

CSS로 웹 디자인 해보자!

- * Web browser는 HTML만을 해석하기 만들어졌음.
- * 디자인 (폰트 색상, 글씨체 등)은 정보 (내용, 문단, 표 등)와 독립적인 개념. → 쉽게.
- * HTML은 이런 디자인을 포함해 이용하는데 몇 가지 불편요소가 존재.

Q. 그렇다면, 불편한 HTML의 개선을 어떻게 할까?

새로운 기능 추가 → 2가지 방법. 有

1. HTML에 새로운 태그 추가.

2. 그런 기능을 가진 새로운 언어 개발.

상연히 1번이 가능.

근데 가독성, 접근성 등에서 생긴 문제는 어떻게 해결 못 함!

↳ 코드만 그냥 더 길어지는 거잖아. 가독성 해결할었더니.

Q 2. 그럼, 새 기능 추가를 어떻게 하나?

- 추가 기능을 단계별 언어를 만들고. (이게 CSS) ~~※~~ style 속성
+
- 그 언어를 변경할 수 있는 태그 생성 (이게 style 태그).
또는 속성 ~~※~~

- * CSS는 HTML과는 완전히 다른 언어.
- * 도입된 이유 1. HTML 웹의 정보를 담고 있는 중요한 애.
 - 정보에 전념시키고 디자인 기능을 뺏어옴.
- 2. 디자인을 효율적으로 할 수 있음.
 - key word
 - 중복표현 제거, 유지관리도 용이!
- * style 태그 안에 {}로 가득이 놓고 속성을 부여함.
 - 이제 바로 효율적인 부분.
 - 코드가 복잡해져서 이상한거 간느낄 수도 있고
중복된 여러 개를 다 바꿔야 될 수도 있음. CSS는 그 문제를 없애버림.

* html 주석처리 : <!-- -->. 신기 ㅋㅋ.

- * CSS 문법
 - 기본적으로 <head> 태그 안에 <style> 태그 이용
- (ex)
- ```

<html>
 <head>
 <title> ~
 <meta> ~
 <style>
 a { }
 span { }
 h1 { }
 </style>
 </head>
 <body>
 <h1>Hello World!</h1>
 <p>This is my first web page.</p>
 Link
 </body>
</html>

```
- 세미콜론 ;**
- : 속성 하나에 여러 효과를  
구사할 경우 구분자.
- 선택자 이용 = selector**
- 효과를 주 대상 =**  
**선택하는 봄 = 선택자.**
- 효. = declaration.**
- property**      **value**

text-decoration: none !

```
</style>
</head>
```

→ 하지만, 유일한 방법은 이걸 => style 속성으로 해석하는 테크닉은  
효과 부여 가능!

→ 즉, style 이는 CSS의 차체이며 이 안에 적힌 코드는  
html() CSS의 문법에 따라 해석하여 결과를 적용.

\* 이제 매우 다양한 property를 용도에 맞게 감색해서 사용하면 됨.  
예시로의 차동성, 퀘션 가능을 적극 활용하자.

\* 사용 예제. 모든 링크를 3개로 구분: 강조, 흐릿, 빠감-

```
<head>
<style>
a {
 color: black;
 text-decoration: none;
}
class ← .saw {
 color: grey;
}
id ← #active {
 color: red;
}
</style>
</head>
<body>
<a>
```

1. HTML
2. CSS
3. Java



```
HTML
CSS
Java

</body>
```

---

## \* 박스 모델.

어떤 태그가 차지하는 공간이 따라서 나뉘는 개념.

웹레이아웃 전체 (가로) 를 보면 "block" element

필요한 만큼 깊이도 함께 쓰면 "inline" element 를 구분.

### (2) <style>

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

마찬가지.  
(block)



title\_h1

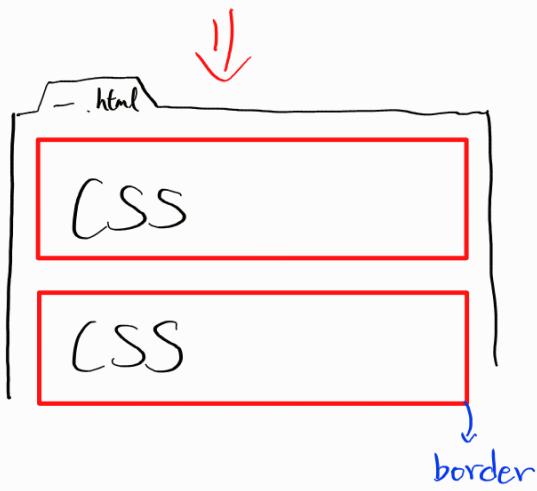
link part is Here, and show.

→ 가능성이 많아져서 표현 가능.

