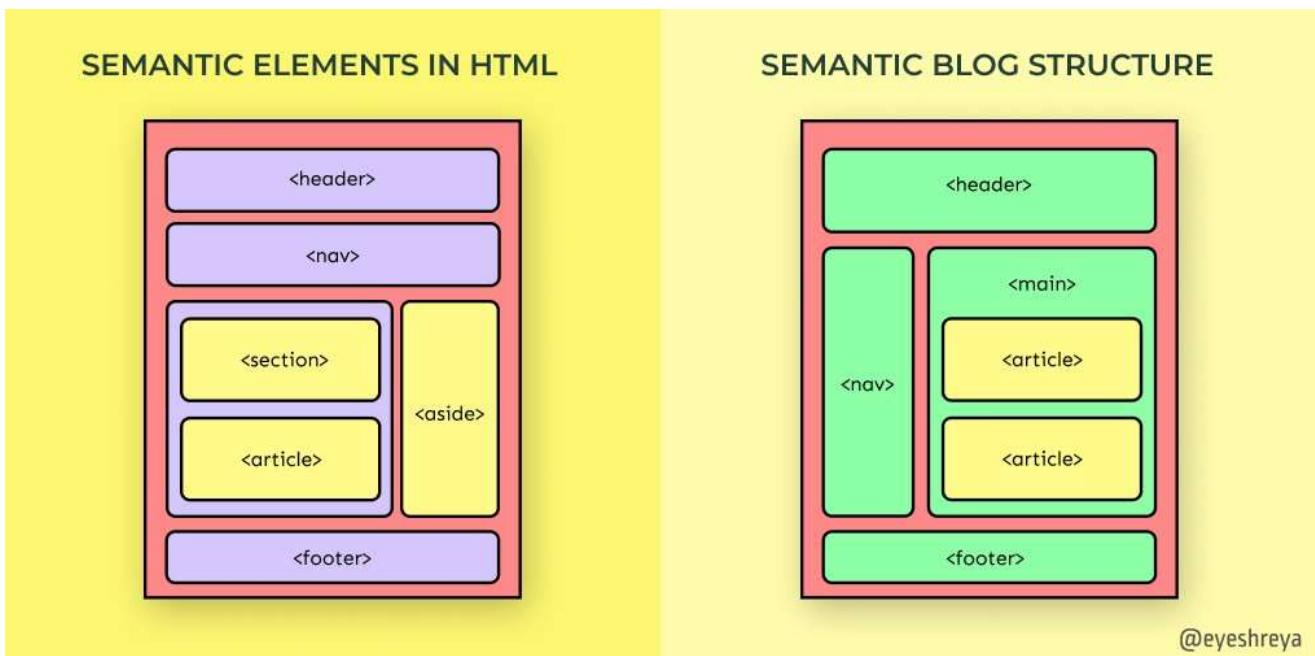
```
<span>this is span</span>
```

خلاصة عامة للدرس:

العناصر الغير دلالية في لغة HTML هي عناصر لا تحمل أي معنى دلالي محدد يمكن فهمه من خلال محتواها، ولكنها تستخدم لتحديد الشكل والتنسيق وتحسين تجربة المستخدم في صفحة الويب.

إنتهينا من درسنا المتواضع والبسيط وسنقوم بإضافة عناصر أخرى ستكتفي بعنصرتين شائعتين يمكنك البحث عن المزيد وشكرا على الدعم و وفقكم الله.

هذه صورة معبرة لكي تفهم أكثر:



@eyeshreya

شرح الوسم p

يعد وسم `p` واحداً من العناصر الأساسية في لغة `HTML` ويستخدم عادة لتحديد فقرات النص العادي في صفحة الويب. يتم استخدام وسم `p` لتجزئة النص إلى فقرات منفصلة، وكل فقرة تكون محاطة بوسم `p`. التنسيق الافتراضي لفقرة `p` يتضمن فراغاً فارغاً قبل وبعد الفقرة، ويتم تنسيق النص بشكل عمودي في الفقرة. يمكن إضافة خصائص إضافية لوسم `p` لتنسيق النص بشكل مخصص. فيما يلي مثال على كيفية استخدام وسم `p` في صفحة `HTML`:

HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>/<title>عنوان الصفحة
</head>
<body>
    <h1>/<h1>عنوان المقالة
    <p>نها . يمكننا استخدام وسم p لتحديد الفقرات النصية في صفحتنا.</p>
    <p>لنص، وسنستخدم خاصية style للتغيير لون النص إلى اللون الأحمر.</p>
    <p style="color: red;">.CSS
    وستتم تنسيقها بشكل مخصص باستخدام .CSS
</body>
</html>
```



في هذا المثال، تم استخدام وسم `p` لتحديد ثلاث فقرات منفصلة. يتم تنسيق الفقرات النصية الأولى والثانية بشكل افتراضي، في حين تم استخدام خاصية `style` للتغيير لون النص في الفقرة الثالثة إلى اللون الأحمر.

يمكن للمستخدمين استخدام وسم `p` بشكل كبير لتحديد وتجزئة النص في صفحات الويب، مما يجعلها أكثر وضوحاً وسهولة في القراءة. كما يمكن استخدام `CSS` والخصائص المخصصة الأخرى لتنسيق الفقرات.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح وسوم h1...h6

وسوم h1...h6 هي وسوم في لغة HTML تستخدم لتحديد عناوين الصفحة وفقراتها. تتيح هذه الوسوم تنسيق النص وتحسين تنظيم الصفحة وتوضيح المحتوى المقدم فيها.

ترتبط الوسوم h1...h6 بحيث يتم تحديد العناوين الرئيسية باستخدام الوسم h1 والعناوين الفرعية باستخدام الوسم h2، وهكذا. ويتم تنسيق النص في هذه الوسوم بشكل افتراضي على النحو التالي:

:h1

يتم تنسيق النص في وسم h1 بحجم الخط الكبير جداً.

:h2

يتم تنسيق النص في وسم h2 بحجم الخط الكبير.

:h3

يتم تنسيق النص في وسم h3 بحجم الخط المتوسط.

:h4

يتم تنسيق النص في وسم h4 بحجم الخط الصغير نسبياً.

:h5

يتم تنسيق النص في وسم h5 بحجم الخط الصغير جداً.

:h6

يتم تنسيق النص في وسم h6 بحجم الخط الصغير جداً.

فيما يلي بعض الأمثلة التوضيحية عن استخدام وسوم h1...h6:

HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>/<title><!--عنوان الصفحة-->
</head>
<body>
    <h1>/<h1><!--عنوان الرئيسي-->
</body>
</html>
```

لتحديد عناوين فرعية نكتب:

HTML

```
<h2></h2>عنوان فرعي رقم واحد
<h3></h3>عنوان فرعي رقم ثلاثة
<h4></h4>عنوان فرعي رقم أربعة
...

```

مثال لنتيجة الكود:

العنوان الرئيسي
العنوان الفرعي 1
العنوان الفرعي 2
العنوان الفرعي 3
العنوان الفرعي 4
العنوان الفرعي 5

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح **img** وسم

وسم **img** هو وسم في لغة HTML يستخدم لإدراج الصور والرسومات في صفحات الويب. ويتم استخدام وسم **img** بطريقة بسيطة وسهلة، حيث يتم تحديد عنوان URL للصورة المراد عرضها في الصفحة. يمكن استخدام وسم **img** كالتالي:

HTML

```
<!DOCTYPE html>
<html>
<head>
<title></title>عنوان الصفحة

```

```
</head>
<body>
    
</body>
</html>
```

كما ترون في سمة src قمنا بوضع رابط الصورة من الأنترنيت يمكنك عرض الصورة بطريقة محلية يعني من ملفات جهازك فقط تحدد إسم الصورة كما في المثال:

HTML

```

```

إذا كانت الصورة داخل مجلد نكتب هكذا:

HTML

```

```

و إذا كان ملفك داخل مجلد للخروج منه أو بعبارة أوضح للرجوع للخلف نستعمل نقطتين ومن ثم باك سلاش ونكتب إسم المجلد وباك سلاش أخرى ونكتب إسم الصورة مع الصيغة بالشكل الصحيح كما هو ظاهر معكم في الكود:

HTML

```

```

src: يتم استخدام هذا الخاصية لتحديد عنوان URL للصورة المراد إدراجها في الصفحة.

alt: يتم استخدام هذا الخاصية لإدراج نص بديل يتم عرضه بدلاً من الصورة إذا لم يتم تحميلها بشكل صحيح أو إذا كان المستخدم لا يمكنه رؤية الصورة. يجب أن يتم استخدام وسم img بشكل صحيح لضمان عرض الصورة بشكل صحيح في الصفحة، وذلك بتحديد عنوان URL صحيح للصورة وإضافة نص بديل مناسب للصورة.

لدينا سمات أخرى لتحديد عرض وطول الصور هنا نتيجة كود لصورة بدون استخدام

سمات width and height ستعرض بشكلها الطبيعي مثال لنتيجة الكود:



<AY/>

وهنا كود ونتيجة للإستخدام سمات width and height نمرر لهما قيمة رقمية مع وحدة ال px طبعا كل هذا سوف ندرسها بالشكل الصحيح في دروس ال CSS

HTML

```

<html>
<head>
    <title></title>عنوان الصفحة
</head>
<body>
    <a href="https://test.com"></a>رابط الموقع
</body>
</html>
```

يجب استبدال "https://test.com" بعنوان URL الذي تريد الارتباط به و "رابط الموقع" بالنص الذي تريد أن يظهر للمستخدمين كنص رابط. بالإضافة إلى الرابط الأساسي ، يمكنك إضافة العديد من الخصائص إلى وسم **a**. واحدة من هذه الخصائص هي **target** ، والتي تحدد ما إذا كان يتم فتح الرابط في نافذة جديدة أو نفس النافذة. على سبيل المثال:

<[a](https://test.com)

في هذا المثال ، تم إضافة "target=_blank" لتحديد فتح الرابط في نافذة جديدة. بالإضافة إلى ذلك ، يمكن استخدام وسم a لإنشاء روابط داخل صفحة ويب واحدة. يتم ذلك عن طريق استخدام العلامة "#" مع الرابط الفرعية. على سبيل المثال:

<[a](#section-1)

في هذا المثال ، يربط وسم a بجزء محدد من الصفحة المعين بواسطة العنصر المميز "section-1", حيث عن الطفظ على الرابط سيأخذك إلى العنصر الذي ال id هو "section-1" هذا هو المقصود بروابط داخلية.

يمكن استخدام وسم a في HTML لإنشاء روابط لتنزيل الملفات من الويب. عند النقر على الرابط ، يتم تنزيل الملف المحدد بدلاً من فتحه في صفحة جديدة، سنشرح كيفية استخدام وسم a لإنشاء رابط لتنزيل ملف.

أولاً ، يجب علينا تحديد عنوان URL للملف الذي نريد تزيله. يمكن أن يكون الملف على نفس الخادم الذي تستضيفه صفحتك أو على خادم آخر.

ثم يجب إنشاء وسم a وإضافة عنوان URL المحدد كقيمة لخاصية href. كما يجب إضافة خاصية "download" لتحديد أن الرابط هو رابط لتنزيل الملف.

على سبيل المثال، إذا كنت تريد إنشاء رابط لتنزيل ملف PDF يسمى "file.pdf" ، يمكنك استخدام الشفرة التالية:

<[ا](file.pdf) تنسيل الملف

في هذا المثال، تم إنشاء وسم a لرابط تنسيل ملف PDF يحمل اسم "file.pdf". يتم تحديد عنوان URL للملف في خاصية href، ويتم إضافة الخاصية "download" لتحديد أن الرابط هو رابط للتنزيل. يجب الانتباه إلى أن خاصية "download" لا تعمل على كل المتصفحات. بعض

المتصفحات القديمة قد لا تدعم هذه الخاصية. لذلك يمكن استخدام خيارات أخرى مثل استخدام JavaScript لتحقيق الغرض من تنزيل الملف أو استخدام نظام إدارة المحتوى لإنشاء روابط تنزيل.

