مقدمة في CSS

**مقدمة في CSS**

في هذهِ الدورة ستتعلم CSS والتي تعتبر اختصار من Cascading Style Sheets. وهي لغة لوصف تنسيق ملفات HTML وكيفية عرض عناصرها، مما يعني أن CSS هي جزء أساسي من تصميم الويب أو Web design لأنها تحدد طريقة عرض وتنسيق صفحات الويب، ويمكن كتابة HTML بدون CSS لكن ستكون طريقة عرض العناصر بسيطة وغير منسّقة وجذّابة، كما أنه لايُمكن كتابة CSS بدون HTML، باختصار CSS عبارة عن مجموعة من القواعد للتنسيق تسمح للمتصفح معرفة كيفية عرض العناصر الموجودة بملف HTML، وتساعد على تفاعل المستخدم مع صفحات الويب.

**طريقة كتابة CSS وCSS Syntax**

في CSS هناك جزئيين أساسيين عند تعريف أي من طرق التنسيق: 1- الخاصيّة: هي اسم لنوع محدد للتنسيق. 2- القيمة: هي ما تريد تحديده لتلك الخاصيّة. مثال لتعريف خاصيّة في CSS:

color: red;

نُلاحظ في المثال أعلاه أن الخاصية هي Color أو لون والقيمة هي Red أو أحمر.

وهناك ثلاث طرق لإضافة كود CSS لتنسيق صفحات الويب:

**طريقة External CSS**

وهي عبارة عن External Style Sheet أو ملف منفصل يوجد فيه جميع التنسيقات ويمكنك تغيير تنسيق موقع كامل من ملف واحد فقط، ويجب لكل ملف HTML أن يحتوي على ما يشير لملف التنسيق CSS وذلك يكون عن طريق العنصر في ملف HTML. مثال على طريقة إضافة ملف CSS والإشارة له عن طريق العنصر من HTML:

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<h1>CSS Course </h1>

<p>Learning CSS is fun!</p>

</body>

</html>

وملف CSS المشار إليه في المثال أعلاه من الممكن أن يتم كتابته في أي Text Editor أو برنامج كتابة لكن بشرط أن يكون الملف بصيغة css ولا يحتوي الملف على أي من عناصر HTML. مثال على كيفية شكل ملف التنسيق محفوظ بصيغة css كالتالي:

h1 {

color: blue;

text-align: center;

}

p {

color: pink;

text-align: center;

}

**طريقة Internal CSS**

وهي طريقة يمكن استخدامها في حال هناك تنسيق مختلف لكل صفحة ويب ويكون التنسيق مكتوب داخليًا في ملف HTML وتحديدًا في داخل العنصر <style> الموجود في داخل العنصر . مثال على ذلك:

<!DOCTYPE html>

<html>

<head>

<style>

h1 {

color: blue;

text-align: center;

}

p {

color: pink;

text-align: center;

}

</style>

</head>

<body>

<h1>CSS Course </h1>

<p>Learning CSS is fun!</p>

</body>

</html>

**طريقة Inline CSS**

وهي طريقة يمكن استخدامها في حال كان هناك تنسيق مُختلف لكل عنصر. ويكون التنسيق مكتوب بداخل العنصر المراد تنسيقه وذلك بإضافة السمة style والتي من الممكن أن تحتوي على خصائص CSS. مثال على ذلك:

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<h1 style="color:blue;text-align:center;">CSS Course </h1>

<p style="color:pink;text-align:center;">Learning CSS is fun!</p>

</body>

**مفهوم Selectors**

يستخدم Selector لتحديد ماهو العنصر المراد تنسيقه من عناصر HTML فيجب قبل أي عملية تنسيق وكتابة خاصيّة وقيمة لها يتم قبلها تحديد Selector وهو الذي يحدد العنصر الذي سيتم تطبيق التنسيق عليه، مثال على طريقة كتابة Selector كالتالي:

A computer screen shot of a computer code

Description automatically generated

وهناك عدة أنواع من Selectors ومن أبرزها: Universal Selector وهو الذي تكون بدايته علامة \* وClass Selector الذي يبدأ بنقطة متبوعه باسم Class تم وضعه مسبقًا بداخل العنصر المراد تنسيقه وElement Selector وهو الذي تكون بدايته اسم العنصر المراد تنسيقه مثل h1 وId Selector الذي يبدأ باسم Id تم وضعه مسبقًا بداخل العنصر المراد تنسيقه، وهناك أيضًا Grouping Selector وهو الذي يختار أو يجمع العناصر المراد تنسيقها بنفس التنسيق مثل: h1, h2, p وتم الفصل بينهم بعلامة الفاصلة.

