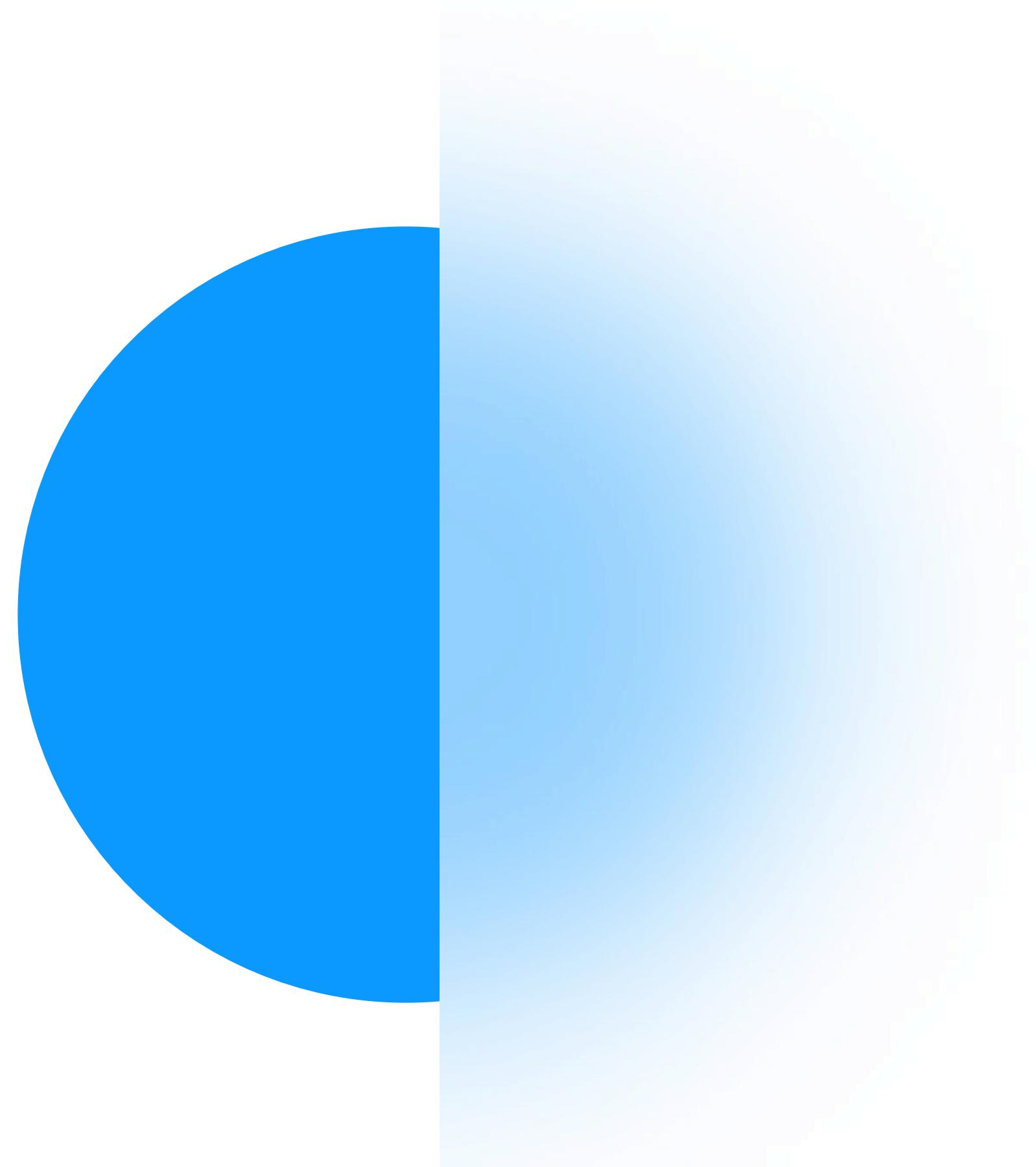


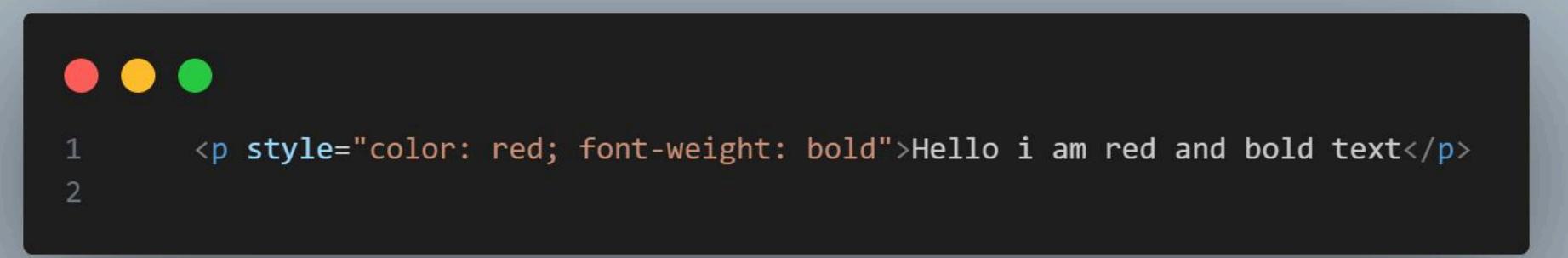
CSS (Cascading Style Sheets)

Lesson 1 Basics & Selectors



الموضوع	الوقت (التوقيت السوري)	التاريخ
HTML: Basics & Elements & page anatomy	12:00	2025\3\1
HTML: Basics check	12:00	2025\3\8
CSS: Basics & Selectors	12:00	2025\3\15
Typography & Colors	12:00	2025\3\22
Positions & :has() :not()	12:00	2025\3\29
Flexbox	12:00	2025\4\5