يمكن استخدام وسم `a` في HTML لإنشاء روابط للتواصل مع أي شخص أو مؤسسة عبر الإنترنت. يمكن استخدامه لإرسال رسالة بالبريد الإلكتروني ، أو للاتصال بحساب وسائل التواصل الاجتماعي الخاص بشخص معين ، أو للاتصال بصفحة اتصال على موقع الويب.

لإنشاء رابط للتواصل ، يتم استخدام وسم `a` وتحديد نوع الاتصال الذي يتم الربط به في خاصية "href". على سبيل المثال ، إذا كنت تريد إنشاء رابط لإرسال بريد إلكتروني ، يمكنك استخدام الشفرة التالية:

HTML

```
<a href="mailto:test@gmail.com">إرسال رسالة إلى البريد الإلكتروني</a>
```

في هذا المثال ، تم استخدام الوسم `a` لإنشاء رابط لإرسال بريد إلكتروني. تم تحديد خاصية "href" وتحديد بروتوكول البريد الإلكتروني "mailto:" وعنوان البريد الإلكتروني "test@gmail.com". عند النقر على الرابط ، سيتم فتح البريد الإلكتروني الافتراضي لديك وتملئ حقل المرسل إليها تلقائياً بعنوان البريد الإلكتروني الذي تم تحديده في خاصية "href".

ويمكن أيضا استخدام وسم `a` لإنشاء روابط للاتصال بأرقام الهاتف وإرسال رسائل SMS.

لإنشاء رابط للاتصال برقم الهاتف المحمول ، يجب استخدام البروتوكول "tel" مع عنوان URL يحتوي على رقم الهاتف المحمول المطلوب. يمكن استخدام الشفرة التالية لإنشاء رابط للاتصال برقم الهاتف المحمول:

HTML

```
<a href="tel:+74829742874">اتصل بنا</a>
-- فقط رقم تقديري يمكنك وضع رقمك الخاص و التجربة --!
```

في هذا المثال ، تم استخدام الوسم `a` لإنشاء رابط للاتصال برقم الهاتف المحمول.

تم تحديد خاصية "href" وتحديد بروتوكول "tel" مع رقم الهاتف المحمول المطلوب. عند النقر على الرابط ، سيتم فتح تطبيق المكالمات على الهاتف المحمول وإجراء مكالمة تلقائية لرقم الهاتف المحمول المحدد.

يمكن أيضًا استخدام وسم a لإرسال رسالة SMS إلى رقم الهاتف المحمول المحدد. يجب استخدام البروتوكول "sms" مع عنوان URL يحتوي على رقم الهاتف المحمول المطلوب. يمكن استخدام الشفرة التالية لإنشاء رابط لإرسال رسالة SMS:

HTML

```
<a href="sms:+74829742874" إرسال رسالة> SMS</a>
-- فقط رقم تقديري يمكنك وضع رقمك الخاص و التجربة --!
```

باستخدام وسم a في HTML ، يمكن للمستخدمين الاتصال بأرقام الهاتف المحمول وإرسال رسائل SMS بشكل سريع وسهل عبر الويب. يمكن استخدام هذه الوظيفة بشكل خاص للشركات والأعمال التجارية التي ترغب في توفير وسيلة سريعة وسهلة للاتصال بالعملاء أو العملاء المحتملين. يمكن تخصيص روابط الهاتف المحمول ورسائل SMS لتضمين معلومات إضافية مثل الاسم ، العنوان ، الرسالة المطلوبة أو أي معلومات أخرى ذات صلة. يمكن ذلك عن طريق إضافة المعلومات المطلوبة إلى عنوان URL باستخدام المتغيرات والقيم. على سبيل المثال ، يمكن استخدام الشفرة التالية لإنشاء رابط للاتصال برقم الهاتف المحمول وتخصيصه بإضافة اسم الشخص المطلوب:

HTML

```
<a href="tel:+74829742874?name=Ayoub+Ayoau" إتصل بأيوب> a
-- فقط رقم تقديري يمكنك وضع رقمك الخاص و التجربة --!
```

في هذا المثال ، تم إضافة معلومات الاسم "Ayoub Ayoau" إلى عنوان URL باستخدام المتغير "name" والقيمة "Ayoub+Ayoau". عند النقر على الرابط ، سيتم فتح تطبيق المكالمات على الهاتف المحمول وإجراء مكالمة تلقائية لرقم الهاتف المحمول المحدد وسيظهر اسم الشخص المطلوب.

بالمثل ، يمكن إضافة المعلومات إلى رسالة SMS باستخدام المتغيرات والقيم. يمكن استخدام الشفرة التالية لإنشاء رابط لإرسال رسالة SMS مخصصة:

HTML

```
<a href="sms:+74829742874?body=Ayoub+Ayoau">إرسال رسالة إلى أيوب</a>
-- فقط رقم تقديري يمكنك وضع رقمك الخاص و التجربة --!
```

في هذا المثال ، تم إضافة معلومات الرسالة "Ayoub Ayoau" إلى عنوان URL باستخدام المتغير "body" والقيمة "Ayoub+Ayoau". عند النقر على الرابط ، سيتم فتح تطبيق الرسائل على الهاتف المحمول وتمهيد رسالة SMS جديدة وسيظهر الرسالة المحددة في مربع الرسالة.

يمكن أيضًا إضافة الأرقام المتعددة للهواتف المحمولة أو الرسائل SMS عن طريق فصل الأرقام بفواصل في عنوان URL. على سبيل المثال:

HTML

```
<a href="sms:+74829742874,+95747384838?body=Ayoub+Ayoau">عديد من الأرقام</a>
-- فقط رقم تقديري يمكنك وضع رقمك الخاص و التجربة --!
```



في هذا المثال ، تم إضافة رقمي الهاتف المحمول "95747384838" و "74829742874" بفواصل واحدة بينهما في عنوان URL. عند النقر على الرابط ، سيتم فتح تطبيق الرسائل على الهاتف المحمول وتمهيد رسالة SMS جديدة تحتوي على رقمي الهاتف المحمول المحدد والرسالة المحددة. في النهاية ، يجب التأكد من أن روابط الهاتف المحمول ورسائل SMS تعمل بشكل صحيح قبل نشرها. يمكن ذلك بتحميل صفحة الويب التي تحتوي على الروابط والتحقق من صحتها عن طريق النقر على الروابط والتحقق من أنها تقوم بفتح تطبيق المكالمات أو التطبيق الرسائل على الهاتف المحمول بشكل صحيح وإجراء المكالمة أو الرسالة المحددة بشكل صحيح.

بهذا تكون قد شرحنا كيفية استخدام وسم **a** في توليد روابط الهاتف المحمول ورسائل SMS وكيفية إضافة نص إضافي مع الرابط. هذا الوسم يتيح للمستخدمين سهولة الوصول إلى معلومات الاتصال الهامة مثل أرقام الهاتف المحمولة وإرسال الرسائل SMS بكل سهولة عن طريق النقر فوق الرابط المعنى. كما أنه يجعل صفحات الويب أكثر تفاعلاً وسهولة في الوصول إلى المعلومات المهمة التي يحتاجها الزوار.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح جميع أنواع القوائم

HTML هي لغة ترميز النصوص التي تستخدم لإنشاء صفحات الويب، وتحتوي على العديد من العناصر التي يمكن استخدامها لتنسيق المحتوى وعرضه بشكل جذاب للزوار. واحدة من هذه العناصر هي القوائم، التي تستخدم لتنظيم المحتوى في مجموعات مرتبة أو غير مرتبة. في هذه المقالة، سنلقي نظرة على أنواع القوائم المختلفة في HTML وكيفية استخدامها.

القائمة غير المرتبة (Unordered List)

تستخدم القوائم غير المرتبة لعرض عناصر القائمة بطريقة غير مرتبة، مع تعين نقطة أو رمز لكل عنصر في القائمة. لإنشاء قائمة غير مرتبة في HTML، يتم استخدام عنصر "ul"، ويتم إدراج كل عنصر في القائمة داخل عنصر "li". على سبيل المثال:

HTML

```
<ul>
  <li>عنصر 1</li>
  <li>عنصر 2</li>
  <li>عنصر 3</li>
</ul>
```

مثال لنتيجة الكود:

- عنصر 1
- عنصر 2
- عنصر 3

يمكن تخصيص نقاط العناصر في القائمة غير المرتبة باستخدام CSS، ويمكن أيضًا إضافة رموز أخرى بدلاً من النقاط الافتراضية.

القائمة المرتبة (Ordered List)

تستخدم القوائم المرتبة لعرض عناصر القائمة بطريقة مرتبة، ويتم تعريف رقم أو حرف لكل عنصر في القائمة. لإنشاء قائمة مرتبة في HTML، يتم استخدام عنصر "ol"، ويتم إدراج كل عنصر في القائمة داخل عنصر "li". على سبيل المثال:

HTML

```
<ol>
  <li>عنصر 1</li>
  <li>عنصر 2</li>
  <li>عنصر 3</li>
</ol>
```

مثال لنتيجة الكود:

- 1. عنصر 1
- 2. عنصر 2
- 3. عنصر 3

يمكن تخصيص رقم العناصر في القائمة المرتبة باستخدام CSS، ويمكن أيضًا تغيير نوع الحرف المستخدم لترقيم العناصر، مثل استخدام حروف بدلاً من الأرقام.

القائمة الفرعية (Nested List)

يمكن إنشاء قوائم فرعية داخل قوائم أخرى في HTML، وتسمى هذه القوائم الفرعية بالقوائم المتداخلة أو القائمة الشجرية. يتم إنشاء القوائم الفرعية باستخدام عنصر "ol" أو "ul" داخل عنصر "li" في القائمة الأم، على النحو التالي:

```
<ul>
  <li>عنصر 1</li>
  <li>عنصر 2</li>
  <li>
    <ul>
      <li>عنصر 3.1</li>
      <li>عنصر 3.2</li>
    </ul>
  </li>
</ul>
```

مثال لنتيجة الكود:

- عنصر 1
- عنصر 2
- عنصر 3.1
- عنصر 3.2

الخلاصة تستخدم القوائم في HTML لتنظيم المحتوى وجعله أكثر سهولة للقراءة والتصفح. يمكن استخدام القوائم غير المرتبة لعرض عناصر القائمة بطريقة غير مرتبة، ويمكن استخدام القوائم المرتبة لعرض العناصر بطريقة مرتبة. يمكن إنشاء قوائم فرعية داخل قوائم أخرى لإنشاء هيكل أكثر تعقيداً، ويمكن تخصيص تنسيق القوائم باستخدام CSS. باستخدام القوائم في HTML، يمكنك تحسين تنظيم وتصميم صفحات الويب الخاصة بك.

الآن سنتعرف على عنصر آخر خاص بالقوائم وهو عنصر القوائم "dl" في HTML هو أحد العناصر التي تستخدم لتنظيم المحتوى على صفحة الويب، ويستخدم عادةً لعرض البيانات والمعلومات بشكل منظم ومفهوم، سنتحدث عن عنصر القوائم "dl" بشكل أكثر تفصيلاً ونوضح كيفية استخدامه وتخصيصه.

مفهوم عنصر القوائم dl

يستخدم عنصر القوائم "dl" في HTML لإنشاء قائمة من العناصر، وكل عنصر

يتكون من زوجين من البيانات: العنصر والقيمة. يتم عرض العناصر بشكل رئيسي في الجزء العلوي من القائمة، بينما تظهر القيم في الجزء السفلي من القائمة. يمكن استخدام عنصر القوائم "dl" لعرض أي نوع من البيانات التي يمكن تنظيمها في شكل زوجي البيانات، مثل عرض النطق والتعرifات والمفاهيم والمصطلحات والأسماء والمفاتيح والقيم وغيرها.