**الألوان**

هناك عدة طرق لإضافة الألوان في CSS إما عن طريق كتابة اسم اللون مباشرة أو بطريقة RGB أو بطريقة RGBA وأخيرًا طريقة Hexadecimal والتي تكون بدايتها بعلامة #.

تنسيق الخلفيات في CSS

هناك خصائص في CSS لتنسيق الخلفيات مثل:

* خاصيّة background-color لتغيير لون خلفيّة العنصر.
* خاصيّة background-image لتغيير خلفيّة العنصر بحيث تكون صورة.
* خاصيّة background-repeat للتحكم بتكرار الخلفيّة.
* خاصيّة background-position للتحكم بموقع الخلفيّة.
* خاصيّة background-attachment للتحكم ما إذا الخلفيّة تكون ثابتة fixed أو scroll متحركة.

وهناك أيضًا خاصيّة لتقليل أسطر الكود وجمع الخصائص للخلفيّة بسطر واحد فقط لاحظ في المثال التالي بدون اختصار لخصائص الخلفيّة:

body {

background-color: #ffffff;

background-image: url("img\_tree.png");

background-repeat: no-repeat;

background-position: right top;

}

ويمكن اختصار الكود لقيمة الخصائص بالشكل التالي:

body {

background: #ffffff url("img\_tree.png") no-repeat right top;

}

في المثال أعلاه تم اختصار الخصائص في سطر واحد وبالترتيب تكون القيمة الأولى للخاصيّة background-color والقيمة الثانية للخاصيّة background-image والقيمة الثالثة للخاصيّة background-repeat والقيمة الرابعة للخاصيّة background-position.

| **الخاصية** | **عملها** |
| --- | --- |
| color | لتغيير لون النص |
| direction | تحدد إتجاه النص مثلاً من اليمين إلى اليسار أو العكس |
| letter-spacing | تزيد أو تنقص المساحة بين الحروف |
| line-height | لتحديد طول السطر |
| text-align | يحدد المحاذاة الأفقية للنص |
| text-decoration | يحدد التزيين المضاف للنص |
| text-indent | يحدد المسافة في بداية السطر الأول في أي Block نصي |
| text-shadow | يحدد تأثير الظل المضاف إلى النص |
| text-transform | يتحكم في الكتابة بالأحرف الكبيرة للنص |
| text-overflow | لتحديد كيفية عرض المحتوى الزائد |
| vertical-align | يحدد المحاذاة العامودية للنص |
| white-space | يحدد كيفية التحكم بالمساحة البيضاء للعنصر |
| word-spacing | تزيد أو تنقص المساحة بين الكلمات |

# الخطوط

هناك عدد من الخصائص التي يمكن استخدامها لتنسيق الخطوط في CSS، منها ما يستخدم لتغيير نوع الخط أو حجمه أو شكله، فيما يلي سنتعرف على أهمها،

### خاصية font-family

تستخدم لتحديد نوع خط النص، فمثلًا لو أردنا إضافة نوع الخط Arial فيمكننا كتابته على النحو التالي:

p {

font-family: Arial;

}

خاصية font-family من الممكن أن تأخذ أكثر من قيمة خط لإنشاء ما يسمى Fallback System أو نظام الاحتياط، في حال أن المتصفح لا يدعم نوع الخط الأول فإنه ينتقل لنوع الخط الثاني وهكذا، يمكننا القيام بذلك بهذا الشكل:

p {

font-family: Arial, "Times New Roman", sans-serif;

