

# تصميم صفحات الويب

## Web Page Design

---

المحاضر: حسام يونسو

تخصص تقنية معلومات - كلية العلوم

- المخرجات المتوقعة من الدرس
- المقدمة
- الواجهة الأمامية (Frontend)
- الخلفية (Backend)
- أدوات تطوير الويب
- أساسيات HTML
- عناصر HTML
- تعرف على CSS
- § استخدامات CSS
- § مقدمة إلى JavaScript
- § أساسيات JavaScript
- § التحكم بالتايل باستخدام JavaScript
- الخاتمة والأسئلة
- مراجع علمية للمادة

## المخرجات المتوقعة من الدرس

- اكتساب فهم شامل لتصميم الويب.
- التعرف على الأدوات المستخدمة في تصميم الصفحات.
- تعلم خطوات تصميم صفحة ويب.
- القدرة على استخدام الأدوات لتغيير الحالات في صفحة الويب.
- تطبيق عملي لتصميم صفحة ويب.

## ما هو تطوير الويب؟

تطوير الويب هو عملية إنشاء مواقع وتطبيقات الويب التي يمكن الوصول إليها من خلال الإنترنت أو الشبكات الداخلية. يتضمن تطوير الويب تصميم وبناء الصفحات والتطبيقات التي توفر معلومات وخدمات للمستخدمين. يمكن تقسيم تطوير الويب إلى جزئين رئيسيين:

- الواجهة الأمامية (Frontend)
- الخلفية (Backend)



# الواجهة الأمامية (Frontend)



- الجزء الذي يراه المستخدم ويتفاعل معه مباشرةً.
- يتضمن التصميم والتخطيط والألوان والخطوط والصور.
- يستخدم لغات مثل **HTML**، **CSS**، **JavaScript**.

# الخلفية (Backend)

- الجزء المسؤول عن معالجة البيانات وتخزينها.
- يتضمن قواعد البيانات، الخوادم، والمنطق البرمجي.
- يستخدم لغات مثل **PHP**، **Python**، **Node.js** مع قواعد البيانات مثل **MySQL** و **MongoDB**.



# أدوات تطوير الويب

تعد أدوات تطوير الويب حجر الأساس لأي مطور ويب، حيث تساعد في إنشاء وتصميم واختبار الصفحات والتطبيقات بكفاءة. تنقسم هذه الأدوات إلى ثلاث فئات رئيسية

- المتصفحات (Browsers)

- محررات الأكواد (Code Editors)

- بيئات التطوير المتكاملة (IDEs)



# المتصفحات (Browsers)

- برامج تُستخدم لعرض صفحات الويب.
- أشهر الأمثلة: Google Chrome، Mozilla Firefox، Safari.
- تحتوي المتصفحات على أدوات المطور (DevTools) لفحص الأكواد وتصحيح الأخطاء.





# محررات الأكواد (Code Editors)

برامج مخصصة لكتابة الأكواد البرمجية.

- الأمثلة الشائعة:

- **Visual Studio Code (VS Code):** خفيف وسريع

ويدعم الإضافات.

- **Sublime Text:** محرر بسيط وسريع يحتوي على ميزات

قوية لتحرير الأكواد.

- **Notepad++:** خيار خفيف وبسيط مناسب للمبتدئين.



# بيئات التطوير المتكاملة (IDEs)

- برامج متقدمة توفر أدوات إضافية مثل التصحيح التلقائي وإدارة الملفات.
- مثال: WebStorm و PhpStorm.



- 1- ما هو تطوير الويب؟
- 2- ما الفرق بين الواجهة الأمامية والخلفية؟
- 3- ما هي أشهر المتصفحات المستخدمة في تطوير الويب؟
- 4- ما هو أفضل محرر أكواد للمبتدئين؟

ما هو HTML؟

HTML هي اختصار لـ **HyperText Markup Language**، وهي لغة تُستخدم لبناء هيكل صفحات الويب.

