

1. CSS (Cascading Style Sheets)의 등장

일반적인 CSS 적용시 하드코딩은 유지보수에 불합리하다.

```
<h1><a href="index.html"><font color="red">WEB</font></a></h1>
<ol>
  <li><a href="1.html"><font color="red">HTML</font></a></li>
  <li><a href="2.html"><font color="red">CSS</font></a></li>
  <li><a href="3.html"><font color="red">Javascript</font></a></li>
</ol>
```

1) 일괄 적용

<style> 태그안에서 CSS 를 작성한다.

<style> 태그안에서 a { } 라고 한다면,

선택자(selector)인 a 는 모든 <a> 태그를 의미한다.

선언(Declaration)인 color:red; 는 선택자에게 적용될 기능을 의미한다.

속성(Property)는 color 에 해당하고,

값(Value)는 red가 해당된다.

```
<style>
  a {
    color:red;
  }
</style>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <ol>
    <li><a href="1.html">HTML</a></li>
    <li><a href="2.html">CSS</a></li>
    <li><a href="3.html">JavaScript</a></li>
  </ol>
```

2) 개별 적용

필요한 부분에 직접 style="color:red" 작성해서 사용

```
<h1><a href="index.html">WEB</a></h1>
<ol>
  <li><a href="1.html">HTML</a></li>
  <li><a href="2.html" style="color:red">CSS</a></li>
  <li><a href="3.html">JavaScript</a></li>
</ol>
```

3) 정리

```
<!doctype html>
<html>
<head>
  <title>WEB1 - JavaScript</title>
  <meta charset="utf-8">
</head>
<style>
  a {
    color:blue;
  }
</style>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <ol>
    <li><a href="1.html">HTML</a></li>
    <li><a href="2.html" style="color:red">CSS</a></li>
    <li><a href="3.html">JavaScript</a></li>
  </ol>
  <h2>JavaScript</h2>
  <p>
    JavaScript
  </p>
</body>
</html>
```

WEB

1. [HTML](#)
2. [CSS](#)
3. [JavaScript](#)

JavaScript

JavaScript

2. CSS 의 속성(Property) 은 개별 공부로

<https://developer.mozilla.org/ko/docs/Web/CSS/Reference>

<https://www.w3schools.com/cssref/index.php>



3. CSS 의 선택자(selector)

```
<!doctype html>
<html>
<head>
  <title>WEB1 - JavaScript</title>
  <meta charset="utf-8">
</head>
<style>
  a {
    color:black;
    text-decoration: none;
  }
  .saw {
    color:gray;
  }
  .active{
    color: red;
  }
  h1 {
    font-size: 45px;
    text-align: center;
  }
</style>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <ol>
    <li><a href="1.html" class="saw">HTML</a></li>
    <li><a href="2.html" class="saw active">CSS</a></li>
    <li><a href="3.html">JavaScript</a></li>
  </ol>
  <h2>JavaScript</h2>
  <p>JavaScript</p>
</body></html>
```

WEB

1. HTML
2. CSS
3. JavaScript

JavaScript

JavaScript

1) 태그 선택자 : 가장 기본적인 선택자로 태그명 { } 의 형태를 가진다. ex) a { }

2) class 선택자 : . dot 은 class 를 의미하며 .className { } 형태이다. ex) .saw { } , .active { }
.saw 는 **class="saw"** 가 적용된 모든 곳에 프로퍼티와 벨류인 **color:gray;** 를 적용한다.

2-1) 다중 class 선택자 가능 : class 는 여러개를 적용할 수 있다. 띄워쓰기로 구분된다.

class="saw active"

Cascading의 특징상 .saw 보다 후술된 .active가 우선순위가 높다.

3) id 선택자 : #idName{ }

id는 가장 높은 우선순위를 가진다.

.saw 보다 앞에 작성되었지만 id로 선언되면 최우선순위다.

```
<!doctype html>
<html>
<head>
  <title>WEB1 - JavaScript</title>
  <meta charset="utf-8">
</head>
<style>
  a {
    color:black;
    text-decoration: none;
  }
  #active{
    color: red;
  }
  .saw {
    color:gray;
  }
  h1 {
    font-size: 45px;
    text-align: center;
  }
</style>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <ol>
    <li><a href="1.html" class="saw">HTML</a></li>
    <li><a href="2.html" class="saw" id="active">CSS</a></li>
    <li><a href="3.html">JavaScript</a></li>
  </ol>
  <h2>JavaScript</h2>
  <p>
    JavaScript
  </p>
</body>
</html>
```



우선순위는 다음과 같다.

- 태그 선택자 < class 선택자 < id 선택자
- 동일한 등급의 선택자라면 후술되는 선택자가 높다.
- 잘 생각해보면 태그가 가장 포괄적으로 적용되고, class 는 그 다음, id 는 단일로만 적용이다.

