

```
<div>
  <label for="check">내가 컴퓨터 좀 만져봤다 거수</label>
  <input type="checkbox" id="check">
</div>
```

<div> 태그는 아무 의미 없이 한 줄을 차지할 때 쓰는 태그이며, 정말 많이 쓴다.  
아마 여러분이 html 코딩 시 쓰는 대부분의 태그는 <div> 태그일 것.  
여기선 input type 이 text 가 아니라 checkbox 일 경우

결과:

내가 컴퓨터 좀 만져봤다 거수 ☒

```
<div>
  <label for="radio1">통신병</label>
  <input type="radio" name="job" id="radio1">
</div>
<div>
  <label for="radio2">소총수</label>
  <input type="radio" name="job" id="radio2">
</div>
<div>
  <label for="radio3">행정병</label>
  <input type="radio" name="job" id="radio3">
</div>
```

라디오버튼 사용할 경우, 하나의 name으로 묶여있어야 한다.

결과:

통신병 ☐

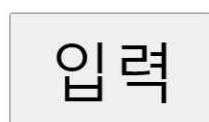
소총수 ☒

행정병 ☐

<button>입력</button>

버튼 태그이다. 여기에 onclick="functionName()" 을 달아서 작동시킴

결과:



<a href="https://www.mma.go.kr/index.do">병무청으로 가기!</a>

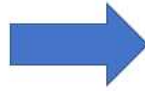
하이퍼링크를 의미한다.

href 는 이동할 페이지 주소이다.

같은 서버 안에 다른 html 경로도 입력 가능하다

결과:

병무청으로 가기!



---

## CSS

css 는 디자인을 위한 언어이다.

여러분의 웹을 예쁘게 만들기 위한 언어이고

역시 프로그래밍 언어가 아니다. 정확하게는 스타일시트 언어이다.

즉, HTML 과 CSS 만 배우고 어디가서 프로그래머라고 하면 안된다!

웹디자인과 가장 관련있는 학과는 산업디자인학과 정도가 있을텐데

거기서도 코딩을 하며, 거기서 하는 코딩이 바로 HTML 과 CSS 이다.

옛날옛적엔, HTML 안에 모든 디자인 요소를 사용했다.

그러다보니 하나의 HTML 파일이 엄청나게 복잡해졌다!

그래서 W3C에서 디자인을 담당하는 정식 언어를 채택한 것.

적용하는 법은 총 3가지가 있는데,

첫째는 태그에 속성으로 직접 넣는 것이다.

두번째는 <style> 태그를 사용하는 것이다.

세번째는 .css 파일을 <link> 태그로 첨부하는 것, 즉 파일을 따로 만드는 것이다.

여기서 가장 권장하는 방법은 빨간색으로 써놓은,

하나의 별개의 파일을 만들어서 작업하는 것이다.

왜 이렇게 해야할까?

CSS를 만든 목적 자체가, 디자인 요소를 HTML에서 분리하기 위함이다.

근데 위에 두 가지 방법은 어쨌든 HTML 안에 CSS 가 존재할 수밖에 없다.  
이것은 설계적으로, 매우 권장되지 않는 방법이다.  
심지어, 몇 줄 안되는 CSS를 쓰고싶다 하더라도 파일을 따로 만드는게 바람직하다.

## 1. 선택자 (selector)

선택자는 css 로 꾸밀 대상을 '선택' 할 때 쓴다

종류는 엄청나게 많다.

구글 검색: css selector w3schools

w3schools 는 전 세계에서 가장 유명한, 웹프로그래밍 튜토리얼 사이트이다.

예제도 매우 풍부하고, 여기 있는것들만 제대로 공부해도 웹 잘한다고 할 수 있다.

영어로 되어있는 게 단점이다.

잘 보면, 우리가 좀 있다가 기본적으로 배우게 될 엘리먼트(태그를 의미함), 아이디, 클래스를 포함해서

엄청나게 많은 것들이 있다. 이게 다 선택자다.