Grid	19:30	2025\4\7
States & Media Querie	19:30	2025\4\9
Animations	19:30	2025\4\11
JS: Basics & Events & First function (html+css+js)	19:30	2025\4\14
JS: Conditional Statements & Arrays & Loops	19:30	2025\4\16
JS: API & Fetch data (pokeapi)	19:30	2025\4\18
Last check	19:30	2025\4\20

1

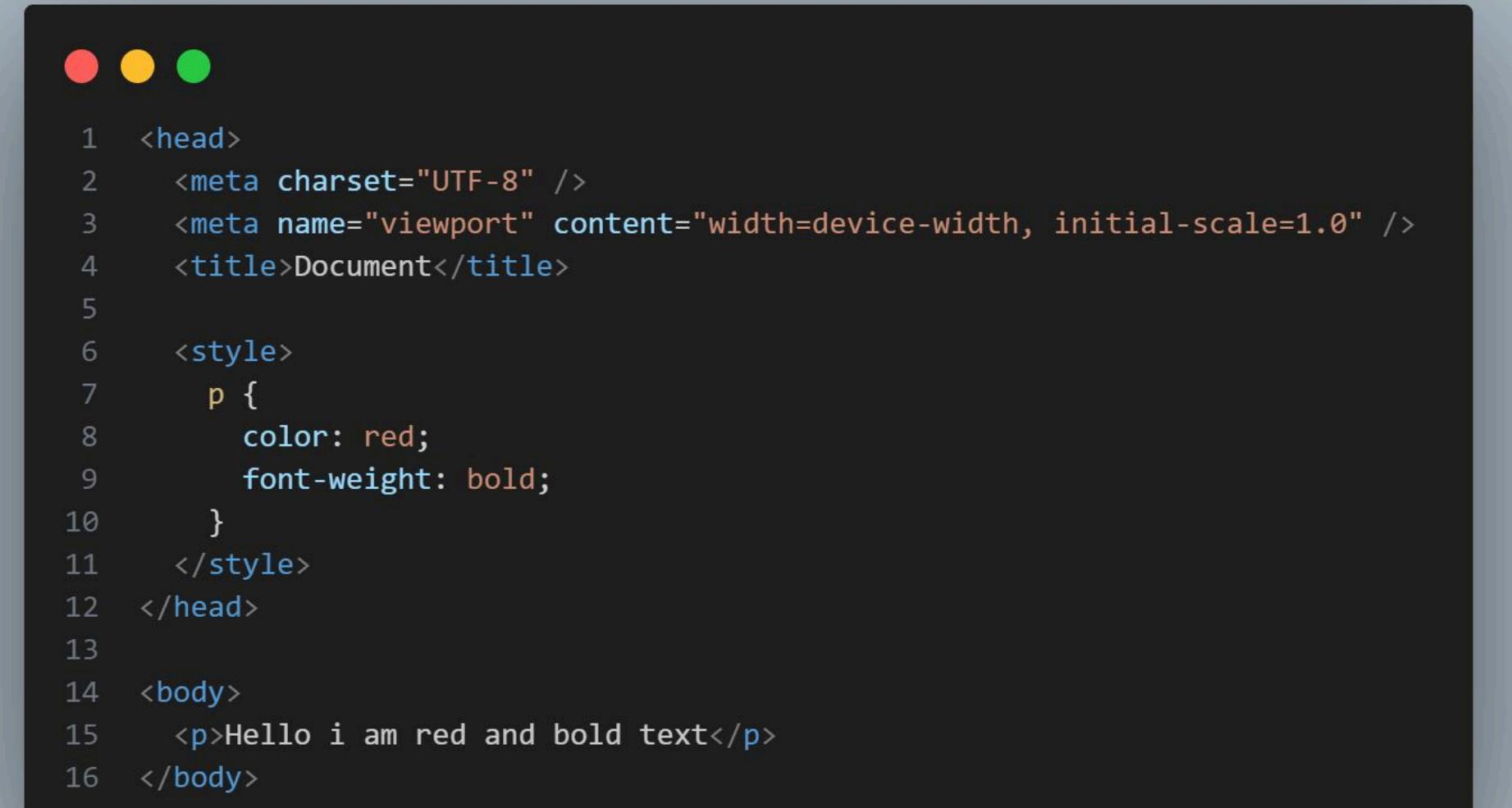


```

1 <p style="color: red; font-weight: bold">Hello i am red and bold text</p>
2