بناء قائمة بعنصر القوائم dl

لبناء قائمة بعنصر القوائم "dl" في HTML، يجب عليك استخدام عناصر القائمة الفرعية "dt" و "dd". يجب وضع العنصر في عنصر "dt" والقيمة في عنصر "dd". يمكنك إضافة العناصر الجديدة إلى القائمة باستخدام عنصر "dt" و "dd" الجديدين. على سبيل المثال، يمكن إنشاء قائمة ببعض التعرifات والمفاهيم باستخدام عنصر القوائم "dl"، على النحو التالي:

HTML

```
<dl>
  <dt>تعريف 1</dt>
  <dd>مفهوم 1</dd>

  <dt>تعريف 2</dt>
  <dd>مفهوم 2</dd>
</dl>
```

مثال لنتيجة الكود:

تعريف 1
المفهوم 1
تعريف 2
المفهوم 2

الاستنتاج يعتبر عنصر القوائم "dl" في HTML وسيلة فعالة لتنظيم البيانات والمعلومات بشكل منظم ومناسب للقراء. باستخدام عناصر "dt" و "dd" في عنصر "dl"، يمكن ترتيب العناصر والقيم بطريقة مفهومة وسهلة الاستخدام. يمكن

تخصيص عنصر القوائم "d1" باستخدام CSS، مما يساعد في إضفاء الأسلوب المناسب للصفحة والمحتوى.

في لغة HTML، تُستخدم القوائم لعرض البيانات والمعلومات بطريقة منظمة ومفهومة. تتضمن القوائم ثلاثة أنواع رئيسية: القائمة المرتبة (ordered list)، والقائمة غير المرتبة (unordered list)، وقائمة الإيضاح (definition list). يمكن استخدام السمة type لتعديل شكل وترتيب العناصر في القوائم. في هذه المقالة، سنتحدث عن السمة type في القوائم وسنوضح كيفية استخدامها مع الأمثلة.

القائمة المرتبة والسمة type

تُستخدم القائمة المرتبة في HTML لترتيب البيانات بطريقة تبعاً للترتيب الرقمي، حيث يتم ترتيب العناصر بشكل تصاعدي بحسب الرقم الذي تحمله كل عنصر. وتُستخدم السمة type في القوائم المرتبة لتحديد النوع الذي يتم عرض الأرقام به.

النوع الافتراضي ("type="1")

النوع الافتراضي للقائمة المرتبة هو ترتيب تصاعدي للأرقام، حيث يبدأ الترتيب بالرقم 1 ويستمر حتى آخر عنصر في القائمة. يتم تحديد هذا النوع بواسطة السمة type، والتي يمكن إضافتها داخل عنصر القائمة المرتبة كما يلي:

HTML

```
<ol type="1">
  <li>عنصر 1</li>
  <li>عنصر 2</li>
  <li>عنصر 3</li>
</ol>
```

مثال لنتيجة الكود:

1. عنصر 1
2. عنصر 2

ترتيب الحروف ("type="a", type="A", type="i", type="l")

يمكن استخدام السمة type في القائمة المرتبة لترتيب العناصر باستخدام الحروف بدلاً من الأرقام. توفر HTML أربعة أنواع مختلفة لترتيب الحروف في القوائم المرتبة، وهي:

ترتيب الحروف الصغيرة ("type="a)

في هذا النوع، يتم ترتيب العناصر باستخدام الحروف الصغيرة، حيث يبدأ الترتيب بالحرف "a" ويستمر حتى آخر عنصر في القائمة. يمكن إضافة هذا النوع بواسطة السمة type كالتالي:

HTML

```
<ol type="a">
  <li>عنصر 1</li>
  <li>عنصر 2</li>
  <li>عنصر 3</li>
</ol>
```

مثال لنتيجة الكود:

- 1. عنصر a
- 2. عنصر b
- 3. عنصر c

ترتيب الحروف الكبيرة ("type="A)

في هذا النوع، يتم ترتيب العناصر باستخدام الحروف الكبيرة، حيث يبدأ الترتيب بالحرف "A" ويستمر حتى آخر عنصر في القائمة. يمكن إضافة هذا النوع بواسطة السمة type كالتالي:

HTML

```
<ol type="A">
  <li>عنصر 1</li>
  <li>عنصر 2</li>
  <li>عنصر 3</li>
</ol>
```

مثال لنتيجة الكود:

A. عنصر 1
B. عنصر 2
C. عنصر 3

ترتيب الأرقام الرومانية الصغيرة ("type="i")

في هذا النوع، يتم ترتيب العناصر باستخدام الأرقام الرومانية الصغيرة، حيث يبدأ الترتيب بالرقم "i" ويستمر حتى آخر عنصر في القائمة. يمكن إضافة هذا النوع بواسطة السمة type كالتالي:

HTML

```
<ol type="i">
  <li>عنصر 1</li>
  <li>عنصر 2</li>
  <li>عنصر 3</li>
</ol>
```

مثال لنتيجة الكود:

- i. عنصر 1
- ii. عنصر 2
- iii. عنصر 3

ترتيب الأرقام الرومانية الكبيرة ("type="I")

في هذا النوع، يتم ترتيب العناصر باستخدام الأرقام الرومانية الكبيرة، حيث يبدأ الترتيب بالرقم "I" ويستمر حتى آخر عنصر في القائمة. يمكن إضافة هذا النوع بواسطة السمة type كال التالي:

HTML

```
<ol type="A">
  <li>عنصر 1</li>
  <li>عنصر 2</li>
  <li>عنصر 3</li>
</ol>
```

مثال لنتيجة الكود:

- A. عنصر 1
- B. عنصر 2
- C. عنصر 3

ترتيب الأرقام الرومانية الصغيرة ("type="i")

في هذا النوع، يتم ترتيب العناصر باستخدام الأرقام الرومانية الصغيرة، حيث يبدأ الترتيب بالرقم "i" ويستمر حتى آخر عنصر في القائمة. يمكن إضافة هذا النوع بواسطة السمة type كال التالي:

HTML

```
<ol type="I">
<li>عنصر 1</li>
<li>عنصر 2</li>
<li>عنصر 3</li>
</ol>
```

مثال لنتيجة الكود:

1. عنصر 1
 2. عنصر 2
 3. عنصر 3

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة يوجد العديد من الأنواع الأخرى تعرفنا في هذا الدرس على أهمها وهي `type=1`, `type=a`, `type=A`, `type=i`, `type=I`. يمكنك البحث عن المزيد و وفقك الله.

table وسم

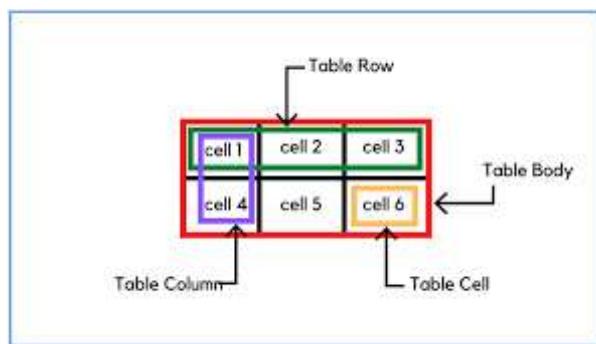
تستخدم لغة **HTML** (لغة وصف النصوص الفائقية) على نطاق واسع لإنشاء صفحات الويب، وهي لغة تتضمن العديد من العناصر والعلامات المختلفة التي تساعد على تنسيق وتنظيم المحتوى على الصفحة. من بين هذه العناصر، يعد الجدول أحد العناصر الأكثر شيوعاً والأكثر استخداماً في صفحات الويب.

عند إنشاء جدول في لغة **HTML**، يتم استخدام عنصر `"table"` الذي يتكون من صفات واحد أو أكثر من الصفوف وعدة أعمدة. وتشتمل كل صف على عنصر `"tr"`، في حين يتضمن كل عمود عنصر `"td"`. يمكنك إضافة محتوى إلى الجدول عن طريق استخدام هذه العناصر.

فيما يلي مثال على كيفية إنشاء جدول في **HTML**:

```
<table>
  <tr>
    <td>الصف الأول، العمود الأول</td>
    <td>الصف الأول، العمود الثاني</td>
  </tr>
  <tr>
    <td>الصف الثاني، العمود الأول</td>
    <td>الصف الثاني، العمود الثاني</td>
  </tr>
</table>
```

صورة ستساعدك في فهم الجداول في لغة HTML:



هذا المثال يتضمن جدولًا بسيطًا يتتألف من صفين وعمودين. يتم تعريف الصف الأول باستخدام عنصر "tr"، ويتم تعريف العمود الأول والثاني في هذا الصف باستخدام عنصر "td". يتم تكرار هذا النمط لإنشاء الصف الثاني.

بالإضافة إلى ذلك، يمكن تخصيص الجدول باستخدام مختلف الخصائص مثل اللون والحدود وعرض المحتوى في الخلايا وتحديد أحجامها. يمكن تحديد هذه الخصائص باستخدام العناصر المختلفة في HTML، ومن بين هذه العناصر:

:<th>

تستخدم لتحديد خلية العنوان في الجدول، وتكون عادةً مركزة تلقائيًا وعرضها عادةً بالخط العريض.

:<caption>

تستخدم لإضافة عنوان للجدول.

:rowspan و colspan
تستخدم لدمج الخلايا عبر عدة صفوف أو أعمدة.

:border
تستخدم لتحديد سماكة حدود الجدول.

:bgcolor
تستخدم لتحديد لون خلفية الجدول.

:height و width
تستخدم لتحديد أحجام الخلايا في الجدول.

وفيما يلي مثال على كيفية تخصيص جدول في HTML باستخدام بعض هذه الخصائص:

HTML

```
<table>
  <caption>جدول المبيعات الشهرية</caption>
  <tr>
    <td>الشهر</td>
    <td colspan="2">المبيعات</td>
  </tr>
  <tr>
    <td rowspan="3">ينايير</td>
    <td>1 المنتج</td>
    <td>$50</td>
  </tr>
  <tr>
    <td>2 المنتج</td>
    <td>$150</td>
  </tr>
  <tr>
    <td rowspan="2">فبراير</td>
    <td>1 المنتج</td>
    <td>$150</td>
  </tr>
  <tr>
```

```
<td> المنتج 2 </td>
<td>$200</td>
</tr>
</table>
```

يمكنك تجربة الكود والتحقق من الجدول. يتضمن هذا المثال جدولًا مخصصًا يحتوي على عدة خصائص مختلفة. يتم استخدام عنصر "border" لتحديد سماكة حدود الجدول، ويتم استخدام "bgcolor" لتحديد لون خلفية الجدول. يتم استخدام "caption" لإضافة عنوان للجدول، ويتم استخدام "th" لتحديد خلية العنوان في الجدول، ويتم استخدام "colspan" لدمج خلية في عدة أعمدة، كما يتم استخدام "rowspan" لدمج خلية في عدة صفوف. يتم استخدام "height" و "width" لتحديد أحجام الخلايا في الجدول.

يجب ملاحظة أن عناصر HTML الأخرى يمكن استخدامها داخل الجدول لتخصيص محتواه، مثل الصور والروابط والنصوص. يمكن أيضًا استخدام CSS لتخفيض جدول HTML بشكل أكبر.

في النهاية، يمكن استخدام جداول HTML لتنظيم البيانات والمحتوى بشكل منظم وجميل، ويمكن تخصيصها باستخدام عدة خصائص مختلفة. يجب مراعاة استخدامها بشكل ملائم وفي الأماكن المناسبة في تصميم صفحات الويب. وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح عناصر block

في HTML، يمكن تقسيم العناصر إلى نوعين رئيسيين: العناصر المنشقة (block) والعناصر الدالة (inline elements). يتميز العناصر المنشقة بأنها تشغّل مساحة كاملة في العرض، وتبدأ عادةً من سطر جديد. في هذا الدرس، سنتحدث عن العناصر المنشقة وسنقدم بعض الأمثلة.

