

**CSS**

**HTML**



*Markup Language*  
**Content**

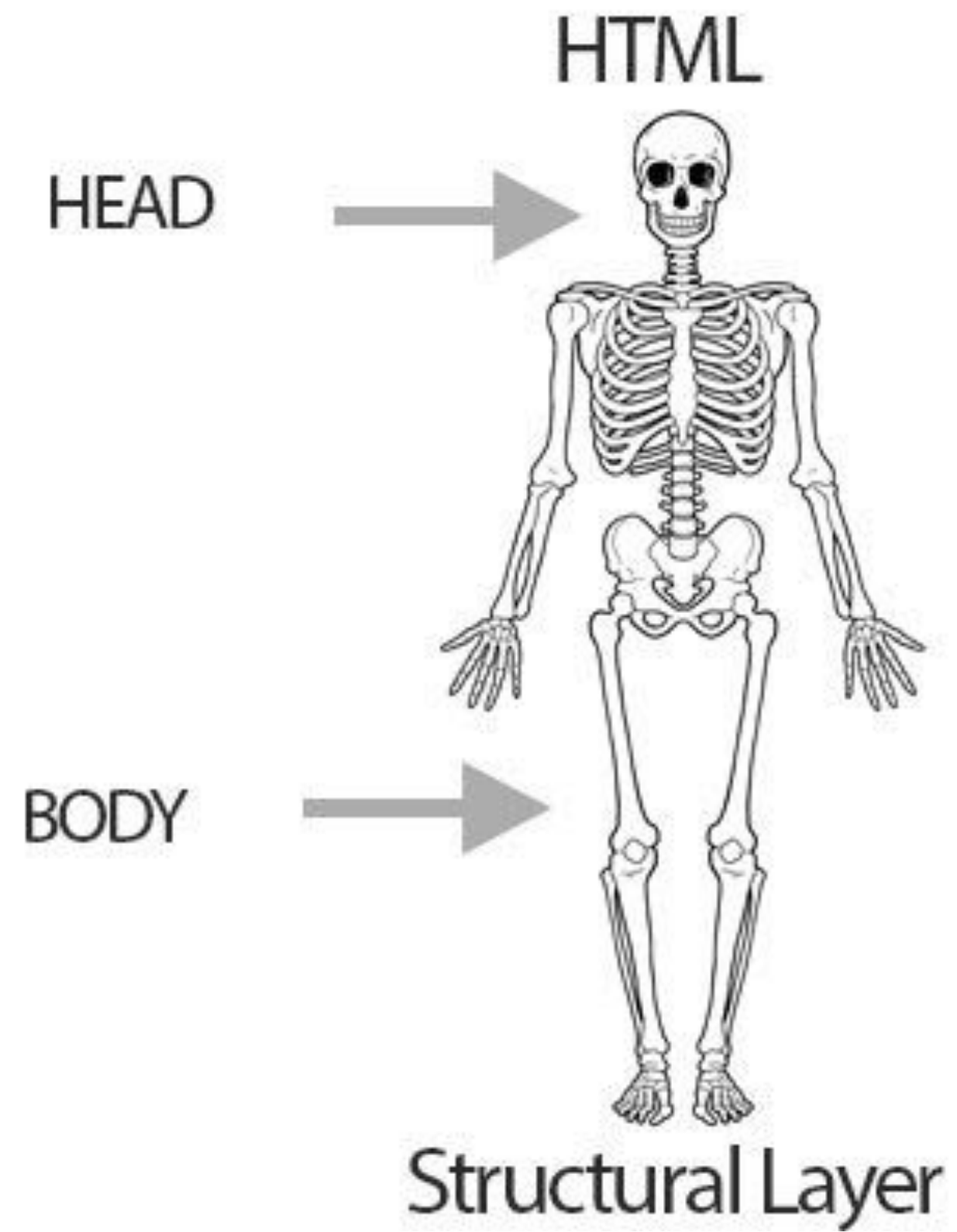
**CSS**



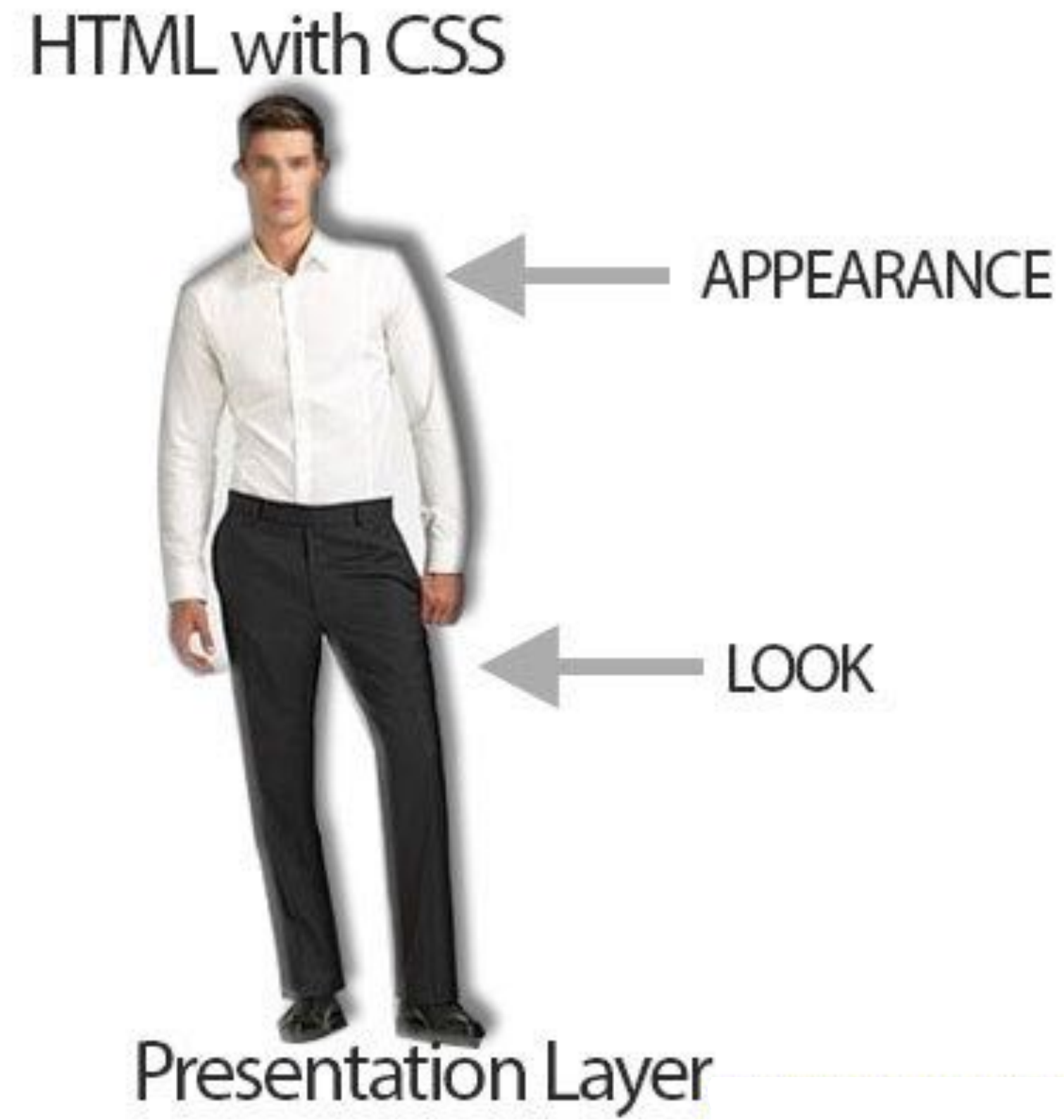
*Style sheet Language*  
**Presentation**

**HTML은 정보와 구조화**

**CSS는 styling의 정의**



**HTML은 정보와 구조화**



**CSS는 styling의 정의**

# 각자 문법이 다른 별개의 언어

하지만 HTML없는 CSS는 무의미...





# 기본 사용법

셀렉터  
(Selector)

선언

선언

```
h1 { color: blue; font-size: 15px; }
```

프로퍼티

(Property)

값

(Value)

# CSS 활용하기 1. Inline(인라인)

```
<> index.html ●
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>mySite</title>
5  </head>
6  <body>
7  |   <h1 style="color: blue; font-size: 100px">This is my site</h1>
8  </body>
9  </html>
```