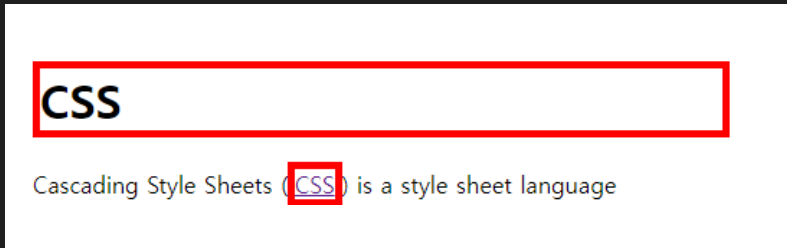
4. 박스모델

1) 태그의 기본 적용 범위

각 태그마다 기본값으로 적용 범위, 영역의 차이가 있다.

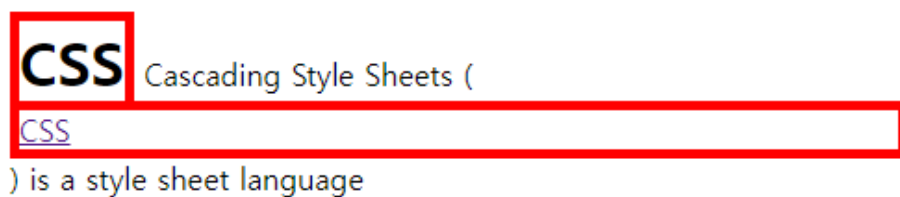
전체를 쓴다면 **block level element**, 자기크기만큼을 사용하면 **inline element** 이다.

```
<!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>
    /* block level element : h1 태그는 블록 단위(화면 전체)를 사용 */
    h1 {
      border-width: 5px;
      border-color: red;
      border-style: solid;
    }
    /* inline element : a 태그는 자산의 크기만큼을 사용 */
    a {
      border-width: 5px;
      border-color: red;
      border-style: solid;
    }
  </style>
</head>
<body>
  <h1>CSS</h1>
  Cascading Style Sheets is a style sheet language
</body>
</html>
```



2) 태그의 적용 범위 변경

속성으로 **display:inline;** 이나 **display:block;** 를 사용하면 적용범위가 달라진다.



3) 동일한 속성과 값을 가진 태그라면 , 콤마로 나열

```
<style>
  h1 , a {
    border-width: 5px;
    border-color: red;
    border-style: solid;
    display:inline;
  }
</style>
```

4) 동일한 속성명은 나열이 가능

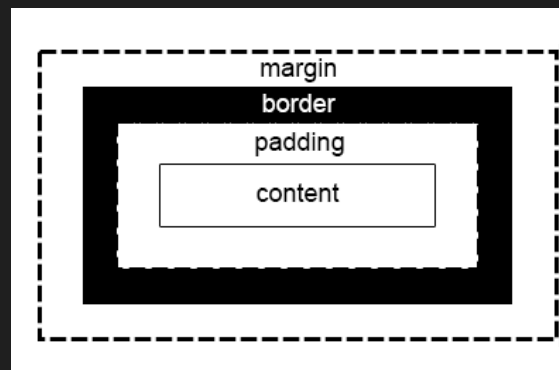
width, color, style 등의 순서는 상관없이 나열할 수 있다.

```
<style>
  h1 , a {
    border: 5px solid red;
  }
</style>
```

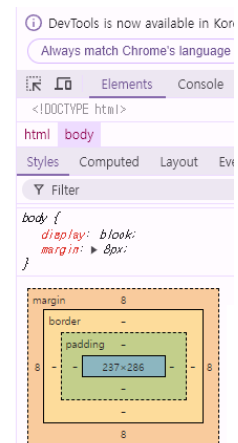
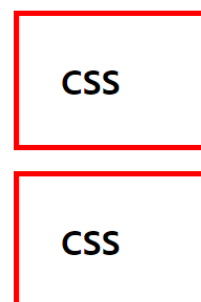
5) padding, margin

```
<style>
  h1 , a {
    border: 5px solid red;
    padding: 40px; /* 내부여백*/
    margin: 20px; /* 외부여백*/

    /* h1의 기본 display는 block 이다.
    여기에 넓이를 지정하면 박스형태가 된다.*/
    display:block;
    width: 100px;
  }
</style>
```



개발자 도구(F12)에서 Elements를 활용하자.



5. 박스모델 활용

```
<style>
  a {
    color:black;
    text-decoration: none;
  }
  #active{
    color: red;
  }
  .saw {
    color:gray;
  }
  h1 {
    font-size: 45px;
    text-align: center;
    border-bottom: 1px solid gray;
    margin: 0px;
    padding: 20px;
  }
  /* border-right 만 준다면 우측끝에 선이 생기므로 width로 박스크기를 제한한다. */
  ol {
    border-right: 1px solid gray;
    width: 100px;
  }
</style>
```

WEB

1. HTML
2. CSS
3. JavaScript

JavaScript

JavaScript

1) Web 아래의 선은 **border-bottom: 1px solid gray;**로 밑줄만 남긴 것이다.

2) ol에 border-right만 준다면 우측끝에 선이 생기므로 width로 박스크기를 제한한다.

```
ol {
  border-right: 1px solid gray;
  width: 100px;
}
```

3) ol의 기본 속성에는 패딩이 적용되어 있다. 이를 세부 수정하면 더 깔끔해진다.

```
ol {
  border-right: 1px solid gray;
  width: 100px;
  margin: 0;
  padding: 20px;
}
```

3-1) 기본 이의 값



3-2) 변경해서 조절