```

2



```

1 <head>
2   <meta charset="UTF-8" />
3   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
4   <title>Document</title>
5
6   <style>
7     p {
8       color: red;
9       font-weight: bold;
10    }
11  </style>
12 </head>
13
14 <body>
15   <p>Hello i am red and bold text</p>
16 </body>

```

3



```

1 <head>
2   <meta charset="UTF-8" />
3   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
4   <title>Document</title>
5
6   <link rel="stylesheet" href="cssFilelocation.css" />
7 </head>
8
9 <body>
10  <p>Hello i am red and bold text</p>
11 </body>

```

طرق تضمين الـ CSS في الـ HTML

1. بـ **Inline CSS**: تضمينها مباشرة داخل العنصر باستخدام `style attribute`

2. بـ **Internal CSS**: كتابة الأنماط داخل `<head>` ضمن `<style>`

3. بـ **External CSS**: ربط ملف خارجي يحتوي على الأنماط باستخدام `<link>`

ملاحظة مهمة: الطريقة 3 هي ماسيتم استخدامه في الكورس

التعامل مع بيئة الـ CSS

بنية الـ CSS: تتألف هذه البنية من مجموعة من القواعد (rules)، والتي تتتألف بدورها من مُحدد (selector) ثم قائمة بالخصائص والقيم المُسندة إليها: كتغيير القياس، نوع الخط، لون الخط، لون الخلفية، ترتيب العناصر داخل العنصر، شكل العنصر ... الخ

صيغة الكتابة (rule) :

{ القيمة : الخاصية } selector { property: value; }

تعريف أنواع المحددات (Selectors) :

1. عنصري (Element Selector): استهداف عنصر معين مثل {...}

2. فئوي (Class Selector): استخدام النقطة قبل اسم الفئة مثل {...}.navButton ويمكن ان يمتلك اكتر من عنصر نفس الكلاس.

3. معرفي (ID Selector): استخدام علامة الهاش قبل المعرف مثل {...}#toTop ويمكن لعنصر واحد فقط ان يمتلك اسم هويته ولا يمكن إعطائه لأكثر من عنصر.