HTML 요소의 style에 CSS를 넣기

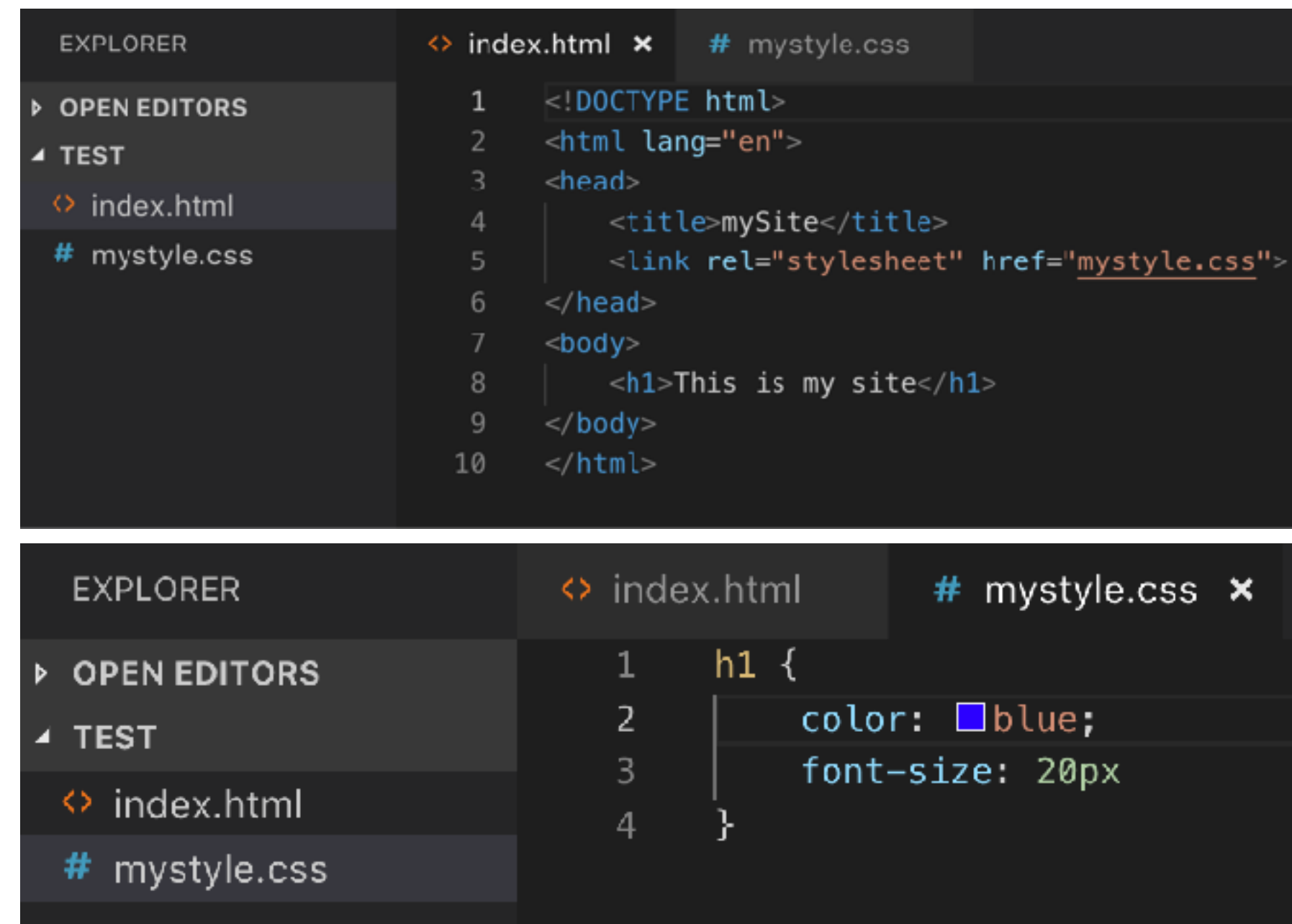
# CSS 활용하기 2. Embedding(내부참조)

```
<> index.html x
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <title>mySite</title>
5      <style>
6          h1 {
7              color: blue;
8              font-size: 100px
9          }
10     </style>
11 </head>
12 <body>
13     <h1>This is my site</h1>
14 </body>
15 </html>
```

HTML 내부에 CSS를 포함시키기



# CSS 활용하기 3.link file(외부참조)



The image consists of two screenshots of a code editor interface, likely VS Code, demonstrating how to link an external CSS file.

**Top Screenshot:** Shows the `index.html` file being edited. The Explorer sidebar on the left shows the project structure with `index.html` and `mystyle.css`. The code in `index.html` is as follows:

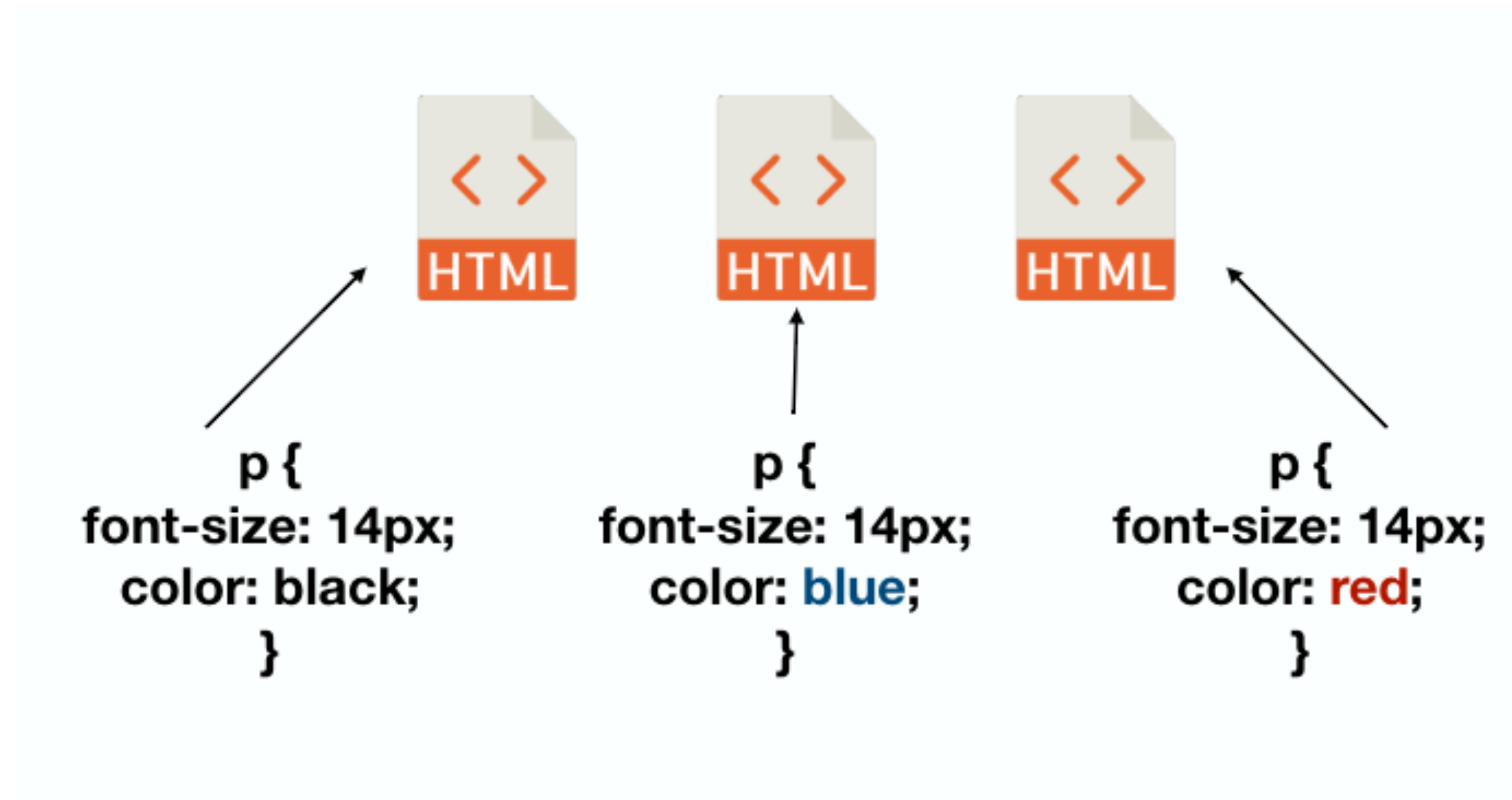
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>mySite</title>
5   <link rel="stylesheet" href="mystyle.css">
6 </head>
7 <body>
8   <h1>This is my site</h1>
9 </body>
10 </html>
```

**Bottom Screenshot:** Shows the `mystyle.css` file being edited. The Explorer sidebar on the left shows the project structure with `index.html` and `mystyle.css`. The code in `mystyle.css` is as follows:

```
1 h1 {
2   color: blue;
3   font-size: 20px;
4 }
```

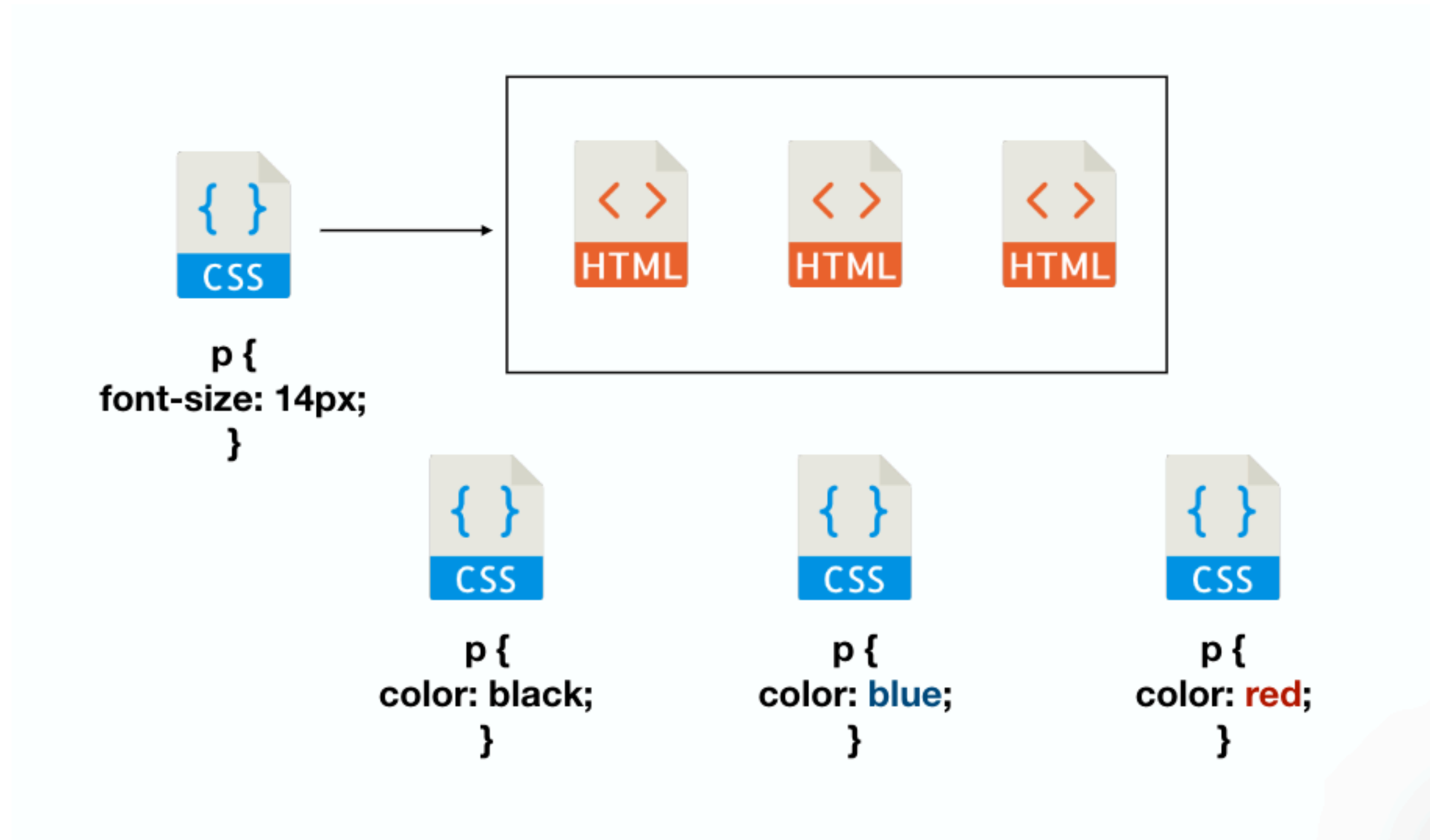
외부에 있는 CSS 파일을 로드하기

## 실제로 프로젝트에서 활용한다면, 어떤 것을 사용해야할까?



# ‘컴포넌트화’

## 일반적으로 외부 파일로서 활용한다!



# 들어가기 전에

<https://developer.microsoft.com/en-us/microsoft-edge/platform/usage/>

# CSS 단위

# 프로퍼티 값의 단위