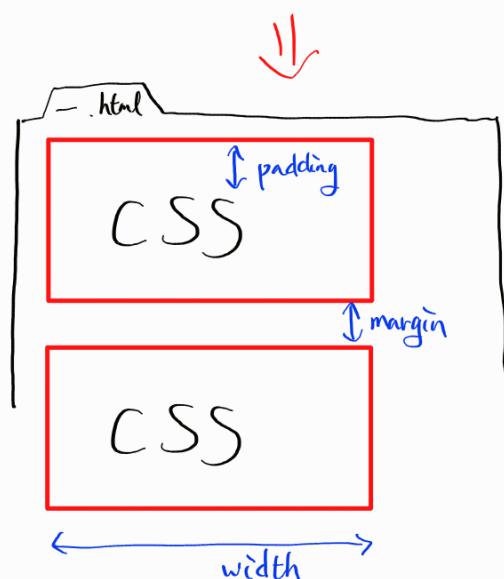
```
<style>
 h1, a { border: 5px red solid; }
</style>
```

## \* 박스 모델 다는거.

```
(Q) <head>
 <style>
 h1 { border: 5px solid red;
 }
 </style>
 </head>
 <body>
 <h1> CSS </h1>
 <h1> CSS </h1>
 </body>
```



```
<head>
 <style>
 h1 { border: 5px solid red;
 padding: 20px;
 margin: 20px;
 display: block;
 width: 100px; }
 </style>
 </head>
 <body>
 <h1> CSS </h1>
 <h1> CSS </h1>
 </body>
```



\* 박스 모델 뿐 아니라 모든 html css 구성을 내용은  
F12 (개발자 도구)에서 style 태그쪽에 차세한 정보 내용.

## \* 그리드

- 박스로 냄프어 내용 나누기.
  - div → block level element
  - span → inline level element
- 각 태그를 중복되게 감싸 구성하고, id를 부여해 조작. 하는 법!
- display 속성에서 grid 선택 \*

(2) <head>  
<style>

```
#my-grid { border: 5px solid pink;
 display: grid; *}
 grid-template-columns: 150px 1fr;
 }
```

전체를 나누는 단위.  
→ 150px 세의 나머지를  
1개로 나눔.

div { border: 5px solid gray; }

</style>

</head>

<body>

<div id="my-grid">

<div> SJ </div>

<div> CSS </div>

</div>

</body>

만약 : 2fr 1fr; 이면

2 1  
Doc start Doc end  
2 4등)

\* 태그를 id로 제한하기

- h2, ol, li 같은 태그 기능은,

문서 전체가 아닌 하나의 grid에서만 바꾸려면?

→ #id-sj h2 {  
}

와 같이 id로 묶어서 사용하면 됨.

ex) <style>

a { color: red; }      ⇒ 모든 링크색 빨간색  
</style>

<style>

#article { display: grid;  
grid-template-columns: 1fr 1fr; }

#article a { color: red; }

</style>



article id 가 부여된  
grid 속 링크만 빨간색

## \* 반응형 웹 / 디자인

- 화면 크기 (또는 구동 환경)에 따라서 각 요소들이 반응해서 작동하는 것.
- CSS에서 반응형 웹 구현은 미디어 쿼리 (mediaquery) 사용.  
↳ ①로 제이.

(2) <style>

```
div { border: 10px solid blue;
 font-size: 60px;
}
```

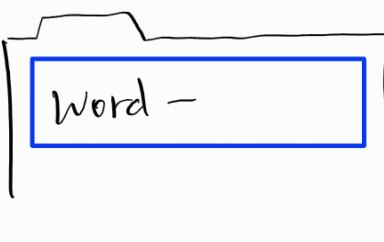
미디어 쿼리 ↗  
① media(min-width: 800px) {  
 div { display: none; }  
}

</style>

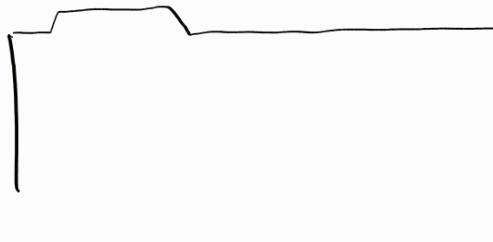
if  
screen width > 800px  
같은 데



↔ 790px



↔ 850px

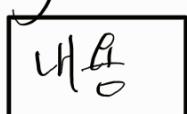


display 활성!

None

## \* CSS (style)의 확장.

<style>



</style>

이러한 방식 뿐만 아니라  
my-style.css 와 같이 만들면  
여러 문서에서 편리하게 사용 가능.

(2)

호출 방법 : link 태그 이용.

<head>

<title> ~

<meta> ~

<link rel="stylesheet" href="my-style.css">

</head>



같은 경로의 my-style.css를 찾아서 불러옴.

→ 이렇게 하게 되면 코드 간편해지고 캐시 메모리의  
사용으로 빠르게 사용할 수 있음. cash memory

