

# CSS

- 우리는 HTML을 이용하여 홈페이지를 만들어 봤습니다.
- 그러나, 우리는 현재의 홈페이지에 만족하지 못하였을 것 입니다.
- 그 이유는 별로 아름답지 않다! 입니다.
- 사람들은 더 이쁜 폰트와 조화로운 배경으로 구성되어 있는 시각적으로 매력적인 홈페이지를 원할 것 입니다.
- 이를 해결하기 위해서 등장한 개념이 CSS 입니다.
- 앞서 말했듯이 HTML이 책의 내용에 관련있는 것 이라면 CSS는 책의 디자인과 관련있는 것 입니다.
- 물론 CSS등장 이전에 우리는 HTML의 다중 속성을 이용해서 문제를 해결할 수도 있었습니다.
- 먼저 이 방식을 알아보겠습니다.

## 쉽지만 한계가 있는 방식 (CSS 등장 이전의 상태)

- `<font>` tag

```
<font color="red">Hello</font>
```

Hello

- 우리는 이 `font tag` 를 사용하여 HTML에 색을 넣을 수 있습니다.
- 그러나 우리는 HTML이 순수히 정보에 해당하는 영역만을 다루는 것을 배웠습니다.
- 그러나 실제로 `font` 태그는 그 정보와 직접적인 관련이 없습니다.
- 시각장애인이 보았을때 의미가 없는 것은 우리가 정보가 아니라고 이해하면 쉽습니다.
- 즉, 우리는 정보에 해당하는 내용만 HTML에 나타내야 하며 그 외의 것들은 CSS를 사용해야 합니다.

```
<!doctype html>
<html>
<head>
  <title>WEB - CSS</title>
  <meta charset="utf-8">
</head>
<body>
  <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>
  <h2>CSS</h2>
  <p>
    Cascading Style Sheets (CSS) is a style sheet language used for describing the
  </p>
</body>
</html>

```

- 이렇듯 웹이 등장한 이후 HTML에는 `font` 와 같은 디자인을 위한 코드가 대거 추가 됩니다.
- 무분별하게 추가된 디자인 기능은 정보로서의 웹이라는 가치를 오히려 퇴보시킵니다.
- 따라서 이를 극복하기 위해서 웹을 만드는 사람들은 디지털 정보의 세계를 완전히 바꿔놓을 기술을 생각해 내기 시작합니다.

## CSS 등장

### 우리가 사용해야 하는 방식 (CSS)

- CSS 는 앞에서 사용하는 방식 보다는 어렵지만 근본적으로 문제를 해결하는 방식 입니다.

```

<li><a href="1.html"><font color="red">리버풀</font></a></li>
<li><a href="1.html"><font color="red">버트런드 러셀</font></a></li>
<li><a href="1.html"><font color="red">고레에다 히로카즈</font></a></li>

```

- 리버풀
- 버트런드 러셀
- 고레에다 히로카즈
- 우리는 먼저 앞에서 사용한 방식을 작동하지 않게 만들어야 합니다.
- 프로그래밍에서 이러한 기능을 Comment라고 합니다.
- Comment 안에 있는 내용은 컴퓨터는 읽지 않고 무시합니다.

```

<!--
여기에 있는 내용은 컴퓨터가 무시합니다.
-->

```

```

<!--
<ol>
  <li><a href="1.html"><font color="red">리버풀</font></a></li>
  <li><a href="1.html"><font color="red">버트런드 러셀</font></a></li>
  <li><a href="1.html"><font color="red">고레에다 히로카즈</font></a></li>
</ol>
-->

```

- 그럼 우리가 CSS를 사용하여 `font` tag 를 사용하지 않고 앞의 내용을 붉은색으로 표시하겠습니다.

```

<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>

    <style>
    a{
      color:red;
    }
    </style>

  </head>
  <body>
    <ol>
      <li><a href="1.html">리버풀</a></li>
      <li><a href="2.html">버트런드 러셀</a></li>
      <li><a href="3.html">고레에다 히로카즈</a></li>
    </ol>
  </body>
</html>