```
h1 { color: blue; font-size: 15px; }
```

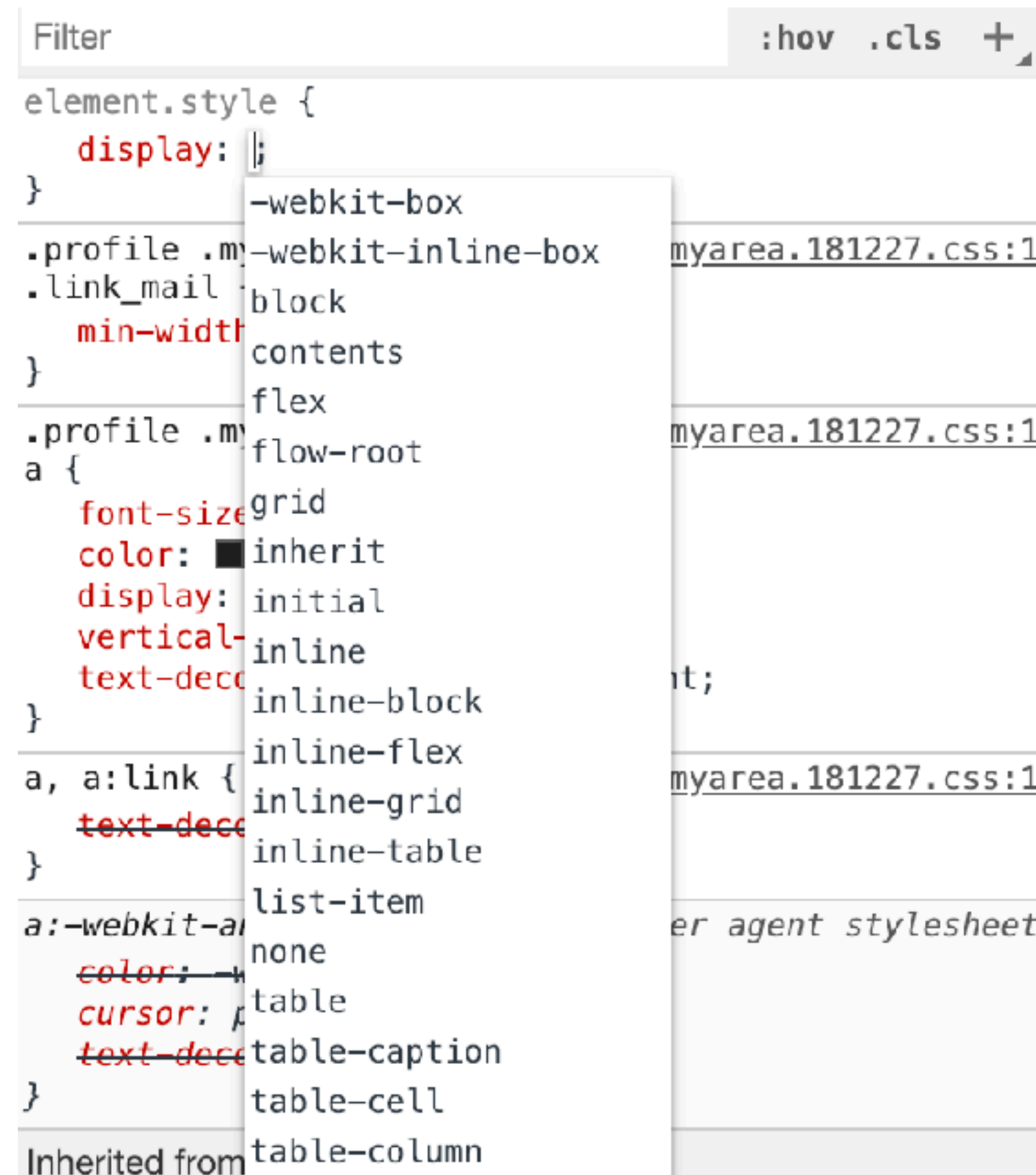


값  
(Value)

1. 키워드
2. 크기단위
3. 색깔




# 1. 키워드



개발자 도구로 확인

## 2. 크기단위

▼ 길이 | 넓이 | 무게 | 부피 | 온도 | 압력 | 속도 | 연비 | 데이터양 | 시간

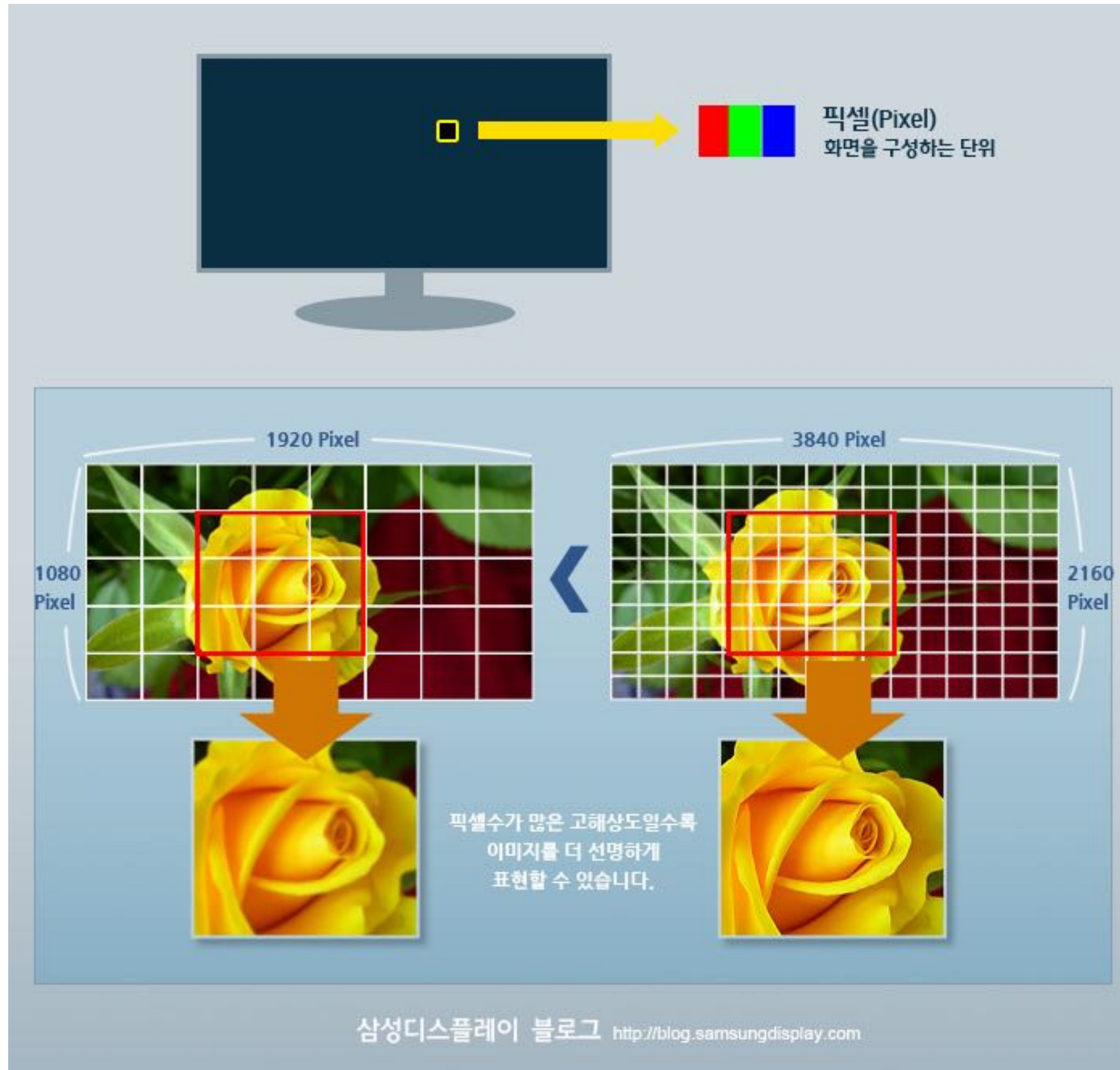
센티미터 (cm) → 인치 (in) 

1 cm = 0.393701 in

10 밀리미터(mm)	1 센티미터(cm)	0.01 미터(m)
0.00001 킬로미터(km)	0.393701 인치(in)	0.032808 피트(ft)
0.010936 야드(yd)	6.2137e-6 마일(mile)	0.033 자(尺)
0.0055 간(間)	0.000092 정(町)	0.000025 리(里)
5.3996e-6 해리(海里)		

우리가 알고 있는 크기단위

## 2.1. px



## 2.1. px

**디바이스별로 픽셀의 크기는 제각각!**

~~대부분의 브라우저는 1px을 1/96 인치의 절대단위로 인식~~

아... 그렇구나

## **2.2. %**

**%는 백분율 단위의 상대 단위이다.  
요소에 지정된 사이즈(상속된 사이즈나 디폴트 사이즈)에  
상대적인 사이즈를 설정한다.**

## 2.2. %