أهمية HTML في بناء صفحات الويب

- تُعد HTML أساس أي موقع ويب، حيث تحدد الهيكل العام للمحتوى.
- تُستخدم لتنسيق النصوص، إدراج الصور، إنشاء الروابط والجداول، وتصميم النماذج.
- يمكن دمجها مع CSS لتنسيق الشكل الخارجي، ومع JavaScript لإضافة التفاعل.

# الهيكل الأساسي لصفحة HTML

- يحدد نوع المستند وإصداره (HTML5).
- العنصر الجذري الذي يحتوي على جميع عناصر الصفحة.
- يحتوي على معلومات حول الصفحة مثل العنوان والمخططات.
- يحدد عنوان الصفحة الظاهر في علامة التبويب.
- يحتوي على محتوى الصفحة المرئي للمستخدم.

```
<!DOCTYPE html>
<html>
  <head>
    <title>عنوان الصفحة</title>
  </head>
  <body>
    <h1>!مرحبًا بك</h1>
  </body>
</html>
```

# العناوين (Heading)

تُستخدم العناوين لتقسيم النصوص إلى أقسام مرتبة، وهي تبدأ من `<1h>` (الأكبر) إلى `<6h>` (الأصغر).

مثال:

```
</h1>العنوان الرئيسي  
</h2>العنوان الفرعي  
</h3>العنوان الفرعي  
</h4>العنوان الفرعي  
</h5>العنوان الفرعي  
</h6>العنوان الفرعي
```

## الفقرات (Paragraph)

تُستخدم الفقرات لعرض النصوص بشكل منظم.  
مثال:

</p>. هذه فقرة نصية <p>

تُستخدم الروابط للانتقال إلى صفحات أخرى أو مواقع مختلفة.  
مثال:

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



# الصور (Images)

تُستخدم الصور لإضافة محتوى مرئي إلى الصفحة.  
مثال:

```

```

## قوائم غير مرتبة (Unordered Lists):

تُستخدم لعرض عناصر بدون ترتيب رقمي.

مثال:

```
<ul>  
  <li>العنصر الأول</li>  
  <li>العنصر الثاني</li>  
</ul>
```

## قوائم مرتبة (Ordered Lists): تُستخدم

لترتيب العناصر بشكل مرقم.

مثال:

```
<ol>  
  <li>العنصر الأول</li>  
  <li>العنصر الثاني</li>  
</ol>
```

الحقول النصية والأزرار:

تُستخدم لإدخال البيانات من قبل المستخدمين.

```
<form>  
  <label for="name">الاسم:</label>  
  <input type="text" id="name" name="name">  
  <button type="submit">إرسال</button>  
</form>
```

```
<table>
  <tr>
    <th>الاسم</th>
    <th>العمر</th>
  </tr>
  <tr>
    <td>أحمد</td>
    <td>25</td>
  </tr>
  <tr>
    <td>ليلى</td>
    <td>30</td>
  </tr>
</table>
```

الجداول:

تُستخدم لعرض البيانات في شكل منظم.

- 1- ما هو HTML؟
- 2- ما أهمية HTML؟
- 3- ما هي أنواع العناوين في HTML؟
- 4- كيف أُضيف صورة أو رابط في HTML؟

CSS (Cascading Style Sheets) هي لغة تنسيق تُستخدم لتنسيق وتزيين صفحات الويب.

CSS توفر طريقة لفصل المحتوى (HTML) عن الشكل (التصميم).

تعريف CSS: هي لغة تُستخدم لوصف طريقة عرض العناصر الموجودة في الملفات المكتوبة بلغة HTML أو XML.



CSS تتحكم في شكل صفحة الويب بما في ذلك الألوان، الخطوط، الهامش، التباعد، والتخطيط.

على سبيل المثال: يمكنها أن تجعل عنوانًا يظهر بحجم أكبر أو نصًا يظهر بخلفية ملونة.