Selector	Example	Example description
<u>.class</u>	.intro	Selects all elements with class="intro"
<u>.class1.class2</u>	.name1.name2	Selects all elements with both <i>name1</i> and <i>name2</i> set within its class attribute
<u>.class1 .class2</u>	.name1 .name2	Selects all elements with <i>name2</i> that is a descendant of an element with <i>name1</i>
<u>#id</u>	#firstname	Selects the element with id="firstname"
<u>*</u>	*	Selects all elements
<u>element</u>	p	Selects all <p> elements
<u>element.class</u>	p.intro	Selects all <p> elements with class="intro"
<u>element,element</u>	div, p	Selects all <div> elements and all <p> elements
<u>element element</u>	div p	Selects all <p> elements inside <div> elements
<u>element&gt;element</u>	div > p	Selects all <p> elements where the parent is a <div> element
<u>element+element</u>	div + p	Selects the first <p> element that is placed immediately after <div> elements
<u>element1~element2</u>	p ~ ul	Selects every <ul> element that is preceded by a <p> element
<u>[attribute]</u>	[target]	Selects all elements with a target attribute
<u>[attribute=value]</u>	[target=_blank]	Selects all elements with target="_blank"
<u>[attribute~=value]</u>	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
<u>[attribute]=value]</u>	[lang=en]	Selects all elements with a lang attribute value equal to "en" or starting with "en-"
<u>[attribute^=value]</u>	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
<u>[attribute\$=value]</u>	a[href\$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
<u>[attribute*=value]</u>	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"
<u>:active</u>	a:active	Selects the active link
<u>::after</u>	p::after	Insert something after the content of each <p> element

여기서 좀 재밌는 것들도 찾아보면 있다. 예를 들면 hover 같은거.

눌러보면 예제가 나오는데,

[w3schools.com](http://w3schools.com) [wikipedia.org](http://wikipedia.org)

**Note:** The :hover selector style links on mouse-over.

hover는 마우스를 해당 부분에 올릴때만 활성화된다.

즉, 선택자를 많이 알면 알수록 여러분들이 할 수 있는게 많아진다.

하지만 전부 다 알려고 하지는 마라.

실습에선 딱 세 가지만 다뤄볼 것이다. element, id, class

selector

```
.test {  
  color: red;  
}
```

property

사용할때는 이런식으로 한다. 선택자 즉 selector를 맨 앞에 쓰고, 중괄호 치고,  
그 안에 css property들을 넣어서 적용한다.

이 예제의 경우엔 선택자로 테스트라는 이름을 가진 클래스를 사용했다.

그리고 컬러라는 property로 글자색깔을 빨간색으로 지정했다.

백문이 불여일견, 코드를 쳐보면서 익혀보도록 하자.

지금부터, selector와 property가 바로 나와서 당황할수도 있는데,

다 가르쳐줄테니깐 차근차근히 따라쳐보기만 하자

css 적용법에는 총 세가지가 있다고 했다.

첫번째는 태그에 바로 적용하는 것이다.

h1 태그인 즐거운 웹 시간! 을 빨간색으로 바꿔볼 것이다.

```
<h1 style="color: red">즐거운 웹 시간!</h1>
```

결과:

**즐거운 웹 시간!**

두번째는 <style> 태그를 이용하는 것이다.

<style> 태그는 <head> 안에다가 쓴다.

문법 기억해보자. 선택자 쓰자. 여기서 태그, 다른말로는 엘리먼트 h1을 선택하고, 중괄호 치고, property 쓰고, property의 값 쓰면 된다.

```
<style>
h1 {
    color: blue;
}
</style>
```

결과:

# 즐거운 웹 시간!

근데, 이렇게 쓰지 말라고 했다.

