

# شروع کار با CSS

بخش هشتم: CSS Specificity

---

سید کاوه احمدی

■ المان p به چه رنگی نمایش داده خواهد شد؟

```
<html>
<head>
  <style type="text/css">
    #mainNote { color: Blue; }
    div .note { color: green; }
  </style>
</head>
<body>
  <div>
    <p id="mainNote" class="note">A Note!</p>
  </div>
</body>
</html>
```

# CSS Specificity

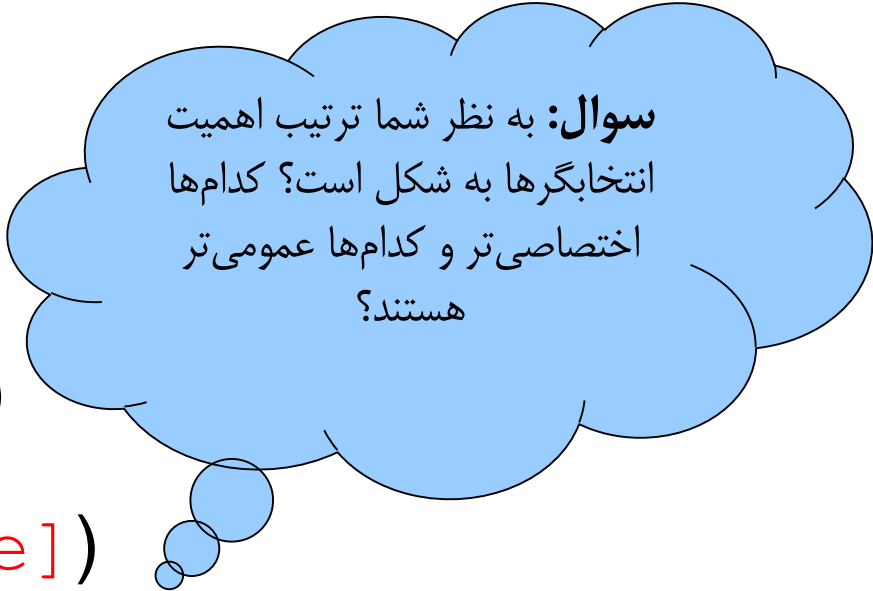
- یکی از مباحث مهم در CSS است.
- زمانی که به یک المان خاص دستورات متعدد CSS اعمال می شود، مرورگر کدام دستورات را اجرا می کند؟
- در این شرایط با مشخص کردن وزن هر دستور، دستورات با وزن بیشتر (دستوراتی که به شکل اختصاصی تر نوشته شده اند) اجرا می شوند.
- تعیین امتیاز (میزان اختصاصی بودن) برای انتخاب گرها یک قانون رسمی نیست و صرفاً روشی است برای درک آسان تر میزان اهمیت دستورات.

# یادآوری: انتخابگرهای CSS

- ID Selector (`#title`)
- Class Selector (`.alert`)
- Element Selector (`p span`)
- Attribute Selector (`p[title]`)
- Pseudo-classes (`a:hover`) المان را هدف قرار می‌دهند و کاری به محتوای آن ندارند
- Pseudo-elements (`p:first-line`) محتوای المان (زیر المان‌ها) را هدف قرار می‌دهند
- Style Attribute (`style="..."`)

# یادآوری: انتخابگرهای CSS

- ID Selector (`#title`)
- Class Selector (`.alert`)
- Element Selector (`p span`)
- Attribute Selector (`p[title]`)
- Pseudo-classes (`a:hover`)
- Pseudo-elements (`p:first-line`)
- Style Attribute (`style="..."`)



سوال: به نظر شما ترتیب اهمیت انتخابگرها به شکل است؟ کدامها اختصاصی تر و کدامها عمومی تر هستند؟

# میزان اختصاصی بودن انتخابگرها

- با وزن دادن به اجزای مختلف به کار رفته در انتخابگر و براساس یک عدد  $n$  رقمی عمل می‌کند.



اختصاصی ترین

عمومی ترین

- `*{}` specificity = 0000
- `li{}` specificity = 0001
- `li:first-line{}` specificity = 0002
- `ul li{}` specificity = 0002
- `ul ol+li{}` specificity = 0003
- `ul ol li.alert{}` specificity = 0013
- `li.alert.level{}` specificity = 0021
- `#title{}` specificity = 0100
- `style = "..."` specificity = 1000

```
<html>
<head>
  <style type="text/css">
    #title { color: red }
  </style>
</head>
<body>
  <p id="title" style="color: green">My Title</p>
</body>
</html>
```

- استایل ID: 0100
- استایل InLine: 1000



```
<html>
<head>
  <style type="text/css">
    div .x2 { color: green; }
    .x3 { color: blue; }
  </style>
</head>
<body>
  <div>
    <p class="x3">Text 1</p>
    <p class="x2 x3">Text 2</p>
  </div>
</body>
</html>
```

■ در P اولویتی مطرح نیست.