1) الفقرة: p

عنصر الفقرة يستخدم لعرض فقرة نصية مستقلة في الصفحة. يبدأ هذا العنصر عادةً من سطر جديد وينتهي بترك فراغ فارغ قبل بدء العنصر التالي. يمكن استخدام التنسيقات CSS لتحديد مظهر هذا العنصر بشكل دقيق، على سبيل المثال:

HTML

```
<p>This is a paragraph</p>
```

2- العنوان :**heading**

تستخدم عناصر العنوان **heading** لعرض عنوان رئيسي للصفحة أو قسم من الصفحة. يوجد في **HTML** ستة عناصر رئيسية للعناوين (**h1, h2, h3, h4, h5, h6**)، حيث يتم استخدام **h1** للعناوين الرئيسية وتنقص الحجم تباعاً للعناوين الفرعية. يمكن استخدام هذه العناصر بالترتيب المناسب لتحديد التسلسل الهرمي للعناوين، على النحو التالي:

HTML

```
<h1>This is a main heading</h1>
<h2>This is a subheading</h2>
<h3>This is a sub-subheading</h3>
...
...
```

3) القائمة :**list**

كما يمكن استخدام العنصر الداخلي (**list item**) **li** داخل العنصر **ul** أو **ol** لتحديد عنصر محدد داخل القائمة. يمكن أيضاً استخدام العنصر (**definition list**) **dl** (**definition term**) **dt** و لعرض قائمة بالتعريفات والأوصاف، واستخدام العناصر (**definition description**) **dd** داخل العنصر **dl** لتحديد التعريفات والأوصاف، على النحو التالي:

HTML

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

<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>

<dl>
  <dt>Term 1</dt>
  <dd>Description 1</dd>
```

```
<dt>Term 2</dt>
<dd>Description 2</dd>
</dl>
```

4- العنصر :`div`

تستخدم عناصر الفصل (**division**) لتقسيم صفحة الويب إلى أجزاء مختلفة، لأغراض التنسيق أو العرض. يمكن استخدام العنصر `div` لإنشاء علامة عمومية للتقسيم، دون تحديد أي نوع محدد من العناصر المنشقة، على النحو التالي:

HTML

```
<div>
  <p>This is a paragraph inside a div</p>
</div>
```

هذه بعض الأمثلة الشائعة للعناصر الـ **block** في [HTML](#). يمكن استخدام هذه العناصر بالترتيب المناسب وباستخدام تنسيدات [CSS](#) الملائمة لتحسين تصميم وتنسيق الصفحات الإلكترونية.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح عناصر `inline`

تُستخدم العناصر المضمنة (**Inline Elements**) في تنسيق صفحات الويب لتحديد جزء صغير من النص أو العنصر الذي يمكن وضعه داخل سطر نصي. ويمكن استخدام العناصر المضمنة لتحديد عنصر مثل رابط النص أو الصورة أو عنصر الفيديو وغيرها. ومن العناصر **Inline** الشائعة في [HTML](#) العناصر التالية:

(1) الرابط (anchor)
يستخدم عنصر الرابط `a` لإنشاء روابط بين صفحات الويب وصفحات الواقع الأخرى. وعند استخدام هذا العنصر، يجب تحديد الرابط الذي يربط إليه النص باستخدام الخاصية `href`، على النحو التالي:

HTML

```
<a href="https://test.com">زيارة الموقع</a>
```

2) الصورة :

يستخدم عنصر الصورة `img` لإدراج صور في صفحات الويب، ويتم تحديد مصدر الصورة بإستخدام الخاصية `src`. ومن الممكن أيضاً إضافة وصف للصورة باستخدام الخاصية `alt`، على النحو التالي:

HTML

```

```

3) النص `span`: يستخدم عنصر النص `span` لتحديد جزء من النص لإضفاء تنسيقات معينة، ولكن دون تغيير النص الأصلي. ويتم ذلك عن طريق إضافة الخاصية `style`, التي يمكن استخدامها لتغيير لون النص أو خط النص أو الحجم، على النحو التالي:

HTML

```
<p><span style="color:red">جزء من النص بلون أحمر</span></p> هذا
```

هذه بعض الأمثلة للاستخدامات الشائعة للعناصر المضمنة في [HTML](#). يمكن استخدام هذه العناصر بأكثر من طريقة، والتي تعتمد على احتياجات التصميم والمتطلبات التي تتعلق بالتنسيقات المطلوبة في الصفحة، كما يمكن استخدام تنسيقات [CSS](#) لتحديد التنسيقات الإضافية لهذه العناصر.

في النهاية، تمثل العناصر المضمنة عناصر مهمة في [HTML](#) وتستخدم بشكل واسع في تنسيق صفحات الويب وجعلها أكثر احترافية وجاذبية. ويجب أن تستخدم هذه العناصر بحكمة وفي الأماكن المناسبة لتحسين تجربة المستخدم وتسهيل التنقل في المحتوى.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح السمة id

سمة **id** في HTML تستخدم لتعيين معرف فريد لعنصر HTML المحدد. وهي تسمح للمطورين بتحديد العناصر والوصول إليها بسهولة باستخدام CSS أو الجافا سكريبت. يتم تعريف السمة **id** بوضع كلمة **id** في عنصر HTML ومن ثم تعيين القيمة المميزة لهذا العنصر بعد كلمة **id**. ويجب أن تكون القيمة المعينة فريدة في الصفحة، ولا يجوز تكرارها على عناصر HTML أخرى في نفس الصفحة. يمكن أن تتكون القيمة المعينة لـ **id** من أي مجموعة من الأحرف والأرقام والرموز، ويفضل تسمية السمة بشيء يمكن فهمه بسهولة.

لتحديد عنصر HTML باستخدام السمة **id** يمكن استخدام رمز الهاش (#) مع القيمة المعينة لـ **id**. على سبيل المثال، إذا كان هناك div يحمل السمة "main-div" يمكن استخدام CSS لتطبيق التنسيق عليها عن طريق الكتابة كما يلي:

HTML

```
<div id="main-div">
    this is div
</div>
```

الآن سنستدعي نفس القيمة الموجودة في السمة **id** وقبلها علامة الهاش (#) في لغة CSS:

CSS

```
#main-div {
    width: 500px;
    height: 300px;
    border: 1px solid black;
}
```

ويمكن استخدام السمة **id** أيضاً في الجافا سكريبت للوصول إلى العناصر HTML المحددة وتغييرها بسهولة. على سبيل المثال، يمكن استخدام الكود التالي في

الجافا سكريبت لتغيير النص الظاهر على العنصر `div` الذي يحمل السمة `id` `: "main-div"`

JavaScript

```
document.getElementById("main-div").innerHTML = "Change text!";
```

ويعتبر استخدام السمة `id` أحد الطرق الرئيسية لتحديد العناصر `HTML` في صفحات الويب، وتعتبر واحدة من الأدوات الأساسية لتنسيق وتحكم في التصميم والسلوك للعناصر المحددة.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح السمة `class`

سمة `class` هي إحدى سمات عناصر `HTML` التي تستخدم لتحديد مجموعة من العناصر التي يتم تنسيقها بنفس الطريقة. يمكن استخدام السمة `class` للوصول إلى العناصر وتنسيقها باستخدام `CSS`، وكذلك لتطبيق سلوك خاص على العناصر باستخدام الجافا سكريبت.

يتم تعريف السمة `class` بوضع الكلمة "`class`" في عنصر `HTML` وتعيين القيمة المناسبة لها. يمكن أن تكون القيمة المعينة للسمة `class` من أي مجموعة من الأحرف والأرقام والرموز، ويفضل تسمية السمة بشيء يمكن فهمه بسهولة. يمكن استخدام السمة `class` لتطبيق تنسيق محدد على مجموعة من العناصر `HTML`. ولتحديد عنصر `HTML` باستخدام السمة `class` يمكن استخدام النقطة (.) مع القيمة المعينة للسمة `class`. على سبيل المثال، إذا كان هناك مجموعة من الفقرات تحمل السمة "paragraph" `class`، يمكن استخدام `CSS` لتطبيق التنسيق عليها عن طريق الكتابة كما يلي:

HTML

```
<p class="paragraph">this is paragraph number 1</p>
<p class="paragraph">this is paragraph number 2</p>
<p class="paragraph">this is paragraph number 3</p>
```

الآن سنستدعي نفس القيمة الموجودة في السمة `class` وقبلها علامة الهاش `(.)` في لغة `CSS`

`CSS`

```
.paragraph {
    font-size: 16px;
    color: #333;
    line-height: 1.5;
}
```

ويمكن استخدام السمة `class` أيضًا لتطبيق سلوك محدد على العناصر HTML باستخدام الجافا سكريبت. على سبيل المثال، يمكن استخدام الكود التالي في الجافا سكريبت لتحويل الصفحة إلى وضع الليلي عند الضغط على زر يحمل السمة `:class "toggle"`

`JavaScript`

```
document.querySelector(".toggle").addEventListener("click", function() {
    document.body.classList.toggle("night-mode");
});
```

ويعتبر استخدام السمة `class` أحد الطرق الرئيسية لتحديد مجموعة من العناصر HTML في صفحات الويب، وتعتبر واحدة من الأدوات الأساسية لتنسيق وتحكم في التصميم والتنظيم العناصر في صفحات الويب. علاوة على ذلك، يمكن استخدام السمة `class` لتعيين أكثر من قيمة واحدة لنفس العنصر HTML، ويتم فصل كل قيمة بمسافة. على سبيل المثال، يمكن تعيين السمة `class` بالشكل التالي لعنصر HTML واحد:

`HTML`

```
<div class="box red-border large-text">content</div>
```

ويمكن استخدام `CSS` لتطبيق التنسيق المحدد على العنصر المعين بالشكل التالي:

```

.box {
    padding: 10px;
}

.red-border {
    border: 1px solid red;
}

.large-text {
    font-size: 20px;
}

```

بهذه الطريقة، يتم تطبيق تنسيق الحاوية العامة `padding` على جميع العناصر التي تحمل السمة "box" `class`، وتطبيق تنسيق الحدود الحمراء على العناصر التي تحمل السمة "red-border" `class`، وتطبيق حجم الخط الكبير على العناصر التي تحمل السمة "large-text" `class`.

الخلاصة: تعد السمة `class` أداة مهمة لتنظيم العناصر في صفحات الويب وتحديد تنسيقات مختلفة لمجموعات من العناصر. ويمكن استخدامها بالتزامن مع `CSS` والجافا سكريبت لتحقيق التنسيق والسلوك المناسبين للعناصر الهامة في صفحات الويب.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح ترويسة `head`

ترويسة "`head`" هي عبارة عن عنصر `HTML` يتم استخدامه في بداية صفحة ويتضمن عدداً من العناصر المهمة التي تحدد محتوى الصفحة وترتبط بالعديد من الأمور الأساسية المتعلقة بالتنسيق والتصميم وتحسين محركات البحث. إليك بعض العناصر التي يمكن تضمينها في ترويسة "`head`" مع الأمثلة:

1) العنوان: تُستخدم عناصر العنوان `title` لتحديد عنوان الصفحة وظهوره في

شريط عنوان المستعرض. يجب أن يحتوي على وصف مختصر ودقيق لمحفوظة الصفحة وأن يحتوي على الكلمات الرئيسية المهمة لتحسين محركات البحث.
مثال:

HTML

```
<head>
  <title>/عنوان الصفحة</title>
</head>
```

2) رابط الأيقونة: تُستخدم روابط الأيقونات **favicon** لتحديد الصورة الصغيرة التي تظهر عادةً في علامة التبويب الخاصة بالصفحة وفي شريط عنوان المستعرض.
مثال:

HTML

```
<head>
  <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
</head>
```

3) الوصف : يستخدم وصف الصفحة **meta description** لتحديد وصف قصير لمحتوى الصفحة ويتم استخدامه في نتائج محركات البحث. يجب أن يكون وصف الصفحة واضحاً ومختصراً ويحتوي على الكلمات الرئيسية المهمة.
مثال:

HTML

```
<head>
  <meta name="description" content="وصف مختصر لمحتوى الصفحة">
</head>
```

4) التنسيقات والأنميات: يمكن تضمين أكواد CSS و JS في ترويسة "head" لتنسيق وتصميم الموقع وإضافة المزيد من الوظائف الديناميكية.
مثال:

HTML

```
<head>
  <link rel="stylesheet" href="styles.css">
  <script src="main.js"></script>
</head>
```

5) كلمات البحث: تستخدم عناصر الكلمات الرئيسية `meta keywords` لتحديد الكلمات الرئيسية المتعلقة بمحفوظة الصفحة والتي يمكن استخدامها لتحسين محركات البحث. ومع ذلك، فإن هذه الميزة قد تكون أقل أهمية الآن لأن محركات البحث تستخدم الآن أساليب أكثر تعقيداً لتحديد المحتوى المهم على الصفحة.

مثال:

HTML