왜? 이렇게 쓰게 되면 어짜피 HTML 안에 존재하게 되고

CSS를 만든 원래의 목적,

HTML 에서 디자인을 분리하겠다는 순수한 목적과도 맞지 않다!

그럼 어떻게 해야 할까? 파일로 따로 분리하는것이다.

단 한줄을 적용하더라도, 파일로 따로 분리해야한다.

보통은, style 이라는 디렉터리를 만들어서,

html 파일 이름과 똑같이 이름짓고 확장자만 html 이 아니라 css라고 쓴다.

우리의 경우에는, index.css 일 것이다. 왜? index.html 이니깐.

항상, css 파일 만들고나면 <link> 태그로 임포트해줘야한다.

임포트는 한국어로 집어 넣는다는 말이다.

html 안에 css 파일을 집어 넣는다는 말이다.

<head> 안에, <meta> 아래에, 주석을 달아서 css임을 분명히 해주겠다.

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <!-- CSS -->
  <link rel="stylesheet" href="/style/index.css">
  <title>웹프로그래밍 실습</title>
</head>
```

참고로, 주석은 언어마다 전부 달라서 다 외우려고 하지 마라!

쉬운 방법으론, 주석 하고싶은거 드래그 하고,

컨트롤 누르고, 오른쪽 쉬프트 옆에 슬래시버튼 누르면

해당 언어에 맞는 주석이 만들어진다.

채팅창으로 단축키 적어두겠다. ctrl + /

그리고 그 아래에 <link> 태그를 달고, css 위치를 입력해주면 된다.

```
/*index.css*/
h1 {
  color: green;
}
```

결과:

# 즐거운 웹 시간!

즉, 어떤 방법이 가장 바람직하다고 했는가?

맨 마지막 방법,

단 한줄을 적용하더라도 css 파일을 따로 작성하는 게 가장 바람직하다.

다음은 선택자에 대해 알아보자. 아무 생각도 없이 css 적용해보았는데, 우린 선택자와 property에 대해 아무런 학습을 하지 않았다.

선택자 종류만 쪽 살펴봤을 뿐이다.

선택자는 딱 세가지만 제대로 써도 기본은 한다.

첫번째가 엘리먼트, 이걸 태그를 말한다.

두번째가 아이디, 이걸 해당 페이지에 딱 하나밖에 없는 유일무이한 존재를 말한다.

클래스는 여러개를 한꺼번에 처리할 때 유리하다.

이 중에서 제일 중요한 거 하나만 꼽으라면 클래스다.

직접 코딩 해보면 알게되겠지만, 여러분은 80%의 경우를 각각의 클래스 디자인하는데 쓰게 된다.

첫번째, 엘리먼트.

아까 h1 태그에 대해 알아보았는데, 만약 h1이 두개가 있다고 생각해보자.

<h1>즐거운 웹 시간!</h1>

<h1>뽀빠라뽀뽀뽀뽀</h1>

우린 이미 다음과 같은 css 파일을 만들어놓았다.

```
/*index.css*/  
h1 {  
    color: green;  
}
```

결과:

즐거운 웹 시간!

뽀빠라뽀뽀뽀뽀

즉, 결과를 보면 알 수 있듯이, <h1> 태그는 모두 초록색이 되었다.

태그 자체의 이름을 쓰면, 해당되는 태그는 전부 다 적용되는것이다.

다음 실습을 위해 css는 지워버리겠다.

두번째, 아이디.

아이디는 웹 페이지에서 오로지 딱 하나만 존재해야하며, 중복이 없어야한다.

쓸 때는 다음과 같이 쓴다.

<h1 id="pageTitle">즐거운 웹 시간!</h1>

<h1>뽀빠라뽀뽀뽀뽀</h1>

내가 하고 싶은 건, 페이지에 존재하는 <h1> 태그 전체가 아니라,

즐거운 웹 시간 부분, 즉, 아이디 이름 pageTitle 만 초록색으로 바꾸고 싶다.

css 파일에선, 다음과 같이 쓴다.