**ومن ميزات CSS:**

**تنسيق مرن وقابل للتخصيص:** CSS تسمح لك بإجراء تغييرات بصرية على صفحات متعددة باستخدام ملف CSS واحد.

**تحسين تجربة المستخدم:** تحسين المظهر والوظائف يجعل التنقل في الموقع أكثر سلاسة وجاذبية.

**فصل التصميم عن المحتوى:** فصل التصميم (CSS) عن المحتوى (HTML) يعني أنك تستطيع تحديث التصميم بسهولة دون الحاجة لتغيير المحتوى نفسه.

**التصميم المتجاوب CSS (Responsive Design):** تجعل من السهل ضبط المواقع لتبدو جيدة على أي جهاز سواء كان هاتفًا أو جهازًا لوحيًا أو كمبيوترًا مكتبيًا.

**تحسين سرعة تطوير المواقع:** باستخدام CSS يمكنك إجراء تغييرات سريعة على المظهر العام للموقع بفضل أسلوب التنسيق المركزي.

```
<h1 style="color: blue;">مرحبا بكم في CSS</h1>
```



**HTML:** يوفر الهيكل الأساسي للصفحة، مثل العناوين، الفقرات، الصور، والروابط.

**CSS:** تضيف مظهرًا إلى هذه العناصر، مثل الألوان، الخلفيات، والتخطيط.

**CSS لا تعمل وحدها:** يجب أن تكون مدرجة داخل HTML لكي يتم تطبيق التأثيرات البصرية. يتم ذلك بثلاث طرق رئيسية.




```
<h1 style="color: ■ red;">هذا عنوان</h1>  
<p>هذه فقرة.</p>
```



هذا عنوان

هذه فقرة.

```
<p style="color:  green;">هذا نص بلون أخضر</p>
```

**Inline CSS:** تتم إضافتها مباشرةً إلى العنصر باستخدام سمة style.

**المزايا:** سهولة وسريعة للتطبيق على عناصر محددة.

**العيوب:** غير قابلة للتكرار عبر الموقع كله.

**Internal CSS:** تُكتب داخل وسم `<style>` في رأس الصفحة.

**المزايا:** مفيد لتنسيق صفحة واحدة.

**العيوب:** يصبح التنسيق أقل فعالية عند العمل على مواقع كبيرة.

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Document</title>
  <style>
    p {
      color: blue;
    }
  </style>
</head>
```

```
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <title>Document</title>  
  <link rel="stylesheet" href="styles.css">  
</head>
```

**External CSS:** يُكتب في ملف CSS منفصل ويُربط مع صفحة HTML.

**المزايا:** تنسيق متعدد الصفحات بسهولة عبر ملف CSS واحد.

**العيوب:** يتطلب المزيد من التخطيط، ولكن يوفر مرونة أكبر.

# قواعد الكتابة الأساسية لـ CSS

**محدد (Selector):** يحدد العنصر الذي يتم تطبيق

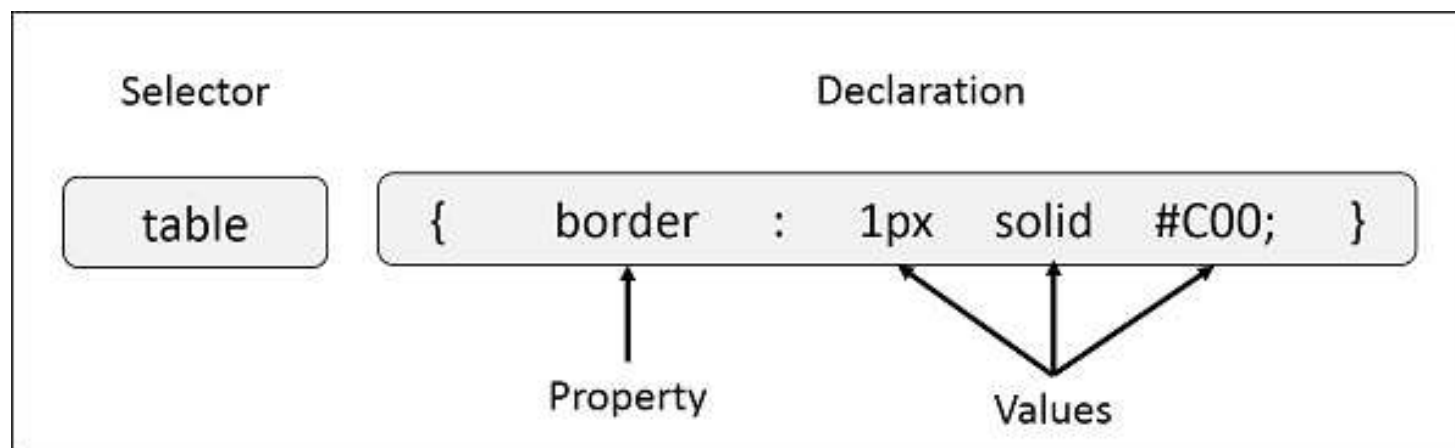
القواعد عليه.

**قيمة (Value):** تحدد كيفية تغيير السمة.

**الإعلان (Declaration):** يشمل الخاصية والقيمة معًا.

**خاصية (Property):** تحدد السمة المراد تغييرها (مثل

اللون، الحجم).



# المحددات (Selectors)