```
<> index.html x # mystyle.css
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>mySite</title>
5  |   <link rel="stylesheet" href="mystyle.css">
6  </head>
7  <body>
8  |   <h1>This is my site</h1>
9  |   <div>
10 |       <h1>h1 tag in div</h1>
11 |   </div>
12 </body>
13 </html>
```

```
<> index.html # mystyle.css x
1  h1 {
2  |   color: blue;
3  |   font-size: 20px;
4  |   width: 50%;
5  | }
6  div{
7  |   width:50%;
8  | }
```



## 2.3. em

**em 은 배수 단위로 상대 단위이다.  
요소에 지정된 사이즈(상속된 사이즈나 디폴트 사이즈)에  
상대적인 사이즈를 설정한다.**

## 2.4. rem

em 의 기준은 상속의 영향으로 바뀔 수 있다.

즉, 상황에 따라 1.2em은 각기 다른 값을 가질 수 있다.

rem은 최상위 요소(html)의 사이즈를 기준으로 삼는다.

rem의 r은 root를 의미한다.

## 2.5. Viewport 단위

디바이스마다 다른 크기의 화면을 가지고 있기 때문에  
상대적인 단위인 viewport를 기준으로 만든 단위

단위	비고
vw	너비의 1/100
vh	높이의 1/100
vmin	너비 또는 높이 중 작은 쪽의 1/100
vmax	너비 또는 높이 중 큰 쪽의 1/100

IE 8 이하는 지원하지 않으며  
IE 9 ~ 11, Edge는 지원이 완전하지 않으므로 주의가 필요하다.

### 3. 색상 표현 단위

<https://www.w3.org/TR/css-color-3/>

이름으로 색상을 쓰려면 요기로!

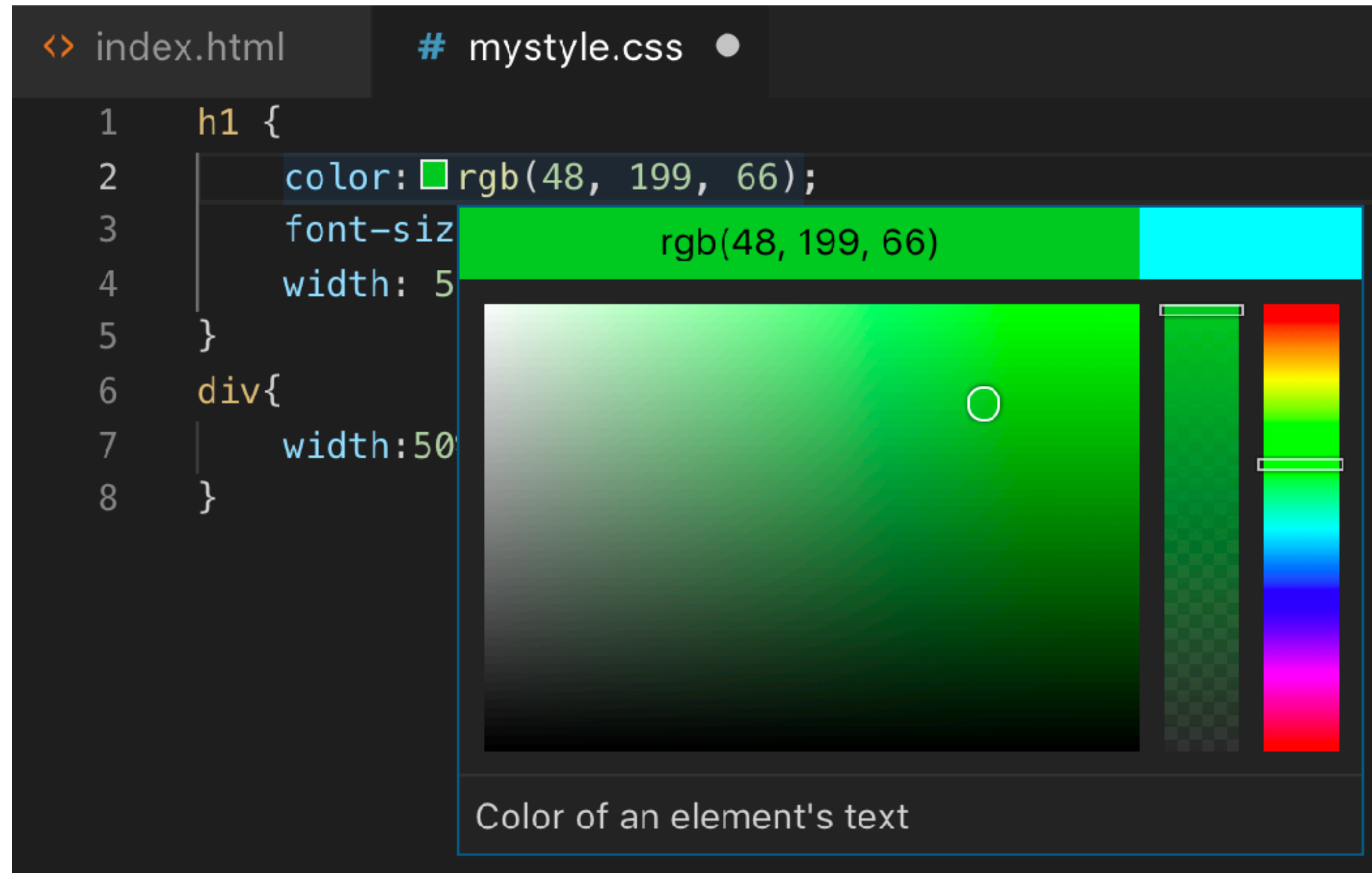
<https://htmlcolorcodes.com/>

더 상세한 색상을 쓰려면 요기로!

### 3. 색상 표현 단위

단위	비고
HEX	#ffffff
RGB	rgb(0, 0, 0)
RGBA	rgb(0, 0, 0, 0.5)

# 3. 색상 표현 단위





# Box model

# This is my site

## h1 tag in div

```
div#circle | 50x50
```



사각형을 돌려깎기 해서 만든 원

당신은 네모네모 멍뭉미와 눈미 마주치고 말  
랐습니다. 당신은 이제 네모네모 멍뭉미의 저  
주로 돛그란 글자를 칠수 없습니다. 멍멍!