```
/*index.css*/
#pageTitle {
    color: green;
}
```

이와 같이, 아이디는 맨 앞에 샵을 붙여서 표현하고, 뒤에 아이디 이름을 쓴다

결과:

**즐거운 웹 시간!**

**뽀빠라뽀뽀뽀**

이 아이디는 웹 페이지에서 유일해야한다.

즉, 이런식으로 쓰지 말라는 것이다.

```
<h1 id="pageTitle">즐거운 웹 시간!</h1>
<h1 id="pageTitle">뽀빠라뽀뽀뽀</h1>
```

그러나 사실, 아이디는 실전 개발할 땐 거의 쓸 일이 없다.

미리 말했듯이, 여러분은 css 코딩의 80%를 클래스 쓰면서 보낼것이다.

세번째, 클래스.

클래스를 쓰면 비슷한 요소들을 한 번에 처리할 수 있다.

```
<h1 class="pageTitle">즐거운 웹 시간!</h1>
<h1 class="pageTitle">뽀빠라뽀뽀뽀</h1>
<h2 class="pageTitle">나는 조교</h2>
```

가만히 보면, <h1> 이라는것과 <h2> 는 태그부터가 다르다.

그러나, 클래스를 쓰면 이것들도 하나의 묶음으로 처리할 수 있다.

```
/*index.css*/
.pageTitle {
    color: green;
}
```

이와 같이, 클래스는 맨 앞에 점을 붙여서 표현하고, 뒤에 클래스 이름을 쓴다.

결과:

**즐거운 웹 시간!**

**뽀빠라뽀뽀뽀**

**나는 조교**

다음과 같이, <h1> 이든 <h2> 든, 하나의 클래스인 pageTitle 로 묶었기 때문에 한 번에 적용 가능한 것이다.

## 2. 기법

property를 하나하나 배우는 것보다 더 중요한 건, css 가 어떤 원리를 가지고 동작하는지 배우는 것이다.

자주 사용되는 기법을 정리해보았다.

- 하나의 태그에 여러개의 클래스 적용
- 동일한 CSS property를 가진 선택자들을 한번에 처리
- 여러 개 태그를 하나의 클래스로 지정
- 부모 자식 관계일 경우

ㄱ. 하나의 태그에 여러개의 클래스 적용

```
<body>  
  <div class="test1 test2">CSS 연습</div>  
</body>
```

```
/*index.css*/
.test1 {
  color: blue;
}

.test2 {
  font-weight: bold;
}
```

결과:

## CSS 연습

test1 클래스에서 컬러를 파란색으로 지정해줬고, test2 클래스에서 볼드체 설정을 해주었다. 이와 같이, 하나의 태그에 여러 개 클래스를 지정해 쓸 수 있다.

ㄴ. 동일한 CSS property를 가진 선택자들을 한번에 처리

```
<body>
  <div class="test1">CSS 연습</div>
  <div class="test2">즐거워!</div>
</body>
```

```
/*index.css*/
.test1,
.test2 {
  color: blue;
}
```

결과:

# CSS 연습 즐거워!

이 경우, 두 개의 태그가 둘 다 파란색이길 원한다.

그런데 property가 중복되는 경우이므로, 따로 쓰지 않고 쉼표로 구분해서 한번에 처리한다.

ㄷ. 여러 개 태그를 하나의 클래스로 지정

```
<body>  
  <div class="test">CSS 연습</div>  
  <div class="test">즐거워!</div>  
  <div class="test">랄랄라~</div>  
</body>
```

```
/*index.css*/  
.test {  
  color: blue;  
}
```

결과:

CSS 연습  
즐거워!  
랄랄라~

클래스는 원래 여러 태그에서 한번에 적용시킬 때 쓴다.

물론, 실제 개발할 땐 하나를 적용하더라도 아이디나 엘리먼트(태그)보다 클래스를 훨씬 많이 쓴다.

ㄴ. 부모 자식 관계일 경우

이것은 css의 철학과 큰 관련이 있다.

css에서 가장 중요하게 기억해야 될 점은, 자식은 부모의 css property를 물려받는다는 것이다!