```
<head>
  <meta name="keywords" content="3 الكلمة الرئيسية 2, الكلمة الرئيسية 1">
</head>
```



6) تحديد اللغة `meta language`: يستخدم هذا العنصر لتحديد لغة الصفحة وهو مهم لمحركات البحث ومستخدمي الإنترنت من أجل عرض الصفحة باللغة المناسبة.

مثال:

HTML

```
<head>
  <meta http-equiv="Content-Language" content="ar">
</head>
```

7) تحديد الترميز `meta charset`: يستخدم هذا العنصر لتحديد ترميز الحروف المستخدم في الصفحة.

مثال:

HTML

```
<head>
```

```
<meta charset="UTF-8">
</head>
```

8) تحديد المؤلف: يستخدم هذا العنصر لتحديد اسم المؤلف للصفحة.
مثال:

HTML

```
<head>
  <meta name="author" content="اسم المؤلف هنا">
</head>
```

9) تحديد الروابط الأساسية: يستخدم هذا العنصر لتحديد الرابط الأساسي لصفحة الويب والذي يستخدم عند تحديد الروابط النسبية.
مثال:

HTML

```
<head>
  <base href="https://test.com/">
</head>
```

10) تحديد تحديث الصفحة: يستخدم هذا العنصر لتحديث الصفحة تلقائياً بعد فترة معينة.
مثال:

HTML

```
<head>
  <meta http-equiv="refresh" content="5; url=https://test.com/">
</head>
```

في هذا المثال، ستم إعادة تحميل الصفحة بعد 5 ثوانٍ وستتم إعادة التوجيه إلى الرابط المحدد <https://test.com>.

11) تحديد التعريفات: يستخدم هذا العنصر لتحديد تخطيط العرض

الأولي للصفحة، مثل حجم الشاشة والمقياس والتدوير.
مثال:

HTML

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```



في هذا المثال، يتم تحديد تخطيط العرض الأولي للصفحة ليتم عرضه على شاشات الأجهزة المحمولة بالحجم الصحيح وبمقياس النسبة المناسب.

12) تحديد الكاتب **meta publisher**: يستخدم هذا العنصر لتحديد اسم الناشر للصفحة.

مثال:

HTML

```
<head>
  <meta name="publisher" content="اسم الناشر هنا">
</head>
```

13) تحديد صلاحية الحقوق **meta copyright**: يستخدم هذا العنصر لتحديد صلاحية حقوق النشر للصفحة.

مثال:

HTML

```
<head>
  <meta name="copyright" content="Copyright ©">
</head>
```

تحديد وضع الصفحة **:color-scheme**

"color-scheme" هو خاصية تستخدم في تحديد نظام الألوان المفضل لمستخدم النهار أو الليل. هذه لتوفير تجربة محسنة للمستخدمين على جميع الأجهزة ، سواء كانوا يستخدمون الهواتف الشخصية أو الهواتف المحمولة أو الأجهزة اللوحية.

يتم تعريف قيمة "color-scheme" على الجسم (body) أو أي عنصر آخر للصفحة. يمكن أن تكون القيمة إما "light" أو "dark" أو "no-preference". تقوم هذه الخاصية بتغيير ألوان الخلفية والنص والأطراف وأيضاً أي لون يتم استخدامه في صفحة الويب لتتناسب مع نظام الألوان المفضل للمستخدم. في حالة استخدام "no-preference" كقيمة لـ "color-scheme" ، سيظل النظام الافتراضي للألوان مفعلاً وسيتم تطبيق الألوان المعتادة.

مثال:

HTML

```
<!DOCTYPE html>
<html>
<head>
<meta name="color-scheme" content="light dark">
<style>
body {
background-color: white;
color: black;
}
@media (prefers-color-scheme: dark) {
body {
background-color: black;
color: white;
}
}
</style>
</head>
<body>
<h1>Hello World</h1>
<p>This is p</p>
</body>
</html>
```

في هذا المثال ، يتم تعريف "color-scheme" إلى "light dark" للسماح للمستخدمين بالتبديل بين نظام الألوان الفاتح والداكن. يتم تطبيق الألوان المناسبة باستخدام قاعدة CSS العادية والميديا استعلامات (media queries) بتفضيل المستخدم للألوان الفاتحة أو الداكنة.

"theme-color" هي خاصية تستخدم في تحديد لون خاص لشريط العنوان

والأيقونة في الهاتف المحمول والجهاز اللوحي الذي يتم استخدامه لإضافة صفحة الويب إلى الشاشة الرئيسية. يتم تعريف قيمة "theme-color" في عنصر "meta" في "head" من صفحة HTML.

تحديد "theme-color" للمستخدمين يمكنهم الحصول على تجربة محسنة للواجهة الرسومية لصفحة الويب على الهواتف المحمولة والأجهزة اللوحية. على سبيل المثال ، إذا كان لون شريط العنوان هو الأزرق ، فيمكن تعريف "theme-color" للأزرق لتحسين توافق الألوان لصفحة الويب على الهواتف المحمولة والأجهزة اللوحية.

مثال:

HTML

```
<!DOCTYPE html>
<html>
<head>
    <meta name="theme-color" content="#f45">
</head>
<body>
    <h1>Hello World</h1>
    <p>This is p</p>
</body>
</html>
```

في هذا المثال ، يتم تعريف "theme-color" إلى اللون الأحمر الفاتح (#f45) لتغيير لون شريط العنوان والأيقونة الموجودة في الهاتف المحمول. يجب على المستخدمين الذين يستخدمون الهواتف المحمولة والأجهزة اللوحية التي تستخدم نظام Android أن يكون لديهم نسخة من Chrome مثبتة على جهازهم لتحديد "theme-color" بشكل صحيح. ومن الجدير بالذكر أنه يجب أن تستخدم قيمة اللون في تنسيق HEX الخاص بها (RRGGBB#).

يمكنك التحكم في كيفية عرض موقعك على منصات الشبكات الاجتماعية مثل Facebook و Twitter باستخدام بعض الأوامر والعناصر الخاصة. هذه العناصر تسمى "ميتا تاج" (Meta Tags) وتمكن المطورين من تحديد العناوين والوصف والصور والألوان الرئيسية التي تظهر عند مشاركة رابط الموقع على منصات الشبكات الاجتماعية.

فيما يلي بعض الميتا تاج الشائعة التي يمكن استخدامها للتحكم في عرض موقعك

على Facebook و Twitter :

1) عنوان الصفحة **Title**: يتم استخدام هذا الميتا تاج لتحديد العنوان الذي سيظهر في عنوان الصفحة على Facebook و Twitter عند مشاركة الرابط. مثال:

HTML

```
<-- facbook -->
<head>
  <meta property="og:title" content="عنوان صفحتك">
</head>
```

HTML

```
<-- twitter -->
<head>
  <meta name="twitter:title" content="عنوان صفحتك">
</head>
```

2) وصف الصفحة **Description**: يتم استخدام هذا الميتا تاج لتحديد الوصف الذي سيظهر تحت العنوان على Facebook و Twitter عند مشاركة الرابط. مثال:

HTML

```
<-- facebook -->
<head>
  <meta property="og:description" content="وصف صفحتك">
</head>
```

HTML

```
<-- twitter -->
<head>
  <meta name="twitter:description" content="وصف صفحتك">
</head>
```

(3) صورة معينة **Image**: يتم استخدام هذا الميتا تاج لتحديد الصورة التي سيتم عرضها عند مشاركة الرابط على Facebook و Twitter. مثال:

HTML

```
<-- facebook -->
<head>
    <meta property="og:image" content="رابط الصورة">
</head>
```

HTML

```
<-- twitter -->
<head>
    <meta name="twitter:image" content="رابط الصورة">
</head>
```

(4) عرض الموقع بالكامل **Card**: يتم استخدام هذا الميتا تاج لتحديد طريقة عرض الموقع على Twitter. يمكن تعين هذا الميتا تاج إلى "summary" لعرض العنوان والوصف والصورة أو "summary_large_image" لعرض الصورة بحجم كبير. مثال:

HTML

```
<-- twitter -->
<head>
    <meta name="twitter:card" content="summary">
</head>
```

HTML

```
<-- twitter -->
<head>
    <meta name="twitter:card" content="summary_large_image">
</head>
```

تحديد هذه الميّة تاج بشكل صحيح ودقيق يتطلّب القليل من الوقت والجهد ولكنه يستحق العناء حيث يمكن أن يزيد من جاذبية موقعك وجعله أكثر احترافية عند مشاركته على منصات الشبكات الاجتماعية.

ويجب أيضًا مراجعة وتحديث هذه الميّة تاج بشكل دوري للتأكد من أنها محدثة وتتوافق مع تغييرات موقعك ومحفواه. ويمكن استخدام أدوات مثل "Facebook Debugger" و "Twitter Card Validator" لفحص وتحليل ميّة تاج موقعك والتأكد من أنها تعمل بشكل صحيح وتعرض بشكل جيد على منصات الشبكات الاجتماعية.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح وسم `iframe`

الوسم `iframe` هو عنصر `HTML` يسمح لك بتضمين صفحات ويب أو ملفات أخرى داخل صفحة ويب واحدة. يتم استخدام الوسم `iframe` على نطاق واسع لتحسين تجربة المستخدم عن طريق إظهار محتوى خارجي داخل صفحة ويب. فيما يلي مثال على استخدام الوسم `iframe` لعرض صفحة ويب أخرى داخل صفحة ويب:

HTML

```
<!DOCTYPE html>
<html>
<head>
    <title> عرض صفحة ويب داخل </title>
</head>
<body>
    <h1> عرض صفحة ويب داخل </h1>
    <p> لعرض صفحة ويب داخل صفحة ويب واحدة يمكن استخدام وسم <br>
        <iframe src="https://test.com/" width="800" height="600" framebor
</body>
</html>
```



في المثال أعلاه، يتم استخدام الوسم `iframe` لعرض صفحة ويب من موقع داخل صفحة HTML الأصلية. تم تعريف العرض والارتفاع لـ 800 و 600

بيكسل على التوالي، ولم يتم تمكين الإطار الجانبي لأنه لا يوجد حاجة إليه في هذا المثال.

يمكنك تجربة هذا المثال عن طريق نسخ الكود الموجود أعلاه ولصقه في ملف HTML ثم فتحه في متصفح الويب. سيتم عرض صفحة test.com داخل صفحة الأصلية باستخدام الوسم `iframe`.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح وسم embed

الوسم `embed` هو عنصر HTML يسمح لك بتضمين ملفات وسائط متعددة مثل الفيديو والصوت والتفاعلية داخل صفحة ويب واحدة. يمكن استخدام الوسم `embed` لعرض مقاطع الفيديو من مواقع مثل YouTube و Vimeo والملفات الصوتية من مثل SoundCloud.

فيما يلي مثال على استخدام الوسم `embed` لعرض مقطع فيديو من YouTube داخل صفحة ويب:

HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>YouTube عرض مقطع فيديو من</title>
</head>
<body>
    YouTube لعرض مقاطع الفيديو من موقع embed يمكن استخدام وسم<h1>
    لعرض صفة ويب داخل صفحة ويب واحدة iframe يمكن استخدام وسم<p>
    <embed src="https://www.youtube.com/embed/VIDEO_ID" width="800" height="450">
</body>
</html>
```

في المثال أعلاه، يتم استخدام الوسم `embed` لعرض مقطع فيديو من YouTube داخل صفحة HTML الأصلية. تم تعيين العرض والارتفاع لـ 800 و 600 بكسل على التوالي، ويجب تغيير "VIDEO_ID" إلى معرف الفيديو الفعلي على YouTube.

يمكنك تجربة هذا المثال عن طريق نسخ الكود الموجود أعلاه ولصقه في ملف HTML ثم فتحه في متصفح الويب. سيتم عرض مقطع فيديو YouTube داخل صفحة HTML الأصلية باستخدام الوسم `.embed`.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح عناصر الوسائط

تتضمن لغة (HTML) العديد من العناصر التي تساعد على تنسيق صفحات الويب وتوفير تفاعل المستخدم مع المحتوى المقدم. وفي هذا الدرس، سأقدم لكم شرحاً لبعض أهم عناصر الوسائط المستخدمة في لغة HTML وأمثلة على استخدامها.