```

- 우리는 CSS 코드를 다음과 같이 `<head>` 태그 안에 넣어줌으로써 이 문제를 해결 할 수 있습니다.
- 그럼 잠깐 이 것의 효과를 상상해 봅시다.
- 우리가 좋아하는 것들이 100개가 넘어서 웹페이지 모두의 색을 바꾼다고 해 봅시다.
- 앞서 배운 `<font>` 를 사용하면 엄청난 노동을 해야 하지만, CSS를 사용하면 웹의 사이즈와 상관없이 한 번의 작업으로 이것을 수행할 수 있습니다.
- 이것은 엄청난 효과 입니다!!
- 우리가 다음에 배울 프로그래밍 언어인 파이썬에서도 이 중복의 제거는 매우매우 중요한 특징 입니다.
- 우리는 이렇게 CSS를 사용하면 웹페이지의 유지 보수가 훨씬 쉬워지고 HTML만을 사용하는 디자인 방

법보다 효율적임을 공감하면 좋겠습니다.

- 앞으로 우리는 이러한 CSS를 사용하여 디자인과 관련된 내용은
- HTML은 너무 중요하기 때문에 HTML로부터 디자인에 대한 기능을 빼온 것 입니다.
- 그리고 훨씬 효율적 입니다.
- 이렇게 해서 우리는 CSS가 등장하게 된 배경을 살펴 보았습니다.

```
<!doctype html>
<html>
<head>
  <title>WEB - CSS</title>
  <meta charset="utf-8">
  <style>
    a {
      color:black;
    }
  </style>
</head>
<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>
  <h2>CSS</h2>
  <p>
    Cascading Style Sheets (CSS) is a style sheet language used for describing the
  </p>
</body>
</html>
```

Resources

## CSS 기본 문법

- 우리는 먼저 HTML 안에 CSS 코드를 넣어야 한 다는 것을 통해서 컴퓨터가 HTML을 읽어나가면서 어디부터 어디까지가 HTML의 내용이고 어디부터 어디까지는 CSS코드인지를 알려줄 필요가 있습니다.
- `<style>` 태그가 그 역할을 합니다.
- HTML은 이 `<style>` 태그를 읽으면 그 안에 들어있는 내용은 CSS로 인식합니다.
- 이 방식 말고 또다른 한 가지 방식이 더 있습니다.

- HTML 태그 안에 속성으로 사용하는 것 입니다.
- `style="color:red"`

```
<li><a href="1.html" style="color:red">리버풀</a></li>
```

## • 리버풀

- 다음과 같이 우리는 HTML 태그 안에 style 속성을 이용하여 CSS를 작성할 수 있습니다.
- 즉 속성으로 style이 등장하면 해당 태그 안에 있는 내용에 적용이 됩니다.

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>

    <style>
      a{
        color:blue;
        text-decoration: none;
      }
    </style>

  </head>
  <body>
    <ol>
      <li><a href="1.html" style="color:red;text-decoration:underline">리버풀</a></li>
      <li><a href="2.html">버트런드 러셀</a></li>
      <li><a href="3.html">고레에다 히로카즈</a></li>
    </ol>
  </body>
</html>
```

- 우리는 웹페이지에 CSS를 삽입하는 방법을 알아보았습니다.
- style 태그와 style 속성을 사용하는 방식 (2가지 방식)
- 여러가지 효과를 삽입하기 위해 이를 구분하는 세미콜론 ; 이 필요합니다.

## Style 태그 구성요소



## CSS 속성(Property)을 알아내는 방법

- 우리의 홈페이지의 큰 제목을 더 크게 만들고 가운데 정렬해보고싶다? (h1 tag)



css text size property



```
h1 {  
  font-size: 50px;  
}
```



css text center property



```
h1 {  
  font-size: 50px;  
  text-align: center;  
}
```

- 우리는 모든 속성을 암기할 필요 없이, 검색 엔진의 도움을 얻어서 해결할 수 있습니다.

```
<!doctype html>
```

```

<html>
<head>
  <title>WEB - CSS</title>
  <meta charset="utf-8">
  <style>
    a {
      color:black;
      text-decoration: none;
    }
    h1 {
      font-size:45px;
      text-align: center;
    }
  </style>
</head>
<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;text-decoration:underline">CSS</a></li>
    <li><a href="3.html">JavaScript</a></li>
  </ol>
  <h2>CSS</h2>
  <p>
    Cascading Style Sheets (CSS) is a style sheet language used for describing the
  </p>
</body>
</html>

```

Resources

## CSS 선택자(Selector)를 스스로 알아내는 방법 (선택자의 종류)

- 우리가 다음과 같은 특징을 같은 웹 페이지를 만든다고 생각해 봅시다.
  - 모든 하이퍼링크는 기본적으로 검색은 + 밑줄 없음
  - 우리가 들어가본 페이지는 회색
  - 우리가 현재 보고있는 페이지는 붉은색
- 우리가 지금까지 배운 방식으로는 해결 할 수 없습니다.
- 이를 해결 하기 위해서는 우리는 class라는 개념을 이용해야 합니다.

