

CSS: Selectors&기본스타일

🕒 작성일시	@2022년 8월 1일 오후 2:49
☰ 내용	CSS 웹
▼ 주차	3주차
📎 자료	

CSS

Cascading Style Sheets

- 스타일을 지정하기 위한 언어

선택자(Selector)

```
h1 {  
  color: blue; 선언(Declaration)  
  font-size: 15px;  
}
```

속성 (Property) 값 (Value)

- CSS 구문은 선택자를 통해 스타일을 지정할 HTML 요소를 선택
- 중괄호 안에서는 속성과 값, 하나의 쌍으로 이루어진 선언을 진행
- 각 쌍은 선택한 요소의 속성, 속성에 부여할 값을 의미
 - 속성(Property): 어떤 스타일 기능을 변경할지 결정
 - 값(Value): 어떻게 스타일 기능을 변경할지 결정

CSS 정의 방법

- 인라인(inline)

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1 style="color: blue; font-size: 100px;">Hello</h1>
</body>
</html>

```

해당 태그에 직접 style 속성을 활용

- 내부참조(embedding) - <style>

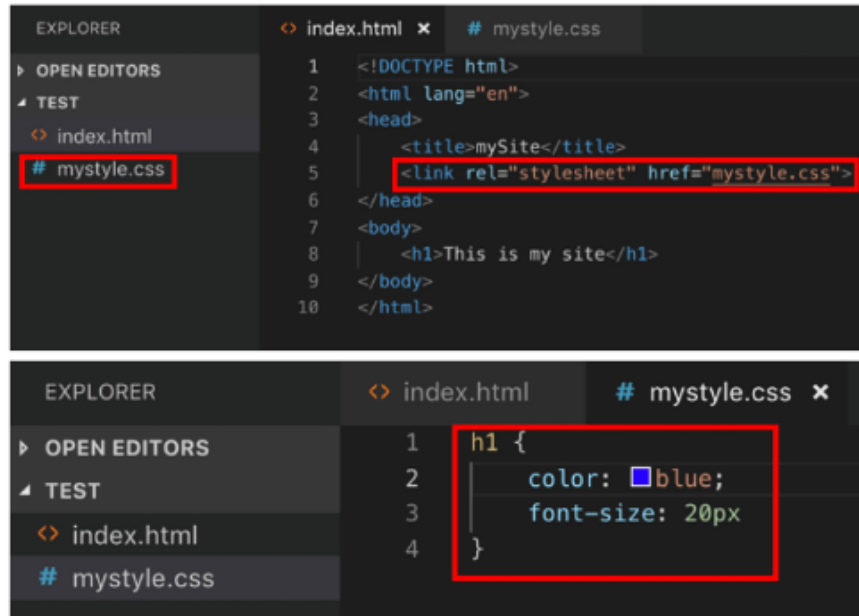
```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    h1 {
      color: blue;
      font-size: 100px;
    }
  </style>
</head>
<body>
</body>
</html>

```

<head> 태그 내에 <style>에 지정

- 외부 참조(link file) - 분리된 CSS 파일



외부 CSS 파일을 <head>내 <link>를 통해 불러오기

CSS with 개발자 도구

- styles: 해당 요소에 선언된 모든 CSS
- computed: 해당 요소에 최종 계산된 CSS

CSS Selectors

선택자(Selector) 유형

- 기본 선택자
 - 전체 선택자, 요소 선택자
 - 클래스 선택자, 아이디 선택자, 속성 선택자
- 결합자(Combinators)
 - 자손 결합자, 자식 결합자
 - 일반 형제 결합자, 인접 형제 결합자
- 의사 클래스 / 요소(Pseudo Class)
 - 링크, 동적 의사 클래스
 - 구조적 의사 클래스, 기타 의사 클래스, 의사 엘리먼트, 속성 선택자

CSS 선택자 정리

- **요소** 선택자
 - HTML 태그를 직접 선택
- **클래스(class)** 선택자 `class="<classname>"`
 - 마침표(.)문자로 시작하며, 해당 클래스가 적용된 항목을 선택
- **아이디(id)** 선택자 `#<idname>`
 - # 문자로 시작하며, 해당 아이디가 적용된 항목을 선택
 - 일반적으로 하나의 문서에 1번만 사용
 - 여러 번 사용해도 동작하지만, 단일 id를 사용하는 것을 권장

CSS 적용 우선순위 (cascading order)

- **범위가 좁을 수록** 우선순위

CSS 상속

- CSS는 상속을 통해 부모 요소의 속성을 자식에게 상속한다.
 - 상속 되는 것 예시
 - Text 관련 요소(font, color, text-align), opacity, visibility
 - 상속 되지 않는 것 예시
 - Box model 관련 요소(width, height, margin, padding, border, box-sizing, display), position 관련 요소(position, top/right/bottom/left, z-index)등

CSS 기본 스타일

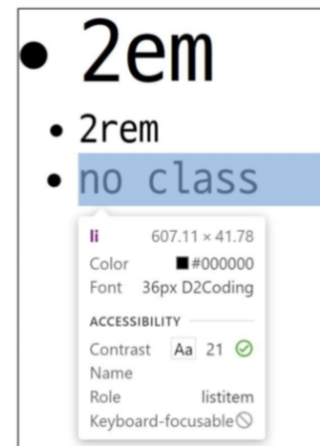
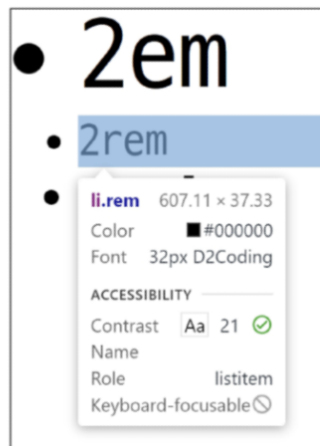
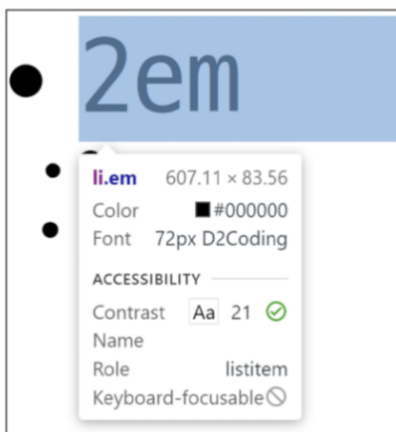
크기 단위

- px(픽셀)
 - 모니터 해상도의 한 화소인 '픽셀' 기준
 - 픽셀의 크기는 변하지 않기 때문에 **고정적인 단위**
- %
 - 백분율 단위
 - **가변적인 레이아웃**에서 자주 사용(모바일)
- em

- (바로 위, 부모 요소에 대한) 상속의 영향을 받음
- 배수 단위, 요소에 지정된 사이즈에 상대적인 사이즈를 가짐
- rem
 - (바로 위, 부모 요소에 대한) 상속의 영향을 받지 않음
 - 최상위 요소(html)의 사이즈를 기준으로 배수 단위를 가짐