:<`img`>

يستخدم هذا العنصر لإدراج صورة في صفحة HTML ويمكن التحكم في خصائص الصورة مثل عرض وارتفاع الصورة، كما يمكن إضافة نص بديل للصورة في حالة فشل تحميلها. الصيغة الأساسية لاستخدام هذا العنصر هي:

HTML

```

```

وفيما يلي مثال على استخدام العنصر <`img`>:

HTML

```

```

:<`audio`>

يستخدم هذا العنصر لتضمين ملف صوتي في صفحة HTML، ويمكن التحكم في خصائص الصوت مثل الحجم والتحكم بالصوت وتكرار الصوت. الصيغة الأساسية لاستخدام هذا العنصر هي:

HTML

```
<audio src="اسم الملف" controls="controls" volume="الحجم" loop="true">
```

وفيما يلي مثال على استخدام العنصر `<audio>`:

HTML

```
<audio src="example.mp3" controls="controls" volume="0.5" loop="true">
```

:`<video>`

يستخدم هذا العنصر لتضمين ملف فيديو في صفحة HTML، ويمكن التحكم في خصائص الفيديو مثل العرض والارتفاع والتحكم بالصوت وتكرار الفيديو. الصيغة الأساسية لاستخدام هذا العنصر هي:

HTML

```
<video src="example.mp3" controls="controls" volume="0.5" loop="true">
```

:`<iframe>`

يستخدم هذا العنصر لإدراج صفحة ويب داخل صفحة HTML أخرى، ويمكن تحديد عرض وارتفاع الصفحة المدرجة والتحكم في عملية التصفح بالداخل. الصيغة الأساسية لاستخدام هذا العنصر هي:

HTML

```
<iframe src="رابط الصفحة" width="عرض الصفحة" height="ارتفاع الصفحة">
```

وفيما يلي مثال على استخدام العنصر `<iframe>`:

HTML

```
<iframe src="https://www.google.com" width="600" height="400">
```

<embed>

يستخدم هذا العنصر لتضمين محتوى متعدد الوسائط في صفحة HTML، مثل مقاطع الفيديو أو الملفات الصوتية أو التطبيقات التفاعلية، ويمكن تحديد عرض وارتفاع المحتوى المضمن وتحديد أنماط العرض والتحكم في حجم العرض. الصيغة الأساسية لاستخدام هذا العنصر هي:

HTML

```
<embed src="اسم الملف" width="عرض المحتوى" height="ارتفاع المحتوى">
```

وفيما يلي مثال على استخدام العنصر <embed>:

HTML

```
<embed src="test.swf" width="400" height="300">
```

<object>

يستخدم هذا العنصر لإدراج محتوى متعدد الوسائط في صفحة HTML، ويمكن تحديد عرض وارتفاع المحتوى المضمن والتحكم في حجم العرض وتحديد أنماط العرض. الصيغة الأساسية لاستخدام هذا العنصر هي:

HTML

```
<object type="نوع الملف" data="اسم الملف" width="عرض المحتوى" height="ارتفاع المحتوى">
```



وفيما يلي مثال على استخدام العنصر <object>:

HTML

```
<object type="نوع الملف" data="اسم الملف" width="عرض المحتوى" height="ارتفاع المحتوى">
```



هذه بعض من العناصر الأساسية للوسائط في لغة HTML، وهي تساعد على تنسيق صفحات الويب وجعلها أكثر جذابة وتفاعلية أكثر، وتتوفر مجموعة متنوعة من

الخيارات لإضافة الوسائط المختلفة إلى صفحات الويب. لا يقتصر استخدام العناصر المذكورة أعلاه على الأمثلة المذكورة فقط، فهناك العديد من الخيارات الأخرى المتاحة في لغة HTML لإضافة الوسائط المختلفة إلى صفحات الويب. ويمكن للمطورين استخدام مزيج مختلف من العناصر والخصائص للتحكم في تنسيق وتصميم صفحات الويب بشكل أفضل وأكثر فعالية. وبما أن لغة HTML هي لغة وسليمة، فإنه يمكن توسيع الوظائف والإمكانات التفاعلية لصفحات الويب باستخدام العناصر المذكورة أعلاه وغيرها من العناصر الموجودة في لغة HTML، وبالتالي تحسين تجربة المستخدمين وزيادة جاذبية الموقع على الإنترنت.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح استخدام HTML مع CSS

تعتبر (HTML) (HyperText Markup Language) و (CSS) (Cascading Style Sheets) من أهم لغات الويب وتستخدمان معاً لتصميم صفحات الويب. يتم استخدام HTML لإنشاء تنسيقات وهيكل الصفحة، بينما يتم استخدام CSS لإضافة تنسيقات إضافية وتحسين الشكل العام للصفحة. يمكن استخدام CSS بعد إنشاء عناصر HTML على الصفحة وتحديد مظهرها. يمكن استخدام CSS لتعيين الخصائص الأساسية مثل اللون والحجم والخط والخلفية والهوامش، بالإضافة إلى تحديد مواضع العناصر على الصفحة.

يمكن استخدام CSS بأكثر من طريقة، ولكن الطريقة الأكثر شيوعاً هي تضمين CSS في صفحة HTML باستخدام عنصر `style`. يمكنك استخدام العنصر `style` داخل عنصر `head` في صفحة HTML لتعريف مظهر الصفحة. يمكن تحديد أي عنصر HTML وتعيين الخصائص المختلفة له باستخدام القواعد. على سبيل المثال، يمكنك تعيين خلفية الصفحة الرئيسية بلون أحمر بالطريقة التالية:

HTML

```
<head>
<style>
body {
background-color: red;
```

```

        }
    </style>
</head>
```

يمكن تعريف أي خاصية **CSS** في النطاق نفسه، يمكنك تعريف اللون والحجم والهامش وغيرها من الخصائص باستخدام هذه الطريقة.

بالإضافة إلى ذلك، يمكن استخدام **CSS** بطريقة خارجية، حيث يتم إنشاء ملف **link** منفصل ويتم ربطه بصفحة **HTML** باستخدام العنصر **link**. يتم تضمين العنصر **link** في عنصر **head** في صفحة **HTML** ويحتوي على مسار إلى ملف **CSS** الذي تريد استخدامه. على سبيل المثال، يمكن ربط صفحة **HTML** بملف **CSS** خارجي بهذه الطريقة:

HTML

```
<head>
    <link rel="stylesheet" href="style.css">
</head>
```

في هذا المثال، يوجد عنصر **link** الذي يحتوي على معرف "stylesheet" ومسار الملف "style.css". يمكنك إنشاء ملف **CSS** خارجي يحتوي على القواعد الخاصة بتصميم صفحتك، مثل الألوان والخطوط والتخطيط والهامش والحجم والخلفية وغيرها.

بالإضافة إلى الطرق السابقتين، يمكن استخدام **CSS** داخل العناصر **HTML** مباشرة باستخدام الخاصية **style**. يمكن تحديد الخصائص المختلفة للعناصر على الصفحة مباشرة داخل عناصر **HTML**. على سبيل المثال، يمكن تعريف لون النص داخل عنصر **p** باستخدام الخاصية **:style**:

HTML

```
<p style="color: red;>تغير لون هذا النص إلى الأحمر</p>
```

هذه هي بعض الطرق الأساسية لاستخدام **CSS** مع **HTML**. يمكنك استخدام **CSS** لتحسين مظهر صفحتك وجعلها أكثر جاذبية وتفاعلية للزوار.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح عناصر تنسيق النصوص

تنسيق النصوص في لغة HTML يشمل العديد من العناصر التي تسمح للمصممين بتخصيص مظهر النصوص، مثل الحجم والخط واللون والهوامش والتبعاد وغيرها. هناك عدة عناصر يمكن استخدامها لتنسيق النصوص، ومنها:

: هذا العنصر يستخدم لتحديد خصائص النص مثل الحجم واللون والخط.
يمكن استخدام الخاصية "size" لتحديد حجم النص والخاصية color لتحديد لون النص والخاصية face لتحديد الخط المستخدم. مثال:

HTML

```
<face size="5" color="red" face="Arial"></face>
```

: هذه العناصر تستخدم لتحديد حجم وعنوان العناصر. يتم استخدام <h1>...<h6> لعنوان الصفحة الرئيسي و <h2> للعناوين الفرعية وهكذا. مثال:

HTML

```
<h1>عنوان رئيسي</h1>
<h2>عنوان فرعي</h2>
...

```

: هذا العنصر يستخدم لإدخال فقرة من النص. يمكن استخدام الخاصية align لتحديد موضع النص داخل الفقرة. مثال:

HTML

```
<p align="center">فقرة نصية</p>
```

: هذه العناصر تستخدم لتحديد نص مهم أو مائل. يتم استخدام **** و **** لتحديد نص مهم و **** لتحديد نص مائل. مثال: ****

HTML

```
<p><b>هذا <em>نص مائل</em></b> وهذا <em>نص مهم</em></p>
```

: هذه العناصر تستخدم لإضافة خط سفلي أو خط متوسط على النص. يتم استخدام **<u>** لإضافة خط سفلي و **** لإضافة خط متوسط. مثال:

HTML

```
<p><u>هذا <del>نص بخط متوسط</del></u></p>
```

: هذا العنصر يستخدم لتحديد نص مهياً مسبقاً، حيث يتم عرض النص بالمحافظة على تنسيقه الأصلي، بدون تنسيق تلقائي للفقرات أو المسافات. مثال:

HTML

```
<pre>
    hello how are
        you today.
</pre>
```

: هذا العنصر يستخدم لتحديد نص مقتبس من مصدر آخر. يتم عرض النص بمسافة بادئة إضافية، غالباً ما يتم استخدام هذا العنصر لإدراج اقتباسات. مثال:

HTML

```
<blockquote>
    <p>هذا هو نص مقتبس</p>
        <cite>اسم المصدر</cite>
</blockquote>
```

<sup> و **<sub>**: هذه العناصر تستخدم لتحديد نص فوقى أو تحتى للنص. يتم استخدام **<sup>** لنص فوقى و **<sub>** لنص تحتى. مثال:

HTML

```
<p><sup>نص فوقى</sup><sub>نص تحتى</sub></p> وهذا
```

mark: يستخدم عنصر **mark** في لغة HTML لتحديد جزء من النص بأنه مهم أو يستحق الانتباه، وعادة ما يتم تمييز الجزء المحدد بلون الخلفية الأصفر. يمكن استخدام هذا العنصر لتحديد النصوص المهمة في الصفحة أو المفاهيم الأساسية التي يجب على القارئ الانتباه إليها.

مثال:

HTML

```
<p><mark>السعر هو 10 دولارات</mark></p>
```

del: يستخدم عنصر **del** في لغة HTML لتحديد جزء من النص كمحذوف، وعادة ما يتم عرض النص المحذوف بخط سريك على النص المحيط به. يمكن استخدام هذا العنصر لإظهار التغييرات في النصوص، مثل النصوص التي تم حذفها من النص الأصلي.

مثال:

HTML

```
<p><del>تمت إزالة الجملة تم حذفها</del></p> هذه الجملة تم حذفها
```

هذه هي بعض العناصر الأخرى التي يمكن استخدامها في لغة HTML لتنسيق النصوص. يمكن استخدام CSS لتطبيق تنسيقات أكثر تفصيلاً على النصوص، مثل تحديد الهوامش والتبعاد والخط والألوان وغيرها.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح وسم script

وسم `script` هو عنصر HTML يستخدم لتحديد كود البرمجة الذي يتم استخدامه في صفحة ويب محددة. يمكن استخدام هذا الوسم لإضافة تفاعلية ووظائف مختلفة إلى صفحة الويب.

