

Specificity چیست؟

specificity به معنای مشخص بودن و دقیق بودن است. همانطور که قبلا گفته ایم در زبان CSS می توانیم برای یک عنصر واحد چندین قانون CSS بنویسیم که شاید حتی در تضاد هم باشند! اگر چنین اتفاقی رخ دهد مرورگر مجبور است که تعیین کند کدام دستور CSS باید روی عنصر اجرا شود. به این مفهوم specificity می گوییم. برای ساده تر شدن موضوع می توانیم از specificity به عنوان یک هرم یا رتبه بندی سلسله مراتبی یاد کنیم که در آن دستورات مختلف CSS جایگاه های مختلف و اولویت های مختلفی دارند و بر اساس همین اولویت ها روی عناصر اعمال می شوند. به طور مثال سلکتور سراسری (*) specificity بسیار کمی دارد در حالی که سلکتور های id اولویت specificity بالایی دارند. یکی از رایج ترین مشکلات در هنگام کدنویسی CSS نیز همین موضوع است.

در واقع specificity دارای چهار سطح مجزا است و هر کدام از این سطوح دارای اولویت خاص خودشان هستند (رابطه ی سلسله مراتبی):

- استایل های inline: همانطور که میدانید این نوع استایل ها مستقیما به عنصر هدفشان متصل هستند مانند

```
<h1 style="color: #ffffff">
```

- ID ها: ID ها وظیفه ی شناسایی عناصر مختلف را بر عهده دارند و نباید تکراری باشند (منحصر به عنصر خاصی هستند)

مانند `navbar#`

- کلاس ها، attribute ها و شبه کلاس ها: کلاس ها را که می شناسید، attribute ها را هم حتما می شناسید، شبه کلاس ها را

هم در قسمت های قبل به طور کامل توضیح دادیم (مثال های `hover:` و `focus:` و ...)

specificity را محاسبه کنید

برای جلوگیری از بروز مشکلات در کدهایتان بهتر است روشی را ابداع کنید تا بتوانید ترتیب اولویت ها (specificity) را به ذهن بسپارید. یکی از راه های پیشنهادی این است: از عدد ۰ شروع کنید. ۱۰۰۰ واحد برای attribute های استایل دهی در نظر بگیرید، سپس برای هر id دقیقا ۱۰۰ واحد اضافه کنید، برای هر attribute ، کلاس یا شبه کلاس ۱۰ واحد دیگر و برای هر عنصر یا شبه عنصر ۱ واحد اضافه کنید. به مثال زیر نگاه کنید:

A: h1

B: #content h1

C: <div id="content"><h1 style="color: #ffffff">Heading</h1></div>

specificity مربوط به A عدد ۱ است (چرا که یک عنصر است)، specificity مربوط به B عدد ۱۰۱ است (چرا که یک id و یک عنصر است) (و specificity مربوط به C عدد ۱۰۰۰ است) (چرا که استایل inline است). (از آنجایی که ۱ کوچکتر از ۱۰۱ و خود ۱۰۱ کوچکتر از ۱۰۰۰ است بنابراین دستور سوم) یعنی (C مقدار specificity بیشتری دارد و دستوری است که در نهایت روی عنصر اعمال می شود).

البته این تنها یک روش از هزاران روش است. ممکن است در اینترنت با روش های دیگری مواجه شوید (یا حتی با همین روش به صورت دیگری مواجه شوید) و بسته به سلیقه ی خود می توانید از آن استفاده کنید.

قوانین دیگر specificity

۱- اگر مقدار specificity برای دو دستور یکسان بود، آخرین دستور در سورس کد اعمال خواهد شد: اگر دو دستور مختلف CSS را برای یک عنصر نوشته باشیم، دستوری که در متن سورس کد از همه پایین تر باشد اولویت بیشتری خواهد داشت. مثال:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {background-color: yellow;}
h1 {background-color: red;}
</style>
</head>
<body>

<h1>This is heading 1</h1>

</body>
</html>
```

This is heading 1

۲- سلکتور های id اولویت بیشتری نسبت به سلکتور های attribute دارند: به کد ساده ی زیر نگاه کنید.

```
<!DOCTYPE html>
<html>
<head>
<style>
div#a {background-color: green;}
#a {background-color: yellow;}
div[id=a] {background-color: blue;}
</style>
</head>
<body>

<div id="a">This is a div</div>

</body>
</html>
```

This is a div

۳- سلکتور های نزدیک تر به عنصر اولویت بیشتری دارند: بر این اساس می توان گفت بین استایل های external (در یک فایل جداگانه CSS) و استایل های internal (در خود سند HTML)، استایل های internal برنده هستند:

```
4- From external CSS file:
5- #content h1 {background-color: red;}
6-
7- In HTML file:
8- <style>
9- #content h1 {
10-   background-color: yellow;
11- }
12- </style>
```

در مثال بالا خروجی ما رنگ yellow (زرد) خواهد بود چرا که در سند HTML است و به عنصر نزدیک تر است اما دستور مربوط به رنگ قرمز در یک فایل جداگانه قرار دارد.

۴- سلکتور های کلاس اولویت بیشتری نسبت به سلکتور های نام عنصر دارند: اگر از یک طرف سلکتور کلاسی مانند `intro` و از طرفی سلکتور نام عنصر مانند `div` یا `p` داشته باشیم، اولویت با سلکتور کلاس خواهد بود. به مثال زیر توجه کنید:

```
<!DOCTYPE html>
<html>
<head>
<style>
.intro {background-color: yellow;}
h1 {background-color: red;}
</style>
</head>
<body>

<h1 class="intro">This is a heading</h1>

</body>
</html>
```

This is a heading

۵- سلکتور سراسری و سلکتور ها یا مقادیری که به ارث رسیده اند، اولویت ۰ دارند: این مورد به نظر من واضح است؛ اگر سلکتوری مانند `* body` داشته باشید تنها در صورتی اعمال خواهد شد که هیچ استایل دیگری برای آن عنصر تعیین نشده باشد. همین مورد برای استایل هایی که از عناصر پدر یا موارد پیش فرض مرورگر می آیند نیز صدق می کند.