```
<style>  
  .info {  
    color: blue;  
  }  
</style>
```

**محدد الصنف (Class Selector):** يستهدف عناصر ذات

صنف معين باستخدام النقطة .class-name

مثال: تنسيق فقرة باستخدام الصنف "info"

```
<style>  
  h1 {  
    color: green;  
  }  
</style>
```

**محدد العنصر (Element Selector):** يستهدف

جميع العناصر من نفس النوع.

مثال: تنسيق جميع العناوين h1

**محدد المعرف (ID Selector):** يستهدف عنصرًا فريدًا باستخدام الرمز #  
مثال: تنسيق عنصر واحد باستخدام معرف

```
<style>  
  #header {  
    background-color: yellow;  
  }  
</style>
```

## أنواع الألوان:

الكلمات المفتاحية: استخدام كلمات مفتاحية مثل red, blue.

```
color: blue;
```

قيم HEX: تمثل الألوان باستخدام الرموز مثل #ff0000.

```
color: #ff0000;
```

قيم RGB: تحديد الألوان باستخدام قيم الأحمر والأخضر والأزرق.

```
color: rgb(11, 148, 61);
```

قيم RGBA: إضافة شفافية إلى ألوان RGB.

```
color: rgba(193, 29, 149, 0.5);
```



# الخطوط (Fonts) في CSS

تحديد نوع الخط **font-family**: يسمح لك بتحديد أنواع الخطوط.

حجم الخط **font-size**: يتحكم في حجم النص.

سماعة الخط **font-weight**: لتحديد درجة السماعة.

ارتفاع السطر **line-height**: يحدد المسافة بين السطور.

```
p {  
    font-family: Arial, sans-serif;  
    font-size: 18px;  
    font-weight: bold;  
    line-height: 1.6;  
}
```

# الهامش والحشوة (Margin & Padding)

الهامش (Margin): يمثل المسافة الخارجية بين العنصر والعناصر الأخرى.

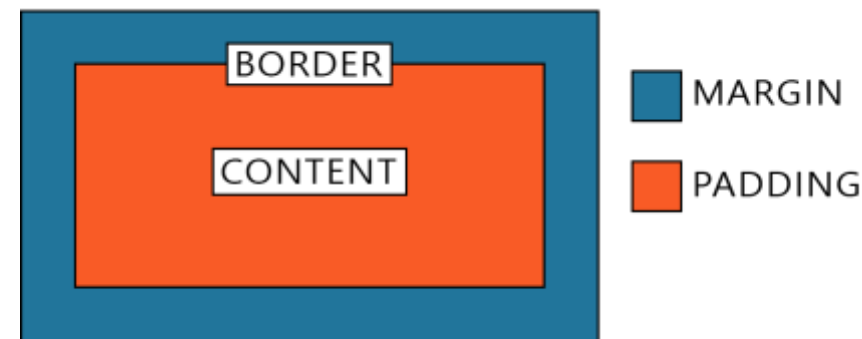
الحشوة (Padding): تمثل المسافة الداخلية بين حدود العنصر والمحتوى داخله.

الهامش مقابل الحشوة:

الحشوة مساحة فارغة داخل حدود العنصر.

الهامش مساحة فارغة خارج حدود العنصر.