```

<style>
  a {
    color:black;
    text-decoration: none;
  }

```

```
h1 {
  font-size: 50px;
  text-align: center;
}
</style>
```

```
<li><a href="1.html" class ="saw">리버풀</a></li>
<li><a href="2.html" class ="saw active">버트런드 러셀</a></li>
<li><a href="3.html" >고레에다 히로카즈</a></li>
```

```
<style>
a {
  color:black;
  text-decoration: none;
}

.saw{
  color:gray;
}

.active{
  color:red;
}

h1 {
  font-size: 50px;
  text-align: center;
}
</style>
```

- 클래스 속성의 값은 여러개가 가능, 띄어쓰기로 구분한다.
- 한 개의 태그에는 여러개의 속성이 들어갈 수 있다.
- 그러나 이러한 방식은 코드의 순서에 영향을 받음.

```
<style>
a {
  color:black;
  text-decoration: none;
}

.active{
  color:red;
}

.saw{
```



```

    color:gray;
}

h1 {
    font-size: 50px;
    text-align: center;
}
</style>

```

- .active 클래스와 .saw 클래스 위치를 변경해 보면서 차이점을 살펴 봅시다.
- class 만으로는 완전히 해결할 수 없다.

## id 선택자의 필요성

- 태그 선택자 -> 클래스 선택자 -> id 선택자
- id 는 단 한번만 등장 해야 함.
  - 학교 -> 반 -> 번호(id)와 같은 개념.

```

<style>
a {
    color:black;
    text-decoration: none;
}
#active{
    color:red;
}
.saw{
    color:gray;
}

h1 {
    font-size: 50px;
    text-align: center;
}
</style>
...

<li><a href="1.html" class ="saw">리버풀</a></li>
<li><a href="2.html" class ="saw" id="active">버트런드 러셀</a></li>
<li><a href="3.html">고레에다 히로카즈</a></li>

```

- 즉, 태그(element), 클래스(class), 아이디(id)의 조합을 해서 우리가 원하는 디자인을 할 수 있습니다.
- [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