```
<div class="parent">  
  <div class="child">부모 자식 테스트</div>  
</div>
```

```
/*index.css*/  
.parent {  
  color: blue;  
}
```

결과:

## 부모 자식 테스트

css property 를 적용한 클래스는 부모인 parent 이지, 자식인 child 가 아니다.

그런데도, 자식인 child 의 글자 색깔이 파란색으로 바뀐 것이다.

즉, 자식은 부모의 css property 를 물려받는다는것을 알 수 있다.

지금까지 한 것으로, css의 본질, 즉 기본적인 사용법은 다 배웠다.

property는 약 250개 정도가 된다. 다 알려고 하지 말자!

이제 각종 다양한 Property 를 자주 쓰는 것만 배워보도록 한다.

크게 다음 네 가지 카테고리에 속한다.

- 폰트
- 박스모델
- 포지션
- 레이아웃

## 폰트

글자와 관련된 부분이다.

font-size	폰트 사이즈를 바꾼다. 단위는 rem을 사용한다.
font-weight	폰트의 무게. 주로 bold 또는 bolder 사용
color	글자의 색깔이다. 세련된 색 쓰고싶으면, google검색 css color recommendation 검색 헥스 코드(#으로 시작하는 RGB 숫자 6자리) 사용가능
text-align	정렬할 때 쓴다.

```
<body>
  <div class="font-test">font 연습</div>
</body>
```

```
/*index.css*/
.font-test {
  font-size: 2rem;
  font-weight: bold;
  color: tomato;
  text-align: center;
}
```

- rem은 소수점 단위로 쓸 수 있다.
- text-align 은 다음을 사용한다.  
left

right  
center  
justify

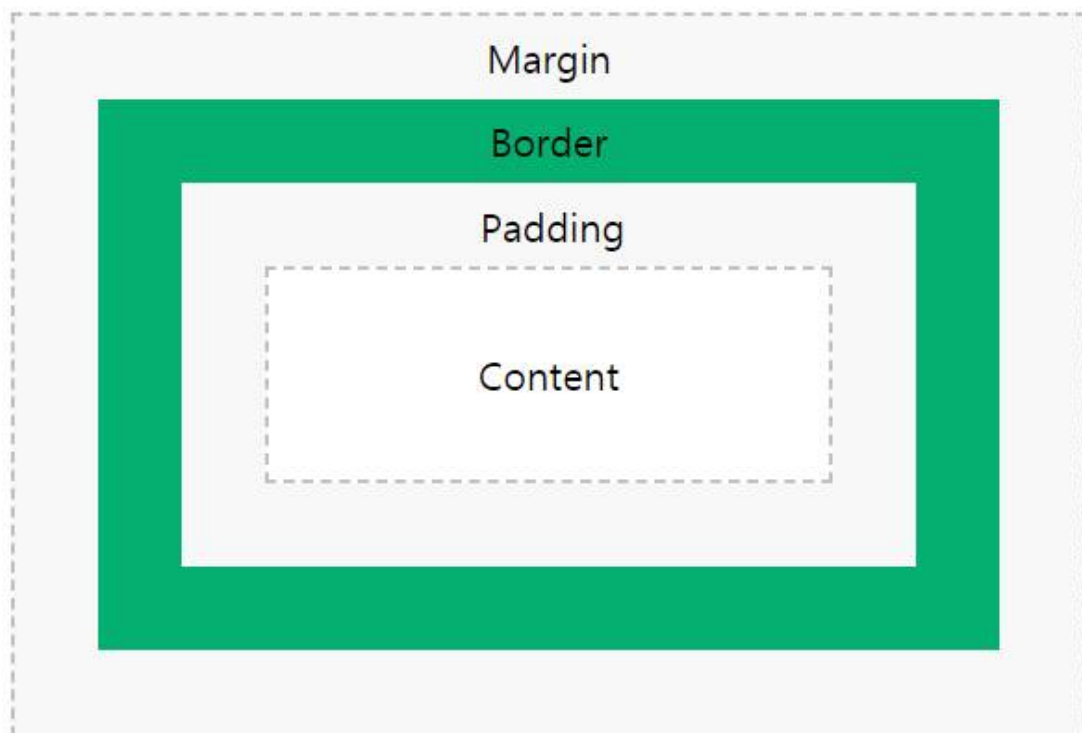
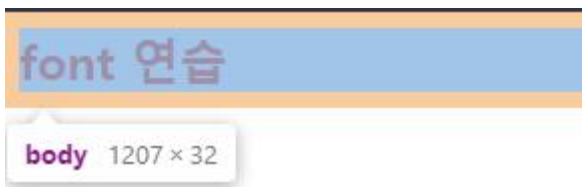
결과:

font 연습

## 박스모델

모든 태그는 박스(box) 로 이루어져있다.

구글 개발자도구 F12로 열어서, element 부분 찍어보면 다음과 같이 나온다.



- border      태그를 둘러싸는 경계선
- padding    태그의 콘텐츠와 보더간의 간격
- margin     태그와 태그간의 간격  
또는, 태그와 화면간의 간격