```
div {  
  margin: 20px;  
  padding: 10px;  
}
```



# الحدود (Borders)

أنماط الحدود: يمكن استخدام الأنماط مثل **solid, dashed, dotted**.

أمثلة على حدود مخصصة:

**border-width:** لتحديد سماكة الحدود.

**border-radius:** لجعل الحواف مستديرة.

```
p {  
  border: 2px solid black;  
  border-radius: 10px;  
}
```



Bordered Paragraph

## مكونات نموذج الصندوق:

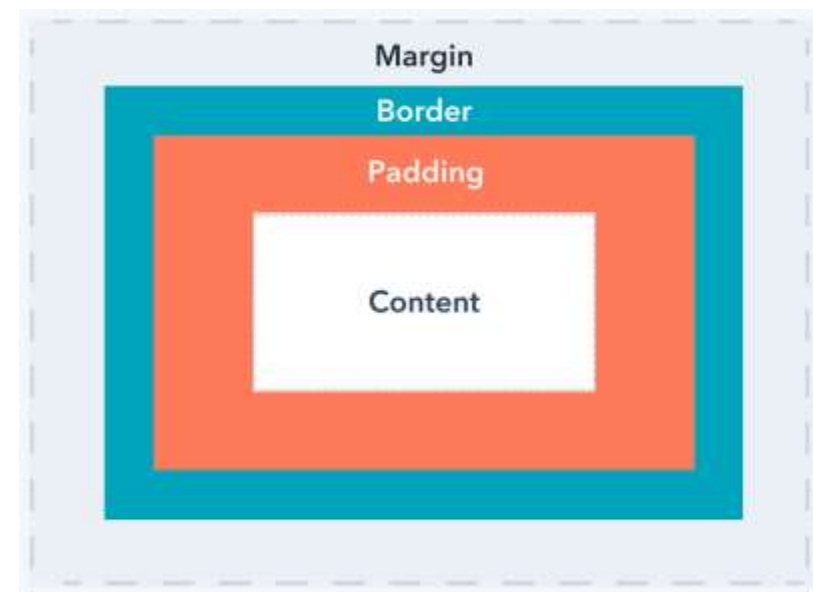
**المحتوى (Content):** النص أو الصور داخل العنصر.

**الحشوة (Padding):** المساحة بين المحتوى وحدود العنصر.

**الحدود (Border):** حدود تحيط بالعنصر.

**الهامش (Margin):** المساحة خارج الحدود.

**كيفية عمله:** يجب فهم كيفية حساب العرض والارتفاع النهائي للعناصر باستخدام نموذج الصندوق.



التحكم في العرض (Width) والارتفاع (Height) باستخدام وحدات مثل

px, em, %.

تحديد الأبعاد:

- الوحدات الثابتة مثل px للحجم المحدد.
- الوحدات النسبية مثل % للمرونة بناءً على حجم النافذة الخاصة بالمتصفح.

