

# 스타일 위치에 따른 분류

html 문서에서 css를 작성하는 위치는 크게 3곳으로 나눌 수 있습니다.

- html 태그내에 스타일 지정
- 내부 스타일 시트
- 외부 스타일 시트

# css를 사용하는 방법 - html 태그내에 스타일 지정

css를 html 태그내에 지정할 수 있습니다. 이런 방식을 '스타일을 인라인으로 작성한다.' 라고 합니다.

```
<p style="color: red; font-size: 100px;">가나다라</p>
```

위의 p 태그는 style 속성을 통해 color, font-size를 정해졌습니다.

이런식으로 인라인 방식으로 스타일을 작성하는 방법은 유지보수가 어렵기 때문에 최대한 피해야할 방법입니다.

# css를 사용하는 방법 - 내부 스타일 시트

html 파일 안에 style 태그를 이용해 해당 문서내의 스타일을 지정해줄 수 있습니다.

```
<head>
  <style>
    h3 {
      color: blue;
      font-size: 50px;
      font-weight: normal;
    }
  </style>
</head>
<body>
  <h3>inline style sheet !!!</h3>
</body>
```

주로 style 태그는 head에 위치하게 됩니다.

이 방식도 html 안에 style, html 코드가 전부 들어가게 되면서 코드가 길어지기 때문에 유지보수가 힘들어지게 만듭니다. 그래서 현업에서는 잘 쓰이는 방식은 아닙니다.

# css를 사용하는 방법 - 외부 스타일 시트

css 파일을 분리시켜 style을 작성해주고 html에서 가져오는 방식입니다.

```
<head>
  <link rel="stylesheet" href="ex1.css">
</head>
<body>
  <div>this is external style !!!</div>
</body>
</html>
```

외부 css 파일을 생성해서 해당 파일에 스타일을 작성하고, html에서 link 태그를 사용해 css 파일을 가져오는 방식입니다. 목적에 따른 파일 분리가 가능하고, 코드를 파악하는것이 쉽기 때문에 주로 사용되는 방법입니다.

# CSS 문법

css를 작성하는 문법은 다음과 같습니다.

```
선택자 {  
  속성1: 값;  
  속성2: 값;  
  속성3: 값;  
}
```

- 선택자

스타일을 적용하기 위해선 어디에 적용할지를 명시해줘야 하는데 css선택자를 통해서 스타일 적용 위치를 알려줄 수 있습니다. 선택자의 문법은 조금 복잡하므로 조금 뒤에 다루겠습니다.

- 속성

해당 html 요소에 적용할 style 속성을 의미합니다. color, font-size, line-height 등등이 있습니다.

- 값

해당 속성에 적용할 값 입니다. 속성에 따라 사용할 수 있는 값들이 정해져 있습니다.

# 선택자 - tag

```
// html
<h1>this is h1 tag</h1>

// css
h1 {
  color: red;
  font-size: 20px;
}
```

- 태그 선택자

위의 코드는 h1 태그에 스타일을 적용하는 코드 입니다.

css에서 선택자로 태그명을 사용할 경우, 그냥 태그명을 사용하면 적용 가능합니다.

# 선택자 - id

```
// html
<h1 id="header-title">has header-title id</h1>

// css
#header-title {
  color: blue;
  font-size: 30px;
}
```

- id 선택자

html 태그에 id 속성이 정의되어 있는 경우, 스타일을 적용할 요소를 id를 통해 지정할 수 있습니다.

id를 통해 지정할 경우 css에선 id명칭 앞에 #을 붙여서 정의할 수 있습니다.

# 선택자 - class

```
// html
<h1 class="header-class">has header-class class</h1>

// css
.header-class {
  color: skyblue;
  font-size: 25px;
}
```

- class 선택자

html 태그에 class 속성이 정의되어 있는 경우, 속성을 적용할 요소를 class를 통해 지정할 수 있습니다.  
통해 지정할 경우 css에선 class명칭 앞에 .을 붙여서 정의할 수 있습니다.

보통 스타일을 지정할 때 class를 많이 사용하게됩니다.



# 선택자 - 여러 속성 사용

```
// html
<h2 class="multiple sub-header">it has multiple class</h2>

// css
.multiple.sub-header {
  color: orange;
  font-size: 25px;
}
```

하나의 html 태그에 여러개의 class가 지정될 수 있습니다.

여러개의 class를 지정하는 방법은 띄어쓰기를 이용해서 적용할 수 있습니다.

띄어쓰기의 활용방법이 html-class 와 css에서의 사용법이 다르기 때문에 이 부분은 잘 구분하셔야 합니다.

위의 h2 태그는 multiple, sub-header 두 가지 class가 적용되어 있습니다.

이렇게 여러가지 속성이 들어있는 태그를 불러오는 방법은 해당 선택자들을 이어붙여서 작성하면 됩니다.

현재 css에선 해당 요소를 가리키기 위해 **.multiple.sub-header**로 작성하고 있습니다.

만약 h2 태그까지 같이 명시하고 싶으면 **h2.multiple.sub-header**로 작성하시면 됩니다.

# 선택자 - 하위 요소 선택

```
// html
<header>
  <h1 class="in-the-header-h1">this is in the header</h1>
</header>

// css
header .in-the-header-h1 {
  color: yellow;
  font-size: 25px;
}
```

html 문서는 구조상 tree 구조를 가지고 있습니다.

이 때 css에서 html 태그 안에 있는 태그를 지정하고 싶을 경우 **띄어쓰기를 통해** 하위 요소를 지정할 수 있습니다.

예시로 든 html에선 header 태그안에 h1 태그가 들어있습니다.

이러한 구조에서 h1 태그를 지정하고 싶으면 **header h1** 이렇게 띄어쓰기로 하위 요소를 지정해줄 수 있습니다.

html에서의 띄어쓰기와 css에서의 띄어쓰기 역할이 다르므로 반드시 주의해서 기억해주세요.

위의 css에선 header안의 in-the-header-h1 클래스를 사용해서 해당 요소를 지정해줬습니다.

# css 우선순위

css에는 우선순위가 있습니다.

같은 요소를 가리키는 선택자라고 해도 어떤방식으로 작성되었는지 따라 적용되는 스타일이 다릅니다.

우선 선택자나 속성에 따른 우선순위는 다음과 같습니다.

- !important
- 인라인 스타일
- 아이디 선택자
- 클래스/속성/가상 선택자
- 태그 선택자
- 전체 선택자

그리고 css가 정의되는 위치나 css 작성 방법에 따라 우선순위가 바뀝니다.

# css 우선순위

```
// html
<header>
  <div class="title-wrapper">
    <h1 class="title">this is title</h1>
  </div>
</header>

// css
.title {color: red;}
.title-wrapper .title{ color: blue;}
header .title-wrapper .title { color: green;}
```

위의 예제에선 h1 태그가 header - div - h1 이 구조로 감싸져 있습니다.

h1태그에 대해서 스타일을 정의할 때 선택자를 작성하는 방법이 여러가지가 있습니다.

이 때 스타일 적용 순서는 선택자를 더 구체적으로 작성할수록 우선적으로 스타일이 정의됩니다.

## css 우선순위

```
// html
<main>
  <article>
    <section>
      <p>this is paragraph</p>
    </section>
  </article>
</main>

// css
main article section p {color: red;}
main article section p {color: blue;}
```

만약 같은 수준의 선택자가 작성되어 있다면 더 나중에 정의된 스타일이 해당 태그에 적용됩니다.