- يتم ذكر الـ selectors بالترتيب من أول اب خارجي وصولا الى العنصر المحدد! كالمثال الجانبي.
- عند ذكر المحدد بشكل فردي فهذا يعتبر استهداف عام لكل عنصر بنفس الاسم/المحدد ولكن عند ذكر السلسلة العائلية وصولا الى العنصر فهذا يجعل الاستهداف اكثر تحديدا!

ملاحظة مهمة: لا انصح المبتدئين بالاعتماد على المحددات من فئة class & id كثيرا الا للضرورة او الحاجة لإنه من المهم التأقلم مع كيفية التعامل مع شجرة الـ DOM خلال المراحل الاولى من التعلم!

```

1 <body>
2   <header>
3     <button class="navBuuton">Open nav</button>
4     <nav>
5       <ul>
6         <li><a href="">Home</a></li>
7         <li><a href="">About</a></li>
8         <li><a href="">Contact</a></li>
9       </ul>
10      </nav>
11    </header>
12
13   <a href="" id="toTop">Back to the top</a>
14 </body>

```

1 استهدف كل "a" موجود داخل الملف

```

1 a {
2   color: red;
3   font-weight: bold;
4 }

```

1 استهدف كل "a" موجودة داخل السلسلة المذكورة

```

1 header nav ul li a {
2   color: red;
3   font-weight: bold;
4 }
5

```

2 استهدف كل عنصر يمتلك الصفة المذكور

```

1 .navBuuton {
2   width: 24px;
3   height: 30px;
4   background-color: red;
5 }

```

3 استهدف العنصر الذي يمتلك الهوية المذكورة

```

1 #toTop {
2   width: 24px;
3   height: 30px;
4   background-color: green;
5 }

```

نموذج الصندوق (Box Model)



مكونات الصندوق:

- المحتوى (Content): المساحة التي يحتويها النص أو الصورة.
- الحشوة (Padding): المسافة بين المحتوى وحدود العنصر.
- الحد (Border): الخط الفاصل حول الحشوة.
- الهامش (Margin): المسافة الخارجية التي تفصل العنصر عن العناصر الأخرى.

ملاحظة مهمة: • من الأفضل تحديد قياس صندوق العنصر عن طريق العرض و الارتفاع بدلا

عن طريق الـ padding

• لا نحتاج دائماً استخدام margin لتحديد مسافة فصل العنصر عن العناصر الأخرى.

• هنالك عناصر تمتلك padding & margin بشكل تلقائي نحتاج لزالتها بأكثر الأحيان او تغيير قيمتها مثل:
`<p>, , <button>, <h1>` و اخواتها ،

الخصائص الأساسية

الهوامش



```
1 margin:;  
2 padding:;
```

النصوص

```
1 color:;  
2 font-family:;  
3 font-size:;  
4 font-variant:;  
5 font-weight:;  
6 font-style:;  
7  
8 text-align:;  
9 text-decoration:;  
10 text-transform:;  
11 text-indent:;  
12 text-shadow:;  
13  
14 line-height:;  
15 letter-spacing:;  
16 word-spacing:;  
17 writing-mode:;
```

التحول و القياس



```
1 width: /* العرض */  
2 height: /* الارتفاع */  
3  
4 transform: /* تحويل العنصر */  
5 /* نقطة بداية التحويل */  
6 transform-origin: /* تحديد مبدأ التحويل */  
7 transition: /* تحديد مدة التحول */  
8  
9 list-style: /* تحديد نوع القائمة */  
10 box-shadow: /* تحديد ظل العنصر */  
11 filter: /* تحديد تأثير على العنصر */  
12 opacity: /* تحديد شفافية العنصر */  
13 visibility: /* تحديد رؤية العنصر */  
14 overflow: /* تحديد ماذا يحدث عند تجاوز العنصر للحدود */
```

الخلفيات



```
1 background-color:;  
2 background-image:;  
3 background-repeat:;  
4 background-position:;  
5 background-size:;
```

الحدود (الاطارات)



```
1 border:;  
2 border-radius:;
```

خصائص العرض المتقدمة



```
/* يجعل العناصر تتتدفق ضمن السطر وفقا للمحاور */  
1 display: flex; /*  
/* يجعل العناصر تتتدفق ضمن السطرين وفقا للشبكة */  
2 display: grid; /*
```