}

يمكن أيضًا اختصار الكود بجمع القيم لتنسيق الخطوط ضمن خاصية واحده فقط وهي font ويكون ترتيب القيم فيها كالتالي:

* قيمة الخاصية font-style
* قيمة الخاصية font-variant
* قيمة الخاصية font-weight
* قيمة الخاصية font-size/line-height
* قيمة الخاصية font-family

تنسيق الإطارات في CSS

**الإطارات**

لتنسيق الإطارات في CSS هناك خصائص لتغيير شكل الإطار وحجمه ولونه:

* خاصيّة border-style تستخدم لتغيير شكل الإطار
* خاصيّة border-color لتغيير لون الإطار
* خاصيّة border-width لتغيير حجم الإطار
* خاصيّة border-radius لتغيير حدة زوايا الإطار

في خاصيّة border-style من الممكن تحديد جهة على وجه الخصوص بتغييرها وحدها بشكل مختلف عن جهات الإطار الأخرى مثال على ذلك:

p {

border-top-style: solid;

border-right-style: dashed;

border-bottom-style: solid;

border-left-style: dotted;

}

ويمكن أيضًا اختصار القيم بسطر واحد وتكون القيمة الأولى للجهة العلوية والقيمة الثانية للجهة اليمنى والقيمة الثالثة للجهة السفلية والقيمة الرابعة للجهة اليسرى، مثال على ذلك:

p {

border-style: solid dashed solid dotted;

}

نظرة على Box Model

في CSS جميع عناصر HTML يتم النظر إليها على أنها Box أو صندوق يحتوي على عدة طبقات وهي:

* أولًا المحتوى: وهو مايحتويه العنصر من نص أو صورة.
* ثانيًا خاصيّة Padding: وهي المسافة بين المحتوى والإطار المحدد للمحتوى.
* ثالثًا خاصيّة Border: والتي تعتبر إطار حول المحتوى وPadding.
* رابعًا Margin: وهي المسافة بين إطار العنصر وأي شيء حوله.

A screenshot of a computer screen

Description automatically generated

مثال على خصائص Box model:

div {

width: 500px;

border: 5px solid pink;

padding: 30px;

margin: 50px;

}

تنسيق الصفحة

## الخاصيّة position

في CSS يمكنك تحديد موقع أي عنصر في الصفحة، الخاصيّة position تسمح بتحديد موضع العنصر في الصفحة وهناك خمس قيم لهذه الخاصيّة:

### القيمة static

وهي القيمة الافتراضية لجميع العناصر ولن تكون قادرًا على تحريكها بأي جهة عند استخدام هذهِ القيمة.

### القيمة relative

وهي التي تسمح بوضع العنصر بأي موقع نسبةً إلى موقعه الأصلي وعند تحديد الخصائص top أو left أو right أو bottom سيتحرك العنصر عن موقعه الأصلي.

### القيمة absolute

والتي تسمح بوضع العنصر نسبةً إلى أقرب عنصر تم إضافة خاصية position بقيمة relative له.

### القيمة fixed

والتي يكون العنصر فيها ثابتًا ولا يتحرك مهما تم النزول والصعود (scroll) بالصفحة.

### القيمة sticky

التي يكون فيها العنصر بموقعه وعند البدء بتحريك الصفحة والمرور من هذا العنصر عند التحرك بالصفحة (scroll) سيثبت العنصر بنفس الموقع الذي تم تحديده له.

## العناصر المتداخلة

هناك خاصية بسيطة ولكن لها فائدة كبيرة وهي z-index وهي التي تحدد ترتيب العناصر في حال أنهم على شكل طبقات. قيم الخاصية z-index من الممكن أن تكون بالسالب أو الموجب على حسب ترتيب العنصر، فالقيمة بالسالب تعني أن العنصر سيكون بالخلف والقيمة بالموجب تعني أنه سيكون في الأمام، مثال على ذلك:

img {

position: relative;

left: 0px;

top: 0px;

z-index: -1;

}

في المثال أعلاه سيكون موقع الصورة بالخلف كالتالي:

A blue and white logo

Description automatically generated