```

<!doctype html>
<html>
<head>
  <title>WEB - CSS</title>
  <meta charset="utf-8">
  <style>
    #active {
      color:red;
    }
    .saw {
      color:gray;
    }
    a {
      color:black;
      text-decoration: none;
    }
    h1 {
      font-size:45px;
      text-align: center;
    }
  </style>
</head>
<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>CSS</h2>
  <p>
    Cascading Style Sheets (CSS) is a style sheet language used for describing the
  </p>
</body>
</html>

```

- 선택자를 공부하기 좋은 사이트 입니다.
  - <http://flukeout.github.io>

---

## CSS box model

---

- 우리는 지난 시간에 CSS의 선택자와 속성에 대해 살펴보았습니다.
- HTML 의 경우는 문서를 위에서 부터 읽어나가면서 순차적으로 태그의 내용이 실행 되었다면, CSS 는 선택자를 이용하여 우선순위를 조절할 수 있었습니다.
- 이번에는 우리의 기획서를 구현하기 위해서 공간에 대한 이해를 해야 합니다. 즉, 이번에 배울 박스모델

은 공간에 관하여 입니다.

- 먼저 우리의 기획서(우리가 만드려는 웹사이트)를 다시 살펴봅시다.

## 내가 좋아하는 것들

1. 리버풀
2. 버트런드 러셀
3. 고레에다 히로카즈

### 버트런드 러셀

러셀은 40여 권에 이르는 수많은 저작을 남긴 철학자요 1950년에 노벨 문학상을 받은 문필가로 널리 알려져 있다. 20세기 지식인 가운데 가장 다양한 분야에서 영향력을 가졌던 러셀의 삶은 활동적인 사회 참여로도 유명하다.

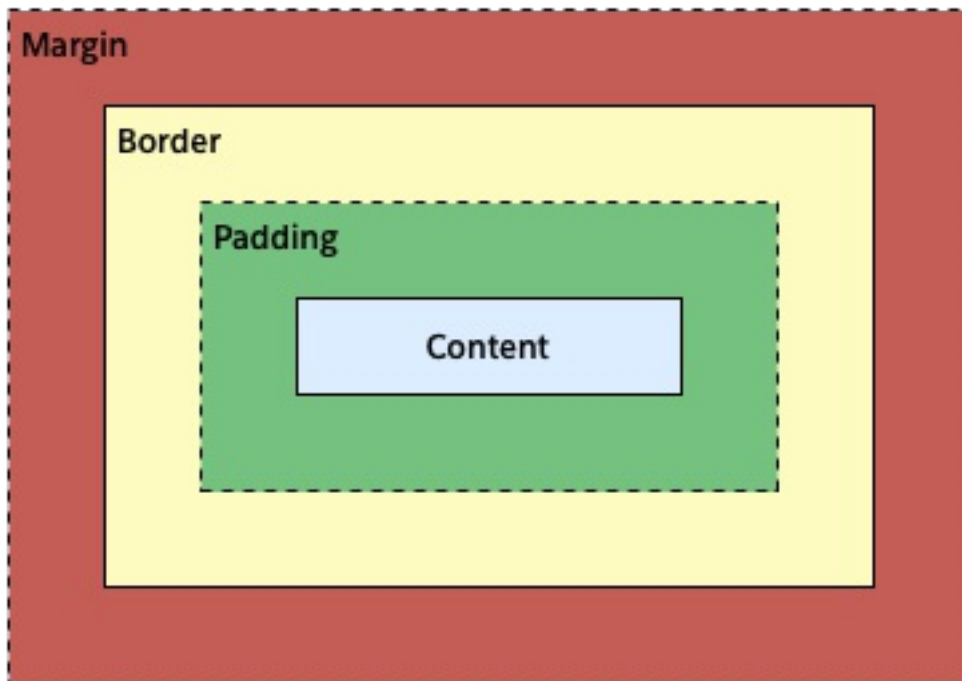
러셀의 저작 중에서 특히 주목받는 책이 바로 '게으름에 대한 찬양'이다.

- 1912. 철학이란 무엇인가
- 1927. 나는 왜 기독교인이 아닌가
- 1929. 결혼과 도덕에 관한 10가지 철학적 성찰
- 1930. 행복의 정복
- 1935. 게으름에 대한 찬양

- 실제로 웹 페이지는 모니터라는 2차원 평면에 내용을 다양한 방식으로 표현하고 있습니다.
- 즉, 우리가 사용하는 HTML 태그들은 기본적으로 모니터 상에서 일정한 공간을 차지하고 있다는 말입니다.
- 이것을 위해 우리는 박스모델을 이해해야 합니다.
- 먼저 박스모델을 이해하기 위해서 우리는 다음과 같은 HTML문서를 만들어 봅시다.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <style>
      h1{
        border:5px solid red;
        padding:20px;
        margin:20px;
        display:block;
        width:100px;
      }
    </style>
  </head>
  <body>
    <h1>CSS</h1>
    <h1>CSS</h1>
```

```
</body>
</html>
```



- 즉, 태그에 따라서 다음의 두가지 형태의 공간을 차지 합니다.
  - block level element: 태그가 화면 전체를 사용하는 경우
  - inline level element: 태그가 텍스트 만큼만 사용하는 경우
- [https://www.w3schools.com/css/css\\_boxmodel.asp](https://www.w3schools.com/css/css_boxmodel.asp)
- 우리의 기획서를 기반으로 `h1` 태그에게 밑줄을 그어 봅시다.

```
h1 {
  font-size:45px;
  text-align: center;
  border-bottom:1px solid gray;
  margin:0;
  padding:20px;
}
```

- 이제 세로 줄을 그어 봅시다.

```
ol{
```

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

- 그럼 우리의 홈페이지는 다음과 같은 형식이 되었습니다.

```
<!doctype html>
<html>
<head>
  <title>WEB - CSS</title>
  <meta charset="utf-8">
  <style>
    body{
      margin:0;
    }
    #active {
      color:red;
    }
    .saw {
      color:gray;
    }
    a {
      color:black;
      text-decoration: none;
    }
    h1 {
      font-size:45px;
      text-align: center;
      border-bottom:1px solid gray;
      margin:0;
      padding:20px;
    }
    ol{
      border-right:1px solid gray;
      width:100px;
      margin:0;
      padding:20px;
    }
  </style>
</head>
<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>CSS</h2>
  <p>
    Cascading Style Sheets (CSS) is a style sheet language used for describing the
```

```
</p>
</body>
</html>
```

- 이제 남은 것은 밑에 있는 내용을 오른쪽 위로 올리는 것 입니다.
- 그리드라는 개념을 통해서 이것을 해 봅시다.

## 그리드

- 그리드가 무엇인지 먼저 살펴 봅시다.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <style>
      #grid{
        border:5px solid pink;
        display:grid;
        grid-template-columns: 150px 1fr;
      }
      div{
        border:5px solid gray;
      }
    </style>
  </head>
  <body>
    <div id="grid">
      <div>NAVIGATION</div>
      <div>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
    </div>
  </body>
</html>
```

- 즉, 디자인을 위해서 아무런 정보를 담지 않고 있는 무색무취의 태그인 div 사용하였습니다.
- div 태그(block level), span 태그(inline level)
- [https://www.w3schools.com/cssref/pr\\_grid.asp](https://www.w3schools.com/cssref/pr_grid.asp)
- <https://caniuse.com/>

## 그리드 사용하여 우리의 웹 페이지 꾸미기

```
<!doctype html>
<html>
<head>
  <title>WEB - CSS</title>
  <meta charset="utf-8">
  <style>
    body{
      margin:0;
    }
    a {
      color:black;
      text-decoration: none;
    }
    h1 {
      font-size:45px;
      text-align: center;
      border-bottom:1px solid gray;
      margin:0;
      padding:20px;
    }
    ol{
      border-right:1px solid gray;
      width:100px;
      margin:0;
      padding:20px;
    }
    #grid{
      display: grid;
      grid-template-columns: 150px 1fr;
    }
    #grid ol{
      padding-left:33px;
    }
    #grid #article{
      padding-left:25px;
    }
  </style>
</head>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <div id="grid">
    <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>
    <div id="article">
      <h2>CSS</h2>
      <p>
        Cascading Style Sheets (CSS) is a style sheet language used for describin
      </p>
```

```
</div>
</div>
</body>
</html>
```

## 반응형 디자인

- 현재 우리는 개인용 PC뿐만 아니라, 스마트폰, 태블릿 등을 이용하여 웹을 사용 합니다.
- 즉, 이러한 환경은 서로 다른 화면의 크기에 따라서 웹의 정보들이 동적으로 반응하여 우리에게 최적의 형태로 전달 되어야 함을 의미합니다.
- 이것을 할 수 있는 것이 반응형 디자인이며, 미디어 쿼리를 통하여 실습해 봅시다.
- [https://www.w3schools.com/css/css\\_rwd\\_intro.asp](https://www.w3schools.com/css/css_rwd_intro.asp)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <style>
      div{
        border:10px solid green;
        font-size:60px;
      }
      @media(max-width:800px) {
        div{
          display:none;
        }
      }
    </style>
  </head>
  <body>
    <div>
      Responsive
    </div>
  </body>
</html>
```

- 우리의 웹 페이지에 적용해 봅시다.

```
<!doctype html>
<html>
```



```

<head>
  <title>WEB - CSS</title>
  <meta charset="utf-8">
  <style>
    body{
      margin:0;
    }
    a {
      color:black;
      text-decoration: none;
    }
    h1 {
      font-size:45px;
      text-align: center;
      border-bottom:1px solid gray;
      margin:0;
      padding:20px;
    }
    ol{
      border-right:1px solid gray;
      width:100px;
      margin:0;
      padding:20px;
    }
    #grid{
      display: grid;
      grid-template-columns: 150px 1fr;
    }
    #grid ol{
      padding-left:33px;
    }
    #grid #article{
      padding-left:25px;
    }
    @media(max-width:800px){
      #grid{
        display: block;
      }
      ol{
        border-right:none;
      }
      h1 {
        border-bottom:none;
      }
    }
  </style>
</head>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <div id="grid">
    <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>
    <div id="article">

```

```
<h2>CSS</h2>
<p>
  Cascading Style Sheets (CSS) is a style sheet language used for describin
</p>
</div>
</div>
</body>
</html>
```

## CSS 코드의 재사용

- 우리가 만들려고 하는 HTML은 총 4개로 이루어져 있습니다.
- 나머지 페이지들에게도 지금까지 한 내용을 적용해야 합니다.
- 그러나 직접 하는 것 보다는 효율적인 방식이 있습니다.
- 중복의 제거(우리가 해야 하는 일)는 코딩에서 중요한 요소 입니다.
- 2.html 은 다음과 같이 구성 합니다.

```
<!doctype html>
<html>
<head>
  <title>WEB - CSS</title>
  <meta charset="utf-8">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1><a href="index.html">WEB</a></h1>
  <div id="grid">
    <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>
    <div id="article">
      <h2>CSS</h2>
      <p>
        Cascading Style Sheets (CSS) is a style sheet language used for describin
      </p>
    </div>
  </div>
</body>
</html>
```

- 모든 HTML에서 사용하기 위해서 다음과 같은 style.css 파일을 만들어 둡니다.

```
body{
  margin:0;
}
a {
  color:black;
}
h1 {
  font-size:45px;
  text-align: center;
  border-bottom:1px solid gray;
  margin:0;
  padding:20px;
}
ol{
  border-right:1px solid gray;
  width:100px;
  margin:0;
  padding:20px;
}
#grid{
  display: grid;
  grid-template-columns: 150px 1fr;
}
#grid ol{
  padding-left:33px;
}
#grid #article{
  padding-left:25px;
}
@media(max-width:800px){
  #grid{
    display: block;
  }
  ol{
    border-right:none;
  }
  h1 {
    border-bottom:none;
  }
}
```

## CSS 정리

---

[https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)

- CSS는 (Cascading Style Sheets)의 약자 입니다.
- CSS는 HTML의 정보가 화면에 표시하는 방법을 설명합니다.

- 많은 양의 웹 페이지들을 한 번에 제어할 수 있기 때문에 시간을 절약할 수 있습니다.
- 외부 스타일 시트는 CSS파일에 저장 됩니다.

- 추가 (Box model을 이용한 디자인)

```
<!doctype html>
<html>
<head>
  <title>WEB - CSS</title>
  <meta charset="utf-8">
</head>
<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>
  <h2>CSS</h2>
  <p>
    Cascading Style Sheets (CSS) is a style sheet language used for describing the
    visually engaging webpages, user interfaces for web applications, and user inte
  </p>
</body>
</html>
```

- 다음의 Style 태그를 추가해 봅시다.

```
<style media="screen">
  a{
    color:black;
    text-decoration: none;
  }
  h1{
    text-align: center;
    border-bottom: 1px solid gray;

  }
</style>
```

- 이제 `ol` 태그 오른쪽에 선을 그어 봅시다.

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

```
width:100px;
}
```

- 그리드(grid)를 사용하여 공간을 분할해 봅시다.

```
<!doctype html>
<html>
<head>
  <title>WEB - CSS</title>
  <meta charset="utf-8">
  <style media="screen">
    a{
      color:black;
      text-decoration: none;
    }
    h1{
      text-align: center;
      border-bottom: 1px solid gray;
    }
    ol{
      border-right: 1px solid gray;
      width:100px;
    }
    #grid{
      display: grid;
      grid-template-columns: 150px 1fr;
    }
  </style>
</head>
<body>
  <h1><a href="index.html">WEB</a></h1>

  <div id="grid">
    <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>
    <div id="article">
      <h2>CSS</h2>
      <p>
        Cascading Style Sheets (CSS) is a style sheet language used for describing
        visually engaging webpages, user interfaces for web applications, and user
      </p>
    </div>
  </div>

</body>
</html>
```

- BOX 모델을 활용해서 어색한 부분을 정리해 봅시다.
  - 크롬 개발자 모드

```
<style media="screen">
  body{
    margin:0;
  }
  h1 {
    font-size:45px;
    text-align: center;
    border-bottom:1px solid gray;
    margin:0;
    padding:20px;
  }
  a{
    color:black;
    text-decoration: none;
  }

  ol{
    border-right:1px solid gray;
    width:100px;
    margin:0;
    padding:20px;
  }
  #grid ol{
    padding-left:33px;
  }
  #grid #article{
    padding-left:25px;
  }
  #grid{
    display: grid;
    grid-template-columns: 150px 1fr;
  }
</style>
```

- 다시 반응형 디자인 부분으로!