خصائص العرض



```
/* عنصر يشغل العرض الكامل ويسبب كسر السطر */  
/* عناصر تتتدفق ضمن السطرين دون كسر السطر */  
1 display: block; /*  
2 display: inline; /*  
3 display: inline-block; /* يجمع بين خصائص العنصرين */
```

سيقوم بتحديد اول عنصر من نوع p داخل الـ section ←

سيقوم بتحديد اول عنصر من نوع p داخل الـ section ←

سيقوم بتحديد ثاني عنصر من نوع p داخل الـ section ←

سيقوم بتحديد ثالث عنصر من نوع p داخل الـ section ←
حسب المثال لا يوجد عنصر م ثالث!

سيقوم بتحديد اخر عنصر من نوع p داخل الـ section ←

```

1  section p:nth-of-type(1){
2      color: red;
3  }
4
5  section p:first-of-type{
6      color: red;
7  }
8
9  section p:nth-of-type(2){
10     color: blue;
11 }
12
13 section p:nth-of-type(3){
14     color: green;
15 }
16
17 section p:last-of-type{
18     color: green;
19 }
```

سيقوم بتحديد كل ثاني عنصر من نوع p داخل الـ section ←

سيقوم بتحديد كل ثالث عنصر ابتداءً من العنصر
الثالث من نوع p داخل الـ section ←

سيقوم بتحديد كل ثالث عنصر من نوع p داخل الـ section ←

سيقوم بتحديد ثالث عنصر من نوع p داخل الـ section ←
لكن ابتداءً من الاسفل الى الاعلى

```

1  section p:nth-of-type(2n){
2      color: red;
3  }
4
5  section p:nth-of-type(2n+3){
6      color: blue;
7  }
8
9  section p:nth-of-type(odd){
10     color: green;
11 }
12
13 section p:nth-last-of-type(2){
14     color: yellow;
15 }
```

css selectors

nth-of-type()

يتيح لك محدد النوع nth-of-type: تحديد عنصر واحد أو أكثر بناءً على ترتيب

CSS Selectors المصدر الخاص به، وفقاً لصيغة. يتم تعريفه في مواصفات

على أنه "فئة شبه هيكلية"، مما يعني أنه يستخدم لتصميم المحتوى بناءً

على علاقته بالعناصر الأصلية والأشقاء.

```

1  <section>
2      <p>element 1</p>
3      <p>element 2</p>
4      
5  </section>
```

سيقوم بتحديد اول ابن من نوع p داخل الـ section ←

سيقوم بتحديد اول ابن من نوع p داخل الـ section ←

سيقوم بتحديد ثاني ابن من نوع p داخل الـ section ←

سيقوم بتحديد ثالث ابن من نوع p داخل الـ section ←

سيقوم بتحديد اخر ابن من نوع p داخل الـ section ←

حسب المثال لن يحدد شيء لأن الابن الثالث ليس p

سيقوم بتحديد كل ثاني ابن من نوع p داخل الـ section ←

سيقوم بتحديد كل ثالث ابن ابتداء من الابن الثالث من نوع p داخل الـ section ←

سيقوم بتحديد كل ثالث ابن من نوع p داخل الـ section ←

سيقوم بتحديد ثاني ابن من نوع p داخل الـ section ←
لكن ابتداء من الاسفل الى الاعلى

```

● ○ ●
1  section p:nth-child(1){
2      color: red;
3  }
4
5  section p:first-child{
6      color: red;
7  }
8
9  section p:nth-child(2){
10     color: green;
11 }
12
13 section p:nth-child(3){
14     color: blue;
15 }
16
17 section p:last-child{
18     color: yellow;
19 }

```

css selectors

nth-child()

يسمح لك محدد nth-child: بتحديد عنصر واحد أو أكثر استناداً إلى ترتيب

المصدر الخاص بهم.

```

● ○ ●
1  section p:nth-child(2n){
2      color: red;
3  }
4
5  section p:first-child(2n+3){
6      color: red;
7  }
8
9  section p:nth-child(odd){
10     color: green;
11 }
12
13 section p:nth-last-child(2){
14     color: blue;
15 }

```

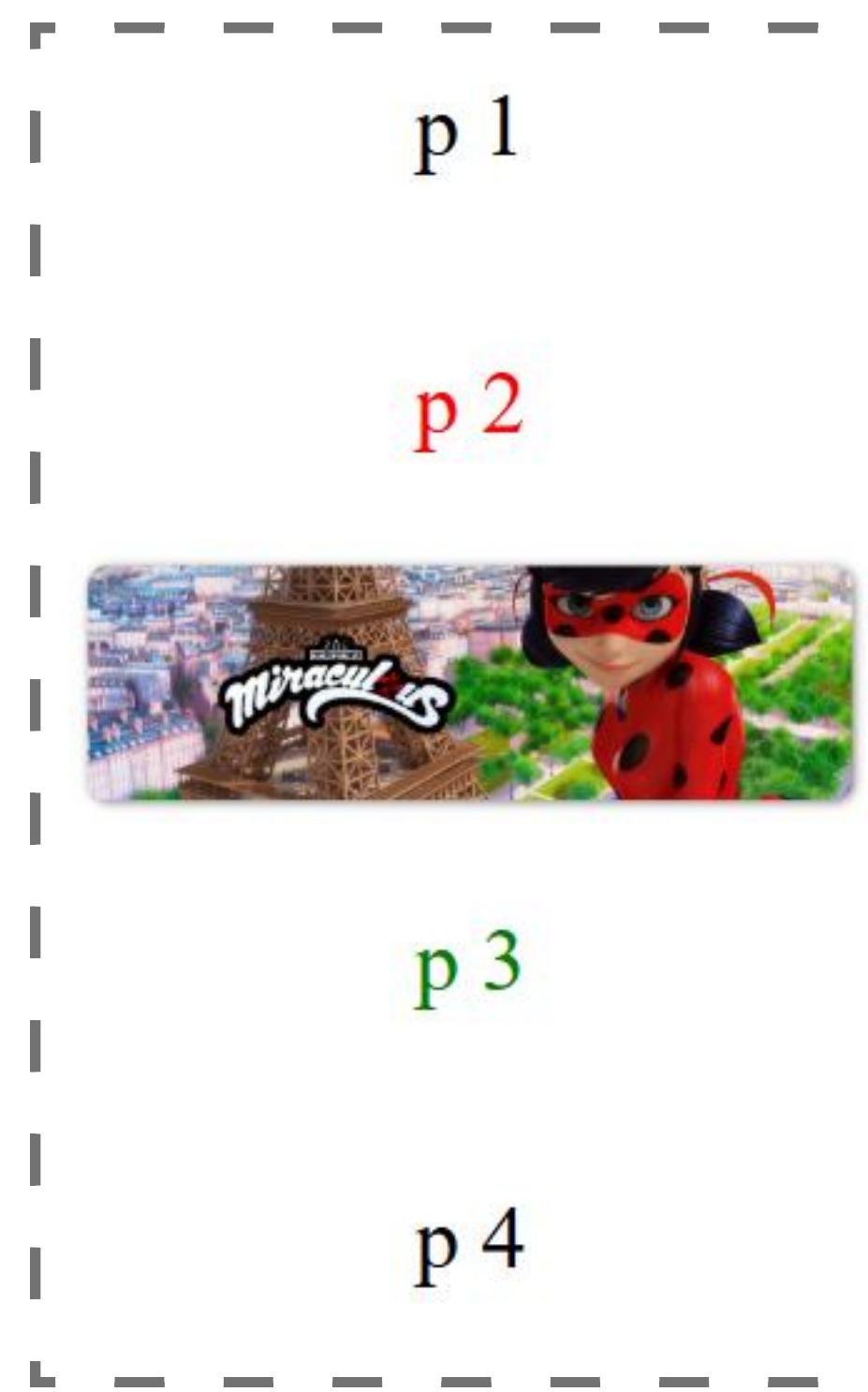
```

● ○ ●
1  <section>
2      <p>element 1</p>
3      <p>element 2</p>
4      
5  </section>

```

nth-child() VS nth-of-type()

