

# CSS 기초 다지기

# CSS

CSS describes how HTML elements should be displayed. - W3C -

- Cascading Style Sheets : 브라우저상에서 HTML 요소들이 시각적으로 어떻게 표현되는지를 정의
- 별도의 파일로 분리하여 모든 HTML 요소에 스타일링 가능 (external)
- HTML 요소에서 직접 스타일링 가능 (inline)

- CSS 예시

```
button {  
    color : red;  
}  
/*  
선택자 {  
    속성 : 값;  
}  
*/
```

- CSS 인라인 속성 예시

```
<p style="color:red;">이 문단은 빨간색입니다.</p>
```

# CSS Selectors

- 화면에 표시된 특정 요소를 접근하기 위해 필요한 문법

```
<section>
  <p>just paragraph</p>
  <p id="1">paragraph with id</p>
  <p class="2">paragraph with class</p>
</section>
```

- 태그 선택자 : 태그 { ... }
- 아이디 선택자 : #아이디 { ... }
- 클래스 선택자 : .클래스 { ... }

```
p { ... } /* p 태그 접근 */
#1 { ... } /* 아이디 값 1을 가진 태그에 접근 */
.2 { ... } /* 클래스 값 2를 가진 태그에 접근 */
```

```
<section>
  <p>1</p>
  <p>2</p>
  <div>
    <p>3</p>
  </div>
</section>
```

- `section p {}` : section 태그 밑의 모든 p 태그에 접근
- `section>p {}` : 바로 상위 태그가 section 인 p 태그에만 접근

```
section p { color : red };
section>p { color : black };
```

## CSS Pseudo-class

- 해당 요소의 **상태**에 따라 스타일 적용 가능

```
a {  
  color: red;  
}  
a:hover {  
  color: blue;  
}
```

# CSS Naming Convention - AirBnb

## 형식

- 들여쓰기는 스페이스 2칸 (= soft tab 1)
- 카멜케이스 기법보다는 `-` 사용
- id 선택자 사용 x
- 여러 선택자 사용시, 선택자 별로 줄 바꿈
- `{` 앞에 스페이스 한칸 사용
- `}` 는 새 줄에서 사용
- `:` 뒤에 스페이스 한칸 사용
- css 규칙 사이에 빈 1 줄 사용

## 나쁜 예시

```
.avatar{  
    border-radius:50%;  
    border:2px solid white; }  
.no, .nope, .not_good { // ... }  
#lol-no { // ... }
```

## 좋은 예시

```
.avatar {  
    border-radius: 50%;  
    border: 2px solid white;  
}  
  
.one,  
.selector,  
.per-line {  
    // ...  
}
```



## 주석

- 주석 줄은 전용으로만 사용 (다른 내용과 혼용하지 않기)

## 선택자

- id 로 요소를 선택하는 것은 안티패턴에 속한다.
- 재사용성이 떨어질 뿐만 아니라, 불필요하게 **specificity** 를 올림

## Border

- `border` 속성 값으로 none 보다는 0 사용

```
.bor {  
  /*border: none;*/  
  border: 0;  
}
```

## CSS Pre-processor

- 반복적인 코드를 줄이고 코드의 재사용성을 최대한으로 끌어올리기 위해 도입
- 기존 CSS 의 문법과 형식에 이어서 각 Pre-compiler 마다 추가적인 문법을 제공
- 잘 사용하면 CSS 의 코드량을 줄여주는 효과가 있다.
- 종류 : Sass, Less, Stylus 등

## Sass

- Ruby 기반의 전처리기이며, CSS 속성을 변수로 선언하여 사용할 수 있는 장점
- 그 외에 Nesting, Import, Extend, Inheritance 등의 기능을 제공
- css 변환을 위해 컴파일이 필요 - 커맨드 명령어나 gulp 와 같은 task manager 를 이용하여 컴파일
- 확장자 `.scss`

## Variable

- 속성 값을 변수로 지정하여 여러 군데 재사용 가능

```
/* CSS */
.title {
  color: #ff0000;
}
.content {
  color: #ff0000;
}

/* sass */
$color-red: #ff0000;
.title {
  color: $color-red;
}
.content {
  color: $color-red;
}
```

## Nesting

- HTML 처럼, 태그 간의 계층 관계가 더 뚜렷하게 보이게끔 스타일링이 가능

```
/* css */
ul { text-align: center; }
ul li {
  font-size: 14px;
}
ul a {
  color: blue;
}

/* sass */
ul {
  text-align: center;
  li {
    font-size: 14px;
  }
  a {
    color: blue;
  }
}
```

## Extend & Inheritance

- 다른 요소에 적용된 속성 값들을 그대로 재활용 가능하다.

```
ul {  
  padding: 0 10px 0 10px;  
}  
ol {  
  @extend ul;  
  color: red;  
}
```

## 참고

- [W3C - CSS](#)
- [Airbnb CSS Style Guide](#)
- [CSS Evolution](#)
- [Sass](#)



끝