```
div {  
  width: 50%;  
  height: 200px;  
}
```

المثال في الأعلى يبين لنا أن عرض العنصر div يساوي نصف حجم عرض النافذة، وارتفاع ثابت 200 بيكسل.

# التخطيط والترتيب باستخدام Flexbox

ما هو **Flexbox**؟: نظام قوي لتخطيط العناصر بشكل متجاوب ومرتب داخل الحاويات (Containers).

أهم الخصائص:

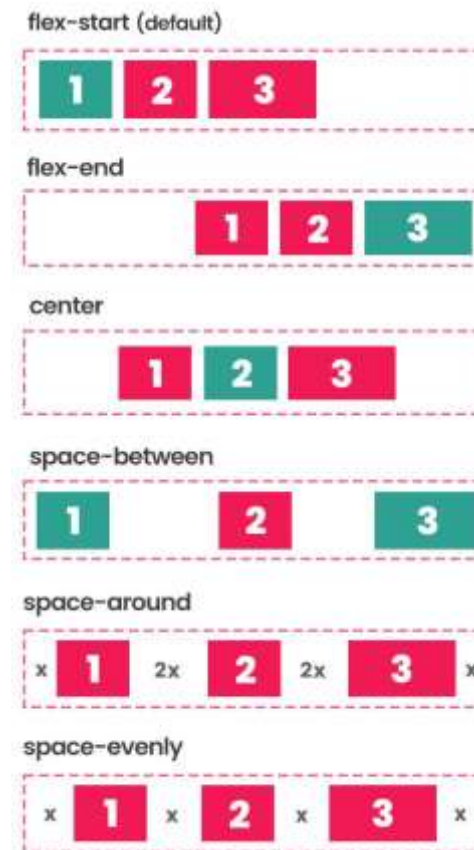
**display: flex:** يفعل Flexbox على الحاوية.

**justify-content:** توزيع العناصر أفقياً.

**align-items:** محاذاة العناصر عمودياً.

```
.container {
  display: flex;
  justify-content: center;
  align-items: center;
}
```

مثال على طريقة الإستخدام



# تأثيرات التفاعل (Hover Effects)

تأثيرات التفاعل CSS: تتيح تغيير مظهر العنصر عند تحويم (تمرير مؤشر (الماوس فوقه, يتم تحقيق ذلك باستخدام pseudoclass يسمى hover في CSS.

```
a:hover {  
  color: green;  
  text-decoration: underline;  
}
```

مثال لطريقة  
الإستخدام

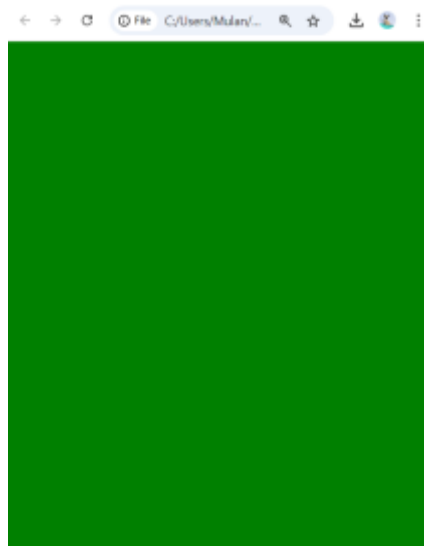
Google ← Google

الحالة الطبيعية

الحالة بعد تمرير  
المؤشر على العنصر

# التصميم المتجاوب (Responsive Design)

**Media Queries:** تسمح لك بضبط التصميم بناءً على حجم الشاشة أو الجهاز.  
مثال على استجابة تصميم CSS لشاشات الهواتف:



```
@media only screen and (max-width: 700px) {  
  body {  
    background-color: green;  
  }  
}
```

حجم شاشة أو نافذة أكبر  
بيكسل 700 من

حجم شاشة أو نافذة أصغر  
بيكسل 700 من

مثال على طريقة استخدام  
media query



**CSS Variables:** تتيح استخدام متغيرات لتخزين قيم يمكن إعادة استخدامها.

**فوائد المتغيرات:** تسهيل صيانة الكود عبر تعيين قيم متغيرة مرة واحدة واستخدامها في أماكن متعددة.