يمكن استخدام وسم `script` بطريقتين: استخدام كود البرمجة مباشرة داخل الوسم، أو الإشارة إلى ملف خارجي يحتوي على الكود المراد استخدامه. فيما يلي بعض الأمثلة على استخدام الوسم `script` في صفحات الويب:

استخدام ال `JavaScript` في صفحة الويب المثال التالي يستخدم الوسم `script` لتحديد كود `JavaScript` الذي يعرض رسالة ترحيبية للمستخدم عندما يتم تحميل صفحة الويب:

HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>welcome</title>
</head>
<body>
    <script>
        alert('Welcome!');
    </script>
</body>
</html>
```

إشارة إلى ملف `JavaScript` خارجي يمكن استخدام الوسم `script` لإشارة إلى ملف `JavaScript` خارجي يحتوي على الكود المراد استخدامه في الصفحة. في المثال التالي، يتم إنشاء ملف `JavaScript` خارجي يتم استدعاؤه في صفحة HTML باستخدام الوسم `script`:

HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>welcome</title>
```

```
</head>
<body>
  <p>this is p</p>
  <script src="main.js"></script>
</body>
</html>
```

: (main.js محتوى ملف JavaScript)

JavaScript

```
document.querySelector("p").innerHTML = "The text has been changed";
```

باستخدام هذا الكود، يتم تحميل ملف JavaScript الخارجي "main.js" ويتم تغيير النص القديم إلى النص الحديث الذي قمنا بإضافته بشكل عام، يمكن استخدام الوسم "script" لإضافة العديد من الوظائف والتفاعلية المختلفة إلى صفحات الوي

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح HTML Entities and emoji

تعتبر الـ Entities عناصر هامة في HTML، حيث تُستخدم لتمثيل الحروف والرموز التي يكون من الصعب إدراجها مباشرة في صفحات الويب. يتم تعريف الـ "Entities" باستخدام شخصيات خاصة وعلامات محددة في HTML. وبما أن الحروف والرموز يجب أن تُعرض بشكل صحيح على جميع المستعرضات، فإن استخدام الـ Entities يسمح بضمان تمثيلها بشكل صحيح دون تغيير معناها. يتم تمثيل الـ Entities باستخدام علامة ampersand (&) وبعد ذلك توضع الشخصية المطلوبة وأخيرًا يتم إغلاق الـ Entity "Entity" بواسطة فاصلة منقوطة (;). فيما يلي بعض الأمثلة على استخدام الـ Entities في صفحات الويب:

تمثيل الأحرف الخاصة

```
<!DOCTYPE html>
<html>
<head>
    <title>welcome</title>
</head>
<body>
    <ul>
        <li>< - &lt;></li>
        <li>> - &gt;</li>
        <li>& - &amp;</li>
        <li>@ - &#64;</li>
    </ul>
</body>
</html>
```

تمثيل الأحرف ذات الأكواذ العشرية والثمانية يمكن استخدام الـ Entities لتمثيل الأحرف ذات الأكواذ العشرية والثمانية، مثل:

```
<!DOCTYPE html>
<html>
<head>
    <title>welcome</title>
</head>
<body>
    <ul>
        <li>&#x2600; - ☺</li>
        <li>&#x1F601; - 😊</li>
        <li>&#9733; - &#x2605; - ★</li>
    </ul>
</body>
</html>
```

يجب أن يتم استخدام الأكواذ الصحيحة لكل حرف لضمان عرضه بشكل صحيح على جميع المستعرضات.

على أي حال، فيما يلي بعض الأمثلة على بعض الـ "Entities" الشائعة المستخدمة في صفحات الويب:

< تمثل علامة "أقل من" (<) > تمثل علامة "أكبر من" (>)

(&) تمثل علامة "و" &

" " تمثل علامة الاقتباس (")

' ' تمثل علامة التنصيص الفردي (')

© تمثل علامة حقوق النشر (©)

® تمثل علامة العلامة التجارية (®)

€ تمثل رمز اليورو (€)

£ تمثل رمز الجنيه الاسترليني (£)

¥ تمثل رمز الين الياباني (¥)

° تمثل علامة الدرجة (°)

± تمثل علامة زائد أو ناقص (±)

× تمثل علامة الضرب (×)

÷ تمثل علامة القسمة (÷)

هناك المزيد من الـ "Entities" المتاحة، ويمكن استخدامها في صفحات الويب لتمثيل الرموز والحرروف والرموز الخاصة التي لا يمكن الكتابة بها مباشرة في HTML. تساعد استخدام الـ "Entities" في ضمان عرض النصوص والرموز بشكل صحيح على جميع المتصفحات والأجهزة. على سبيل المثال، يمكن استخدام الـ "Entities" لتمثيل حروف وحروف مشتقة من

لغات أخرى، مثل:

ـ تمثل حرف "A" مع حرف اللفظ القصير الإسباني (á)

ـ تمثل حرف "C" مع حرف اللفظ القصير الفرنسي (ç)

ـ تمثل حرف "O" مع الحرف اللفظ القصير الإيطالي (ò)

ـ تمثل حرف "S" مزدوج الحرف (ß) المستخدم في اللغة الألمانية ß

ـ كما يمكن استخدام الـ "Entities" لتمثيل الرموز الرياضية والرموز العلمية، مثل:

ـ تمثل علامة لا نهاية (∞)

ـ تمثل علامة المجموع (Σ)

ـ تمثل حرف "ألفا" اليوناني (α)

ـ تمثل حرف "بيتا" اليوناني (β)

HTML Emoji

يمكن استخدام الإيموجي (emoji) في صفحات الويب باستخدام عناصر HTML المخصصة لها، والتي تدعمها معظم المتصفحات الحديثة. يمكن استخدام الإيموجي في العناصر مثل

ـ و `<a>` و `<h1>` والعديد من العناصر الأخرى. يمكن استخدام الإيموجي مع النص العادي لإضفاء الشخصية على صفحات الويب وجعلها أكثر تفاعلية وجذابة. وهناك عدة طرق لإضافة الإيموجي إلى صفحات HTML، ومنها استخدام أكواد HTML الخاصة بالإيموجي، ويمكن الوصول إليها باستخدام الحروف أو الأكواد المختصرة للإيموجي. وفيما يلي بعض الأمثلة للاستخدام الصحيح للإيموجي في HTML: إضافة إيموجي بالحروف المباشرة:

HTML

```
<ul>
  <li> أهلاً بالعالم </li>
</ul>
```

إضافة إيموجي باستخدام الرموز المختصرة:

HTML

```
<ul>
  <li> أهلاً بالعالم &#x1F30E;</li>
</ul>
```

وفيما يلي قائمة ببعض أكواد الإيموجي الأكثر شيوعاً:

😀	: 😊
😃	: 😃
😄	: 😄
😁	: 😆
😆	: 😅
😉	: 😇
😊	: 😈
😍	: 😍
😘	: 😢
😜	: 😜
😝	: 😕
😛	: 😋
😎	: 😎
😏	: 😏
😇	: 😌
🤔	: 😰
🤨	: 😔
🤯	: 😐
😴	: 😴
🥰	: 😸

- ❤ : 
- 🎉 : 
- 🎁 : 
- ...

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

HTML Symbols شرح

تستخدم **HTML Symbols** لتمثيل رموز خاصة ورموز الإيموجي في صفحات الويب. ويمكن استخدام رموز **HTML Symbols** بدلاً من استخدام رموز **ASCII** أو رموز الإيموجي التي ليس لها مرادف بسيط في **ASCII**. يتم تمثيل **HTML Symbols** عن طريق الجمع بين رمز أمثلة & ومعرف الرمز ، ويجب عليك دائمًا إغلاق الرمز بـ ; بعد اسم الرمز .

HTML> greater | less than sign | >
HTML less than sign | يستخدم لتمثيل علامة الأقل من في علامات HTML.

ampersand | & .HTML
 quotation mark | " .HTML | يستخدم لتمثيل علامات الاقتباس في HTML.

apostrophe | ' .HTML | يستخدم لتمثيل الفاصلة الفارغة في HTML.

copyright sign | © .HTML | يستخدم لتمثيل علامة حقوق النشر في HTML.

registered sign | ® .HTML | يستخدم لتمثيل علامة تسجيل العلامة التجارية في HTML.

heart symbol | ❤ .HTML | يستخدم لتمثيل رمز القلب في HTML.

star symbol | ☆ .HTML | يستخدم لتمثيل رمز النجمة في HTML.

checkmark symbol | ✓ .HTML | يستخدم لتمثيل رمز علامة الصح في HTML.

يمكنك استخدام هذه الرموز على سبيل المثال في عناوين الصفحات أو النصوص التوضيحية أو عناصر النموذج والكثير من الأماكن الأخرى في صفحات الويب.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح buttons and inputs

buttons

وسم **button** هو عنصر HTML يستخدم لإنشاء زر يمكن للمستخدمين النقر عليه للقيام بإجراء معين في صفحة الويب، مثل إرسال نموذج أو تحميل ملف أو تنفيذ إجراء معين. يمكن استخدام هذا الوسم لإنشاء أزرار مختلفة مثل أزرار الإرسال، الإلغاء، والحذف وغيرها.

الشكل الأساسي لوسم **button** هو كالتالي:

HTML

```
<button>Click Me</button>
```

هنا بعض الخصائص الشائعة لوسم **:button**:
disabled: يتم استخدام هذه الخاصية لتعطيل الزر ومنع المستخدمين من النقر عليه. يمكن تعريف القيمة **false** على لتعطيل الزر وعلى **true** لتفعيله.

HTML

```
<button disabled>Disabled</button>
```

و **name** و **value**: يتم استخدام هذين الخصائص لتمرير قيم إضافية من خلال نموذج HTML.

HTML

```
<button name="submit" value="1">Submit</button>
```

:type: يتم استخدام هذه الخاصية لتحديد نوع الزر والتأكد من تنفيذ الإجراء المناسب عند النقر عليه. يمكن تعريف القيمة **submit** لإرسال النموذج، و **reset** لإعادة تعريف النموذج، و **button** لتنفيذ إجراء معين بدون إرسال النموذج.

HTML

```
<button type="submit">Submit</button>
```

يمكن استخدام وسم button بشكل شائع في صفحات الويب لإنشاء أزرار النموذج وغيرها من الأزرار التي تحتاج إلى إجراءات من المستخدم. كما يمكن تخصيص هذا الوسم باستخدام CSS لتعديل الشكل والتصميم وفقاً لاحتياجات الموقع.

inputs

يُستخدم عنصر input في HTML لتمكين المستخدم من إدخال بيانات على صفحة الويب، مثل النصوص والأرقام والتاريخ والإيميلات والصور والملفات والمزيد. يعتبر input عنصراً مهماً في إنشاء نماذج HTML لجمع المعلومات وإرسالها إلى الخادم للمعالجة.

يمكن استخدام عدة أوضاع لعنصر input بتحديد الخاصية type وفقاً للبيانات المراد إدخالها، مثل:

.text: لإدخال النصوص والعبارات القصيرة.

HTML

```
<label for="name">Enter your name:</label>
<input type="text" id="name" name="name" required>
```

.password: لإدخال كلمة المرور والإخفاء الحروف المكتوبة.

HTML

```
<label for="password">Enter your password:</label>
<input type="password" id="password" name="password" required>
```

.email: لإدخال عنوان البريد الإلكتروني، ويتحقق من صحته تلقائياً.

HTML

```
<label for="email">Enter your email:</label>
<input type="email" id="email" name="email" required>
```

.number: لإدخال الأرقام والأعداد، ويمكن تحديد الحد الأدنى والأقصى.

HTML

```
<label for="age">Enter your age:</label>
<input type="number" id="age" name="age" min="18" max="100" required>
```

: لإدخال التواريخ والتوقيت، ويمكن تحديد التنسيق.

HTML

```
<label for="date">Enter your date:</label>
<input type="date" id="date" name="date" required>
```

: لإدخال الوقت والتوقيت، ويمكن تحديد التنسيق.

HTML

```
<label for="time">Enter your time:</label>
<input type="time" id="time" name="time" required>
```

: للسماح للمستخدمين بتحميل الملفات.

HTML

```
<label for="interests">Select your interests:</label>
<input type="checkbox" id="sports" name="interests" value="Sports">
<label for="sports">Sports</label>
<input type="checkbox" id="music" name="interests" value="Music">
<label for="music">Music</label>
<input type="checkbox" id="travel" name="interests" value="Travel">
<label for="travel">Travel</label>
```

: للسماح للمستخدمين بتحديد خيار واحد من بين الخيارات المتاحة.