```
<body>
  <div class="margin-test">box1</div>
  <div class="margin-test">box2</div>
</body>
```

```
/*index.css*/
/*box model 연습*/
.margin-test {
  border: 1px solid black;
  /* margin-top: 10px;
  margin-right: 10px;
  margin-bottom: 10px;
  margin-left: 10px; */
  margin: 10px 10px 10px 10px;
}
```

border 는 경계선을 말한다. 뒤엔 순서대로,

1px    경계선 굵기

solid    경계선 형태. solid 외엔 거의 쓰지 않는다

black    선의 색깔

순서는 꼭 지키지 않아도 상관없다. 즉, black 1px solid 라고 써도 잘 동작한다.



border 는 실전에선 주로 테스트할때 사용한다.

내 태그가 위치에 맞게 잘 들어갔는지 보는 용도다.

margin의 순서는

top

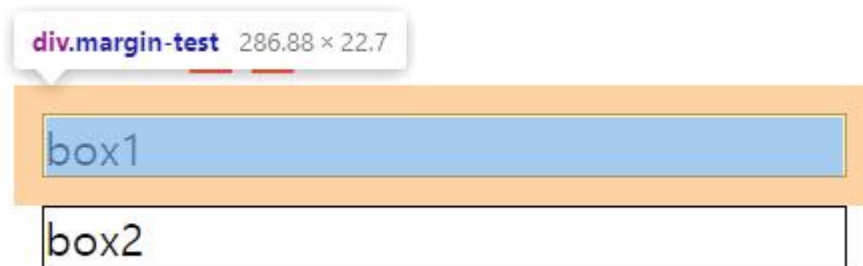
right

bottom

left

즉, 시계방향이며, 분리해서(margin-top 등) 쓸수도 있다.

결과:



이 경우, 마진 겹침 현상이 발생하며, 마진이 큰 쪽의 태그를 따른다.

```
<body>
  <div class="padding-test">padding-test</div>
</body>

/*index.css*/
.padding-test {
  border: 1px solid black;
  padding: 10px 10px 10px 10px;
}
```

결과:



즉, 패딩은 태그의 보더와 콘텐츠간의 간격임을 알 수 있다.

추가로, 태그를 채우는 색을 넣을수도 있다.

이 경우, background-color 를 사용한다.

```
.padding-test {  
  border: 1px solid black;  
  padding: 10px 10px 10px 10px;  
  background-color: tomato;  
}
```

결과:

