

CSS SELECTORS

1. ****Simple Selectors and Combinators:****

(المحددين البسيطين والمشتقين): **Simple selectors and combinators.**

باستخدام `<p>` بناءً على نوعه. يمكن تحديد عنصر فقرة `<p>` HTML باستخدام تحديد عنصر **Simple Selector** (المحدد البسيط) - `<h1>` باستخدام المحدد البسيط `<h1>` وعنوان `<p>` المحدد البسيط.

بناءً على العلاقة بينه وبين عنصر آخر. هناك ثلاثة أنواع رئيسية HTML يستخدم لتحديد عنصر **Combinator** (المحدد المشتق) - من المشتقين:

يحدد عناصر `div p` يحدد عنصرًا داخل عنصر آخر. على سبيل المثال، **Descendant Combinator** (المشتق الفرعي) - `<div>` الموجودة داخل عناصر الديف `<p>` الفقرة.

يحدد `ul > li` يحدد عنصرًا مباشرًا متداخل في عنصر آخر. على سبيل المثال، **Child Combinator** (المشتق المباشر) - `` التي تكون مباشرة داخل عنصر القائمة الغير مرتبة `` عناصر القائمة.

يحدد عنصرًا يأتي مباشرة بعد عنصر آخر في نفس المستوى. على سبيل المثال، **Adjacent Sibling Combinator** (المشتق الجار) -

`<h1>` الذي يأتي مباشرة بعد عنصر العنوان `<p>` يحدد عنصر الفقرة `<h1 + p>`.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h2 {
      color: blue;
    }
  </style>
</head>
<body>
  <h2>This is a blue heading</h2>
  <h2>Another blue heading</h2>
</body>
</html>
```

- **Simple selectors are methods used to target elements in an HTML or XML document. For example, `p`**

- selects all `<p>` elements.

- Simple selectors are the most fundamental type of selectors.

- They target HTML elements based on their name or tag. For example, `h1` selects all `<h1>` elements.

- Simple selectors are essential for applying styles to specific types of elements across your web. pagepa

2. ****Class and ID Selectors:****

- Class selectors allow you to target a group of elements based on the class applied to them. For example, `

- ID selectors allow you to uniquely target a single element, such as `

- ****Class Selectors****:

- Class selectors are denoted by a dot (`. `) followed by the class name (e.g., `.highlight`).

- They allow you to target multiple elements with the same class attribute.

- Useful for applying consistent styles to related elements.

- Example: `

.highlight.

- ****ID Selectors****:

- ID selectors are denoted by a hash (`#`) followed by the ID name (e.g., `#header`).

- They target a single, unique element with a specific ID attribute, and IDs should be unique on a page.

- Useful for applying unique styles to specific elements.

- Example: `

#header.

`<p class="highlight">This is highlighted text.</p>`

`<p class="highlight">Another highlighted text.</p>`

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .highlight {
      background-color: yellow;
    }
    #header {
      font-size: 24px;
    }
  </style>
</head>
<body>
  <p class="highlight">This is hig
  <p class="highlight">Another hig
  <div id="header">This is a heade
</body>
</html>
```

.Class and ID selectors:(محدد الكلاس والمعرف)

-) Class Selector: `class` التي تحتوي على الكلاس المحدد. يتم تعريف الكلاس باستخدام السمة `class` في HTML. يستخدم لتحديد عناصر `example-class` التي تحتوي على الكلاس `example-class`. على سبيل المثال، `example-class` يحدد جميع العناصر التي تحتوي على الكلاس `example-class`.

-) ID Selector: `id` الذي يحتوي على المعرف المحدد. يتم تعريف المعرف باستخدام السمة `id` في HTML. يستخدم لتحديد عنصر `example-id` الذي يحمل المعرف `example-id`. على سبيل المثال، `example-id` يحدد العنصر الذي يحمل المعرف `example-id`.

CSS SELECTORS

3. ****Attribute Selectors:****

Common pseudo-elements include `::before` and `::after`, used for adding content before or after an element's content.

- Useful for adding decorative content or special styling to specific elements.
- Example: `p::before` adds content before all `

` elements.
- They are used to select elements based on the values of their attributes. For example, `input[type="text"]` selects all `` elements with a "text" type.
- Attribute selectors target elements based on their attributes and attribute values.
- They are enclosed in square brackets `[]` and can match specific attributes and values.
- Useful for styling elements with specific attributes.

- Example: `[target="_blank"]` selects elements with a "target" attribute set to "_blank". Attribute selectors: (محددات السمات)

أو `attribute` بناءً على السمة وقيمتها. يتم تحديد السمة باستخدام السمة `HTML` يستخدم لتحديد عناصر `Attribute Selector` - محدد السمة (- `text` وقيمتها "type" يحدد جميع العناصر التي تحتوي على سمة `text` على سبيل المثال، `attribute=value`]

4. ****Pseudo-Class Selectors:****

- They are used to target specific states or interactions of elements based on user actions. For instance, `a:hover` changes the formatting of links when the user hovers over them.
- Pseudo-class selectors target elements based on their state or position within the document.
- Common pseudo-classes include `:hover`, `:active`, `:focus`, `:first-child`, and `:nth-child`.
- Useful for creating interactive and state-dependent styles.
- Example: `a:hover` selects links when the user hovers over them.

محددات الصفات الزائفة (Pseudo-class selectors).

يتم تعيين المحدد الزائف باستخدام العلامة الزائفة ":" مع الاسم المناسب للحالة. على HTML يستخدم لتحديد حالة معينة أو حالة خاصة في عنصر (Pseudo-class Selector) المحدد الزائف (- يحدد العنصر الأول في قائمة العناصر المتداخلة. `first-child` يحدد العنصر عند تحويم المؤشر فوقه، و `:hover` سبيل المثال، `:

بعض الأمثلة على محددات الصفات الزائفة: -

يحدد العنصر عند تحويم المؤشر فوقه. `:hover` -

يحدد العنصر الأول في قائمة العناصر المتداخلة. `:first-child` -

يحدد العنصر الذي يأتي في الموضع الثاني أو الثالث أو الرابع إلخ في قائمة العناصر المتداخلة. `:nth-child(n)` -

```
<!DOCTYPE html>
<html>
<head>
  <style>
    a[target="_blank"] {
      text-decoration: underline;
    }
  </style>
</head>
<body>
  <a href="https://example.com" target="_blank">Example 1</a>
  <a href="https://example.org">Example 2</a>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
  <style>
    a:hover {
      color: red;
    }
  </style>
</head>
<body>
  <a href="#">Hover over me</a>
</body>
</html>
```

5. **Pseudo-Element Selectors:**

- They are used to target parts of an element itself, rather than selecting the whole element. For example, `p::before` allows you to add content before a paragraph element.
- Pseudo-element selectors target specific parts of an element's content, allowing you to style those parts separately.
- They are preceded by a double colon (`::`). Common pseudo-elements include `::before` and `::after`, used for adding content before or after an element's content.
- Useful for adding decorative content or special styling to specific elements.
- Example: `p::before` adds content before all `<p>` elements.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    h1::before {
      content: "Chapter ";
      font-weight: bold;
    }
  </style>
</head>
<body>
  <h1>Introduction</h1>
  <h1>Main Content</h1>
</body>
</html>
```

12.

محدد العنصر الزائفة (Pseudo-element selectors):

يتم تعيين المحدد الزائفة باستخدام اسم العنصر الزائفة بفصلتين HTML. يستخدم لتحديد جزء معين من عنصر (Pseudo-element Selector) المحدد الزائفة (Pseudo-element Selector).
يحدد الجزء بعد محتوى العنصر. `after` يحدد الجزء قبل محتوى العنصر، و `before` مزدوجتين `::`. على سبيل المثال، `::`.

بعض الأمثلة على محدد العنصر الزائفة:

- `::before`: يحدد الجزء قبل محتوى العنصر.

- `::after`: يحدد الجزء بعد محتوى العنصر.

CSS SELECTORS

Combinations

CSS selectors can be combined to create more specific and targeted styling rules. Here are some common examples of combining CSS selectors:

1. ****Combining Simple Selectors****:

- You can combine multiple simple selectors to target specific elements. For example:

```
h1.title {  
    /* Styles for h1 elements with class title */  
}
```

2. ****Combining Class and Element Selectors****:

- Combining class selectors with element selectors to target specific elements with a particular class:

```
div.sidebar p.highlight {  
    /* Styles for paragraphs with class highlight inside a sidebar div */  
}
```

3. ****Combining Class and Pseudo-Class Selectors****:

- Combining class selectors with pseudo-class selectors for more precise styling:

```
.button:hover {  
    /* Styles for buttons with class button when hovered */  
}
```

CSS SELECTORS

4. ****Combining Descendant Selectors****:

- Using a space between selectors to target elements that are descendants of another element. For example:

```
article p {  
    /* Styles for all paragraphs wit  
}
```

5. ****Combining Child Selectors****:

- Using the `>` symbol to target direct child elements. This is useful for styling specific child elements without affecting nested elements:

```
ul > li {  
    /* Styles for direct child list  
}
```

6. ****Combining Attribute and Element Selectors****:

- Combining attribute selectors with element selectors to target elements with specific attributes:

```
input[type="text"] {  
    /* Styles for input elements wit  
}
```

7. ****Combining Multiple Selectors****:

- Combining multiple selectors by separating them with commas to apply the same styles to different elements.

```
h2, h3, h4 {  
    /* Styles for h2, h3, and h4 ele  
}
```

8. ****Combining Class and Pseudo-Element Selectors****:

- Combining class selectors with pseudo-element selectors to add content before specific elements with a class:``

```
.quote::before {  
    content: "";  
}
```