```
:root {  
  --main-color: blue;  
}  
  
p {  
  color: var(--main-color);  
}
```



# مقدمة إلى JavaScript

ما هي JavaScript؟

JavaScript هي لغة برمجة تُستخدم لإضافة التفاعل والديناميكية إلى صفحات الويب.

أهمية JavaScript

تجعل صفحات الويب تفاعلية مثل النماذج الديناميكية وعروض الشرائح.

تُستخدم في تطوير تطبيقات الويب والجوال.

يمكن دمجها بسهولة مع HTML وCSS.

## كتابة أول برنامج JavaScript

```
<!DOCTYPE html>
<html>
  <body>
    <h1>مرحبًا بك</h1>
    <button onclick="sayHello()">اضغط هنا</button>
    <script>
      function sayHello() {
        alert('JavaScript! مرحبًا بك في');
      }
    </script>
  </body>
</html>
```

## التفاعل مع العناصر

يمكن استخدام JavaScript لتغيير المحتوى والعناصر في الصفحة.

```
<!DOCTYPE html>
<html>
  <body>
    <p id="text">هذا نص.</p>
    <button onclick="changeText()">تغيير النص</button>
    <script>
      function changeText() {
        document.getElementById('text').innerHTML = 'تم التغيير';
      }
    </script>
  </body>
</html>
```

# التحكم بالسلاسل باستخدام JavaScript

تغيير الأنماط ديناميكيًا

يمكن استخدام JavaScript لتغيير الأنماط (CSS) للعناصر.

مثال لتغيير اللون والخط:

```
<body>
  <p id="styleText">هذا نص.</p>
  <button onclick="changeStyle()">غير النمط</button>
  <script>
    function changeStyle() {
      let text = document.getElementById('styleText');
      text.style.color = 'blue';
      text.style.fontSize = '20px';
      text.style.fontWeight = 'bold';
    }
  </script>
</body>
```

1- ما هي وظيفة JavaScript الأساسية في صفحات الويب؟

- (A) تنسيق النصوص والألوان.
- (B) إضافة التفاعل والديناميكية.
- (C) إنشاء قواعد البيانات.
- (D) تصميم الصور.

2- أي خاصية CSS تُستخدم لتغيير لون النص؟

- A) font-size
- B) color
- C) background-color
- D) text-align

3- ما هي الدالة التي تُستخدم لعرض رسالة تنبيه للمستخدم؟

- (A) print
- (B) alert
- (C) display
- (D) message

4- أي من الخيارات التالية يُستخدم لتغيير لون النص في عنصر HTML باستخدام JavaScript؟

- A) element.color = "red";
- B) element.style.color = "red";
- C) element.textColor = "red";
- D) element.fontColor = "red";

## الأجوبة ( كويز 1 )

- 1- تطوير الويب هو بناء وتصميم مواقع وتطبيقات تعمل على الإنترنت.
- 2- الواجهة الأمامية (Frontend): ما يراه المستخدم مثل الألوان والتصميم.  
الخلفية (Backend): ما يعمل خلف الكواليس مثل تخزين البيانات.
- 3- أشهر المتصفحات هي: Google Chrome ، Firefox ، Safari.
- 4- Visual Studio Code هو خيار ممتاز لأنه سهل الاستخدام ويدعم الإضافات.

## الأجوبة ( كويز 2 )

- 1- HTML هي لغة تُستخدم لبناء صفحات الويب وتنظيم محتواها.
- 2- HTML مهمة لأنها تُستخدم لتحديد هيكل الصفحة مثل النصوص، الصور، والروابط.
- 3- العناوين تبدأ من `<1h>` (الأكبر) إلى `<6h>` (الأصغر) لتقسيم النصوص.
- 4- 

```

```

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



## الأجوبة ( كويز 3 )

- 1- الإجابة: إضافة التفاعل والديناميكية.
- 2- الإجابة: color
- 3- الإجابة: alert
- 4- الإجابة: "element.style.color = "red";

- موقع : schools3w
- الرابط: <https://www.w3schools.com/>



شكرا لكم