■ برای P دوم:

0011 ::x2 —

0010 ::x3 —

```
<html>
<head>
  <style type="text/css">
    p { color: red; }
    div p { color: blue; }
  </style>
</head>
<body>
  <p>Text 1</p>
  <div>
    <p>Text 2</p>
  </div>
</body>
</html>
```

- در P اولویتی مطرح نیست.
- برای P دوم:
  - p :0001
  - div p :0002

```
<html>
<head>
  <style type="text/css">
    div#foo .bar p#baz {color: blue;} /* 0212 */
    #baz {color:green;} /* 0100 */
    p#baz {color: red;} /* 0101 */
  </style>
</head>
<body>
  <div id="foo">
    <div class="bar">
      <p id="baz">
        The quick brown fox jumps over
      </p>
    </div>
  </div>
</body>
</html>
```

```
<html>
<head>
  <style type="text/css">
    div p.bio {color: red;}      /* 0012 */
    #sidebar p {color: blue;}   /* 0101 */
  </style>
</head>
<body>
  <div id="foo">
    <div id="sidebar">
      <p class="bio">text here</p>
    </div>
  </div>
</body>
</html>
```

## CSS Specificity - Cheat Sheet

	*	id	classes	elements	result*
	a	b	c	d	
li	0	0	0	1	1
.lorem	0	0	10	0	10
#ipsum	0	100	0	0	100
style = " "	1000	0	0	0	1000
ul li	0	0	0	2	2
ul ol + li	0	0	0	3	3
ul ol li.red	0	0	10	3	13
li:first-line	0	0	0	2	2
#ipsum li	0	100	0	1	101
a:link	0	0	10	1	11
<i>Legend:</i>					
a=1000	style = " "				<b>*Bigger is more specific</b>
b=100	ID attributes				
c=10	Classes and pseudo-classes				
d=1	Elements and pseudo-elements				

- فرض کنید دو دستور، به یک میزان اختصاصی باشند (امتیاز مشابهی داشته باشند).
- در این شرایط دستوری که دیرتر نوشته شده باشد قوی تر است.

```
p {  
  background: red;  
}
```

```
p {  
  background: blue;  
}
```

**WINNER!**

# محل قرارگیری کدها و تاثیر آن در قدرت اجرا

- کدها CSS داخلی، در حالی که امتیاز آنها مساوی دستورات CSS داخلی باشد، قوی تر محسوب می شوند.

## Style.CSS

```
#content h1 {  
    padding: 5px;  
}
```

```
<style type="text/css">  
#content h1 {  
    padding: 10px;  
}  
</style>
```

**WINNER!**

# !important

- این خصوصیت تمامی اولویت‌ها را باطل می‌کند و خودش اجرا می‌شود!
  - فرض کنید قدرت آن 10000 است.



# !important

```
<html>
<head>
  <style type="text/css">
    div#foo .bar p#baz {color: blue;} /* 0212 */
    #baz {color:green;} /* 0100 */
    p#baz {color: red;} /* 0101 */
    p {color: yellow !important;} /*10000 */
  </style>
</head>
<body>
  <div id="foo">
    <div class="bar">
      <p id="baz">
        The quick brown fox jumps over
      </p>
    </div>
  </div>
</body>
</html>
```

# نکات پایانی

- تعیین امتیاز (میزان اختصاصی بودن) برای انتخاب‌گرها یک قانون رسمی نیست و صرفاً روشی است برای درک آسان‌تر میزان اهمیت دستورات.
- — برای مثال امتیاز ۱۵ کلاس ۱۵۰ است و امتیاز یک آی دی ۱۰۰، اما همچنان ID اولویت بالاتری دارد و از ۱۵ کلاس هم قوی‌تر است!
- همیشه از حداقل تعداد سلکتورها استفاده کنید. نیازی نیست، در زمانی که لازم نباشد تعداد زیادی المان و کلاس را پشت هم ردیف کنید.
- با توجه به اینکه ID از کلاس اولویت بالاتری دارد، می‌توانید همیشه از کلاس استفاده کنید و ID را برای زمانی که بخواهید دستورات قوی بدهید کنار بگذارید.
- سعی کنید از `important` استفاده نکنید. استفاده از آن می‌تواند مشکلات جدی‌ای ایجاد کند. سعی کنید با اضافه کردن IDها به مقدار لازم دستور را سنگین کنید ولی دست به `important` نبرید.

## ■ عدد اولویت؟

- #footer \*:not(nav)
- ul > li ul li ol li:first-letter

## ■ کدام دستور اجرا می شود؟

```
p.content { color: red; }  
div .content { color: blue; }
```

```
<div>  
  <p class="content">  
    The quick brown fox jumps over  
  </p>  
</div>
```

- در مواردی که با CMSهایی نظیر جوملا، وردپرس و... کار می‌کنیم، ممکن است نتوانیم برخی استایل‌های از پیش نوشته شده را تغییر دهیم. اما می‌توانیم با نوشتن قوانین اختصاصی‌تر به هدف خود برسیم.

```
.main-content .block .content {  
  color:blue;  
}
```

```
<div class="page main-content">  
  <div id="block-system-main" class="block block-system">  
    <div class="region region-content">  
      <div class="content">  
        <p>Text 1</p>  
      </div>  
  
      <div class="content2">  
        <p>Text 2</p>  
      </div>  
    </div>  
  </div>  
</div>
```

## نکته اضافی

- فرض کنید با کد اسلاید قبل مواجه شده‌ایم.
  - چگونه می‌توانیم `Text 1` را بدون ایجاد تغییر در سایر بخش‌ها قرمز کنیم؟
  - استفاده از `.content` . ما را به هدف نمی‌رساند.
  - استفاده از `important`؟! راه راحت‌تری هست!
  - نوشتن انتخابگری با اولویت بالاتر!
- `#block-system-main .content`