HTML

```
<label for="gender">Select your gender:</label>
```

```
<input type="radio" id="male" name="gender" value="Male">
<label for="male">Male</label>
<input type="radio" id="female" name="gender" value="Female">
<label for="female">Female</label>
```

submit: لإرسال البيانات المدخلة إلى الخادم.

HTML

```
<input type="submit" value="go">
```

reset: لإعادة تعيين البيانات المدخلة.

HTML

```
<input type="reset" value="Reset">
```

hidden: لإخفاء حقل الإدخال عن المستخدمين.

HTML

```
<input type="hidden" id="user" name="user" value="ayoub">
```

color: لإدخال الألوان.

HTML

```
<label for="color">Select Color</label>
<input type="color" id="color" name="color" value="#000000">
```

هذه هي بعض الأوضاع الشائعة لعنصر input في HTML، ويمكن استخدام العديد من الخيارات والأوضاع الأخرى لتلبية احتياجات تطبيقك.

placeholder attribute

السما placeholder في HTML تستخدم لتوفير نص مساعدة للمستخدمين في حقل الإدخال عندما يكون الحقل فارغاً. تساعد هذه السمة المستخدمين على فهم ما المتوقع منهم لإدخاله في الحقل، وتساعد أيضاً في توفير تجربة مستخدم أفضل عند التفاعل مع موقع الويب أو التطبيق.

يمكن استخدام السمة placeholder على أي عنصر input ، مثل "input type="text" ، و "input type="password" و "input type="email" وما إلى ذلك. لا يعرض النص الموجود في السمة placeholder قيمة مدخلة حقيقية، بل يتم إزالته تلقائياً عندما يبدأ المستخدم بإدخال قيمة في الحقل.

يمكن تعريف قيمة السمة placeholder باستخدام النص الذي تريد عرضه في الحقل. على سبيل المثال، يمكن استخدام الكود التالي لعرض نص مساعدة في حقل البريد الإلكتروني:

HTML

```
<input type="email" name="email" placeholder="Enter your email address">
```

تظهر قيمة السمة placeholder في مكان الحقل الفارغ حتى يقوم المستخدم بالنقر فوق الحقل للبدء في الإدخال. يتم حذف قيمة السمة placeholder تلقائياً عندما يبدأ المستخدم بالكتابة في الحقل.

يمكن تخصيص نمط ولون النص في السمة placeholder باستخدام CSS. على سبيل المثال، يمكن استخدام الكود التالي لتخصيص لون النص:

CSS

```
::placeholder {  
    color: #999;  
}
```

باستخدام السمة placeholder بشكل صحيح، يمكن تحسين تجربة المستخدم وتسهيل عملية التفاعل مع موقع الويب أو التطبيق.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى
دعمنا بمشاركة الدرس لكي تعم الفائدة.

شرح forms

يمكن إنشاء نماذج HTML باستخدام عناصر النموذج. يتم استخدام هذه العناصر لجمع البيانات من المستخدمين عبر الويب. فيما يلي بعض الأمثلة الشائعة لعناصر النموذج في HTML:

1) الحقل `input`: يستخدم هذا العنصر لإنشاء حقل يمكن للمستخدمين إدخال البيانات فيه، وتستخدم القيمة الخاصة بـ "النوع" لتحديد نوع الحقل. على سبيل المثال، يمكن استخدام `"type=text"` لإنشاء حقل نص، أو `"type=email"` لإنشاء حقل بريد إلكتروني.

HTML

```
<input type="text" name="firstname" placeholder="الاسم الأول">
<input type="email" name="email" placeholder="البريد الإلكتروني">
```

2) الحقل `textarea`: يستخدم هذا العنصر لإنشاء حقل نص طويلاً يمكن للمستخدمين إدخال البيانات فيه، وتستخدم القيمة الخاصة بـ "النوع" لتحديد نوع الحقل.

HTML

```
<textarea name="message" rows="5" cols="40">
write letter text here...
</textarea>
```

3) الحقل `select`: يستخدم هذا العنصر لإنشاء قائمة من الخيارات يمكن للمستخدمين اختيار واحدة منها، وتستخدم العناصر `option` لتحديد الخيارات.

HTML

```
<select name="gender">
  <option value="male">ذكر</option>
  <option value="female">أنثى</option>
</select>
```

Forms

يمكن كتابة وسم form في HTML باستخدام العنصر `form`, ويستخدم هذا الوسم لتحديد مجموعة من عناصر النموذج التي تشكل النموذج الكامل. يجب عليك تحديد الطريقة المستخدمة لإرسال البيانات (GET أو POST) باستخدام القيمة الخاصة بـ "الطريقة" في الوسم `.form`.

هناك العديد من الخيارات الممكنة لوسم form في HTML، ولكن الشكل الأساسي لوسم form يبدو كالتالي:

HTML

```
<form action="url" method="POST/GET">
    -- مجموعة من عناصر النموذج --
</form>
```

حيث:

`action`: يحدد URL الذي سيتم إرسال البيانات إليه بمجرد تقديم النموذج. يمكن استخدام رابط URL الذي يؤدي إلى ملف PHP أو أي لغة برمجة أخرى لمعالجة البيانات المرسلة من النموذج.

`method`: يحدد طريقة إرسال البيانات. يمكن استخدام POST أو GET، حيث يتم استخدام POST عندما تحتوي النماذج على بيانات سرية ويتم استخدام GET عندما تحتوي النماذج على بيانات عامة.

يمكن إضافة العناصر المختلفة المذكورة أعلاه داخل الوسم `form` وذلك بحسب الاحتياجات، وفي النهاية يتم إضافة زر "submit" للنموذج لإرسال البيانات إلى الخادم.

HTML

```
<form action="/action_page.php" method="POST">
    <label for="name">الاسم</label>
    <input type="text" id="name" name="name"><br><br>
    <label for="email">البريد الإلكتروني</label>
```

```
<input type="email" id="email" name="email"><br><br>
<label for="password">كلمة المرور</label >
<input type="password" id="password" name="password"><br><br>
<input type="submit" value="إرسال">
</form>
```

يتم تحديد العنصر `label` لربط التسمية بحقل الإدخال، مما يجعل النموذج أكثر قابلية للاستخدام.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

HTML Graphics شرح

يمكن استخدام عناصر الرسوم البيانية في HTML لإضفاء جاذبية على الموقع وتحسين تجربة المستخدم. وفيما يلي بعض العناصر الرسومية الشائعة في HTML:

1) الصور: `img`

يستخدم عنصر `img` لإدراج الصور في صفحة HTML، حيث يمكن تحديد مسار الصورة (`src`)، الوصف (`alt`)، العرض (`width`) والارتفاع (`height`) لتعيين الأبعاد المناسبة للصورة.

HTML

```

```

2) الشعارات: `svg`: يمكن استخدام الشعارات SVG لإدراج رسومات متوجهة قابلة للتغيير في HTML، حيث يمكن تعديل حجم الرسومات وألوانها بسهولة.

HTML

```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="</circle>
```

(3) الخرائط الحرارية **canvas**: يمكن استخدام عنصر **canvas** لإنشاء الخرائط الحرارية ورسومات أخرى متقدمة باستخدام البرمجة النصية، حيث يمكن تعريف أبعاد العنصر والأوامر البرمجية الالزامية لإنشاء الرسوم (width, height).

HTML

```
<canvas id="myCanvas" width="200" height="100"></canvas>
<script>
  var c = document.getElementById("myCanvas");
  var ctx = c.getContext("2d");
  ctx.fillStyle = "red";
  ctx.fillRect(10, 10, 150, 80);
</script>
```

(4) الرسوم المتحركة **gif**: يمكن استخدام ملفات الصور المتحركة **GIF** لإضافة رسوم متحركة إلى صفحات HTML، حيث يتم تشغيل الصور بتتابع سريع لإنشاء تأثير الحركة.

HTML

```

```

هذه بعض العناصر الرسمية الشائعة في HTML، وهناك العديد من الخيارات الأخرى المتوفرة لإضفاء جاذبية على الموقع. يمكن استخدام هذه العناصر بشكل فعال لتحسين تجربة المستخدم وتوفير معلومات أكثر بصرية وجذابة للزوار. يجب على المطورين الحرص على استخدام هذه العناصر بحكمة وتنسيقها بشكل صحيح مع النص والمحتوى العام للصفحة.

وهكذا إنتهينا من درسنا أو مقالتنا المتواضعة أتمنى أنك فهمت و وفقك الله أتمنى دعمنا بمشاركة الدرس لكي تعم الفائدة.

كيفية كتابة كود سريع

في هذه المقالة سنتحدث عن كيفية الكتابة بشكل أسرع في محرر visual studio طريقة كتابة كود سريع بإستعمال Emmet اعتبارها مثل اختصارات لكتابة code

الأوسم هنا سنتعرف على طريقة كتابة عنصر داخل عنصر بطريقة سريعة كنا في السابقة نفتح وسم ونتهيه ونغلقه ونكتبه داخله وسم آخر وهذا بطيء جداً لهذا سنكتب بطريقة الـ Emmet أولاً لكتابة عنصر داخل عنصر نكتب هكذا

```
div>h1
```

ستكون النتيجة عند الطغظ على Enter

```
<div>
  <h1></h1>
</div>
```

كما ترون بدل كتابة كل هذا سنكت بطريقة مختصرة كنا ترون الآن كيف أكتب نص داخلها بهذه الطريقة :

```
div>h1{Hello World}
```

كما ترون بسيطة نضغط على Enter ستكون النتيجة

```
div> <h1>Hello </h1>
      <World></h1> </div>
```

```
<div>
  <h1>Hello World</h1>
</div>
```

أعجبك الأمر نعم ستكون الآن سريعاً في كتابة الكود الخاص بك الآن لكتابة السمات أو هنا لكتابة attributes id and class بهذه الطريقة

```
.nama
```

ستكون النتيجة عند الطغظ على Enter

```
<div class="name"></div>
```

كما ترون نحدد الكلاس فقط بنقطة هنا لماذا ظهر معنا div لأنه وسم مشهور لتحديد عنصر معين نكتب

`h1.test`

النتيجة ستكون

```
<h1 class="test"></h1>
```

كنا ترون أمر بسيط نفس الشيء بالنسبة لل id نعم أعتقد أنك خمنت فيها انت ذكي
نستعمل بذل النقطة علامة الهاش # لكتابة ال id

`#nama`

ستكون النتيجة

```
<div id="name"></div>
```

كما ترون أمر بسيط هنا لدينا عنصر div للتغييره نكتب هكذا

`p#test`

ستكون النتيجة

```
<p id="test"></p>
```

هنا تعرفنا على ال id و ال class لعمل attributes أخرى نكتب هكذا

`div[title="test"]`

بهذه الطريقة هناك يمكنك كتابة أي سمة تريدها نطغظ على Enter ستكون النتيجة

```
<div title="test"></div>
```

الآن إذا أردنا كتابة عنصران مع بعض يعني ليس عنصر داخل عنصر سنكتب عنصر
يليه عنصر نكتب هكذا بعلامة +

div+p

ستكون النتيجة عند الطغظ على p

```
<div></div>
<p></p>
```

كنا ترون الأمر بسيط جدا الأن مع التكرار يمكننا تكرار العنصر أكثر من مرة بدلا من كتابتها بنفسك مثلا تريد عمل 100 li ستكتتبها بنفسك ولكن بإستعمال ال Emmet ستكتتبها في ثانية نستعمل هنا علامة *

```
ul > li * 3
```

هنا ستكون النتيجة

```
<ul>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

هنا كما ترون يمكنك دمج كل ماسبق نعم وكتابة نص داخلها و attribute معينة هناك العديد من العلامات و الإختصارات يمكنك البحث عن المزيد مثل grouping علامة \$ علامة ^ هناك العديد ماذا تنتظر كن سريعا في كتابة الكود الخاص بك وشكرا أتمنى أن تعجبكم مقالتنا المتواضعة و البسيطة وشكرا لكم.

This is how our book ended. we will add more in this book. for more books and free lessons, visit our website
<https://ayoubayaou.vercel.app>