Browser



CSS



```
1 section p:nth-of-type(2) {  
2   color: red;  
3 }  
4  
5 section p:nth-child(4) {  
6   color: green;  
7 }
```

HTML



```
1 <section>  
2   <p>p 1</p>  
3   <p>p 2</p>  
4     
5   <p>p 3</p>  
6   <p>p 4</p>  
7 </section>
```

يمكن استخدام لعبة [CSS Diner](#) للتدريب اكثر على هذه المحددات **ملاحظة مهمة:**

Understand '+', '>' and '~' symbols in CSS Selector

> Sign

إنه محدد فرعي، والذي يقوم باختيار عناصر فرعية مباشرة لعنصر رئيسي محدد.

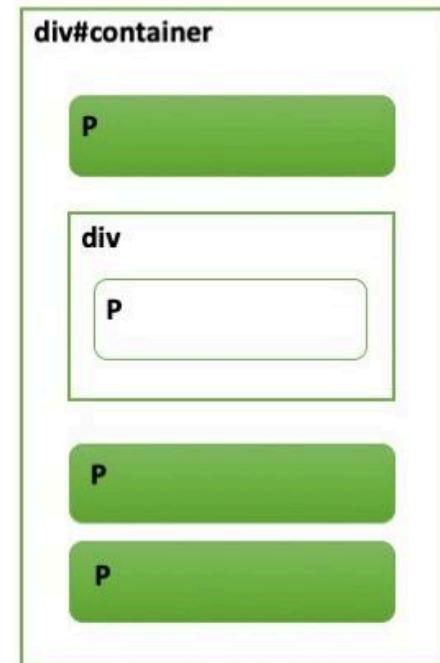
CSS child selector (>)
div#container > p

```

1 <div id="container">
2   ✓ <p>First</p>
3   <div>
4     ✗ <p>Child Paragraph</p>
5   </div>
6   ✓ <p>Second</p>
7   ✓ <p>Third</p>
8 </div>

```

TechBrij.com



سيتم استهداف جميع عناصر P التي تعد أبناءً مباشرين لـ div الحاوية، وليس أبناء div الطفل.

Space

إنه محدد النسل. يسمح لك بتصميم جميع عناصر الأبناء المتداخلة لعنصر الأصل، بغض النظر عن مدى عمقها في التسلسل الهرمي.

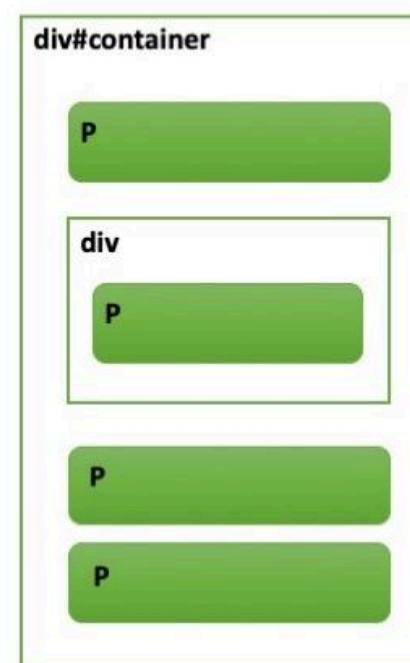
CSS descendant selector (Space)
div#container p

```

1 <div id="container">
2   ✓ <p>First</p>
3   <div>
4     ✓ <p>Child Paragraph</p>
5   </div>
6   ✓ <p>Second</p>
7   ✓ <p>Third</p>
8 </div>

```

TechBrij.com



سيتم استهداف جميع عناصر P التي تعد أبناءً لـ div الحاوية، أياً كان مكانها داخل الحاوية.

Understand '+', '>' and '~' symbols in CSS Selector

~ (tilde) Sign

إنه مُجَمّع أشقاء عام ومشابه لمُجَمّع الأشقاء المجاورين. والفرق هو أن المُحدّد الثاني لا يجب أن يتبع المُحدّد الأول مباشرةً، بل سيحدد جميع العناصر التي يسبقها المُحدّد السابق.

CSS general sibling selector (~)

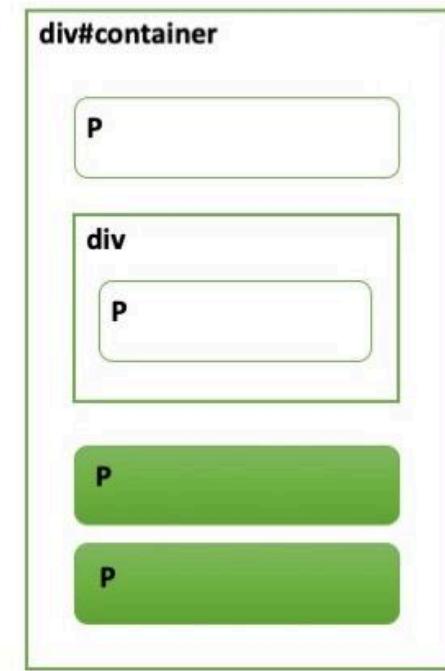
`div ~ p`

```

1 <div id="container">
2   <p>First</p>
3   <div>
4     <p>Child Paragraph</p>
5   </div>
6   <p>Second</p>
7   <p>Third</p>
8 </div>

```

TechBrij.com



سيستهدف جميع عناصر P التي تتبع div، أي الثاني والثالث.

+ (plus) Sign:

إنه عبارة عن مُجَمّع شقيق متجاور. فهو يجمع بين تسلسلين من المحددات البسيطة التي لها نفس الأصل، ويجب أن يأتي المحدد الثاني مباشرةً بعد المحدد الأول.

CSS adjacent sibling selector (+)

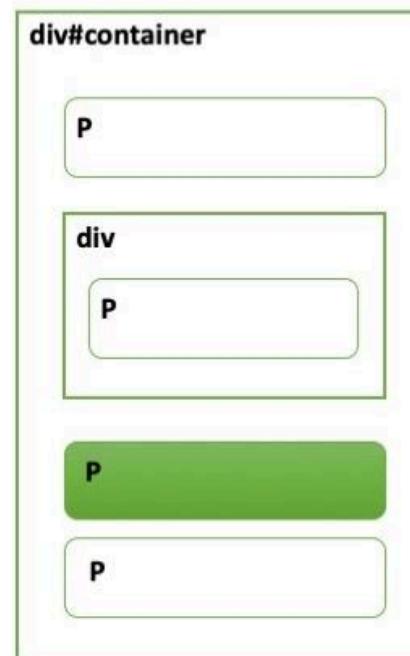
`div + p`

```

1 <div id="container">
2   <p>First</p>
3   <div>
4     <p>Child Paragraph</p>
5   </div>
6   <p>Second</p>
7   <*> <p>Third</p>
8 </div>

```

TechBrij.com



سيحدد فقط العنصر الأول الذي يسبقه مباشرةً المحدد السابق. في مثالنا، سيستهدف العنصر الثاني فقط لأن عنصر المالك P يأتي بعد علامة Div مباشرةً

selector[attribute= “value”]

حيث يتم تحديد جميع العناصر التي تحتوي السمة المحددة

مثال: `input[type="text"]{...}`

الذي يقوم بتحديد جميع حقول الإدخال `input` التي تحتوي على
نوع الإدخال "نص"

```
● ● ●  
1 <body>  
2   <label for="firstName">First Name</label>  
3   <input type="text" id="firstName" />  
4  
5   <label for="lastname">Last Name</label>  
6   <input type="text" id="lastname" />  
7  
8   <label for="email">Email</label>  
9   <input type="email" id="email" />  
10  
11  <label for="password">Password</label>  
12  <input type="password" id="password" />  
13 </body>
```

```
● ● ●  
1 input[type="text"] {  
2   background-color: red;  
3 }  
4  
5 input[type="password"] {  
6   background-color: blue;  
7 }
```

First Name	<input style="background-color: red; width: 100%; height: 20px;" type="text" value=""/>
Last Name	<input style="background-color: red; width: 100%; height: 20px;" type="text" value=""/>
Email	<input style="background-color: white; width: 100%; height: 20px;" type="text" value=""/>
Password	<input style="background-color: blue; width: 100%; height: 20px;" type="password" value=""/>

The end of CSS les 1