```
<body>
  <ul class="font-big">
    <li class="em">2em</li>
    <li class="rem">2rem</li>
    <li>no class</li>
  </ul>
</body>
```

```
<style>
  .font-big {
    font-size: 36px;
  }
  .em {
    font-size: 2em;
  }
  .rem {
    font-size: 2rem;
  }
</style>
```



- viewport
 - 웹 페이지를 방문한 유저에게 바로 보이게 되는 웹 콘텐츠의 영역(디바이스 화면)
 - 디바이스의 viewport를 기준으로 상대적인 사이즈가 결정됨
 - 브라우저의 크기에 따라 크기가 변함
 - vw, vh, vmin, vmax

색상 단위

- 색상 키워드(background-color: red;)
 - 대소문자를 구분하지 않음
 - red, blue, black 과 같은 특정 색을 직접 글자로 나타냄
- RGB 색상(background-color: rgb(0, 255, 0);)
 - 16진수 표기법 혹은 함수형 표기법을 사용해서 특정 색을 표현하는 방식
- HSL 색상(background-color: hsl(0, 100%, 50%);)
 - 색상, 채도, 명도를 통해 특정 색을 표현하는 방식
- a
 - alpha(투명도)

```
p { color: black; }  
p { color: #000; }  
p { color: #000000; }  
p { color: rgb(0, 0, 0); }  
p { color: hsl(120, 100%, 0); }  
  
p { color: rgba(0, 0, 0, 0.5); }  
p { color: hsla(120, 100%, 0.5); }
```

모두 black !

CSS 문서 표현 - 추후에 하나씩

- 텍스트
 - 서체(font-family), 서체 스타일(font-style, font-weight 등)
 - 자간(letter-spacing), 단어 간격(word-spacing), 행간(line-height) 등
- 컬러(color), 배경(background-image, background-color)
- 기타 HTML 태그별 스타일링
 - 목록(li), 표(table)

CSS Selectors 심화

결합자(Combinators)

- **자손 결합자(공백)**
 - selectorA 하위의 모든 selectorB 요소

```
<style>
  div span {
    color: red;
  }
</style>
```

```
<div>
  <span>이건 빨강입니다.</span>
  <p>이건 빨강이 아닙니다.</p>
  <p>
    <span>이건 빨강입니다.</span>
  </p>
</div>
```

- **자식 결합자(>)**
 - selectorA 바로 아래의 selectorB 요소

```
<style>
  div > span {
    color: red;
  }
</style>
```

```
<div>
  <span>이건 빨강입니다.</span>
  <p>이건 빨강이 아닙니다.</p>
  <p>
    <span>이건 빨강이 아닙니다.</span>
  </p>
</div>
```

- 일반 형제 결합자(~)
 - selectorA의 형제 요소 중 뒤에 위치하는 selectorB 요소를 모두 선택

```
p ~ span {
  color: red;
}
```

```
<span>p태그의 앞에 있기 때문에 이걸 빨강이 아닙니다.</span>
<p>여기 문단이 있습니다.</p>
<b>그리고 코드도 있습니다.</b>
<span>p태그와 형제이기 때문에 이걸 빨강입니다!</span>
<b>더 많은 코드가 있습니다.</b>
<span>이것도 p태그와 형제이기 때문에 빨강입니다!</span>
```

- 인접 형제 결합자(+)
 - selectorA의 형제 요소 중 바로 뒤에 위치하는 selectorB 요소를 선택


```
p + span {  
  color: red;  
}
```

```
<span>p태그의 앞에 있기 때문에 이건 빨강이 아닙니다.</span>  
<p>여기 문단이 있습니다.</p>  
<span>p태그와 인접한 형제이기 때문에 이건 빨강입니다!</span>  
<b>더 많은 코드가 있습니다.</b>  
<span>p태그와 인접한 형제가 아니기 때문에 이건 빨강이 아닙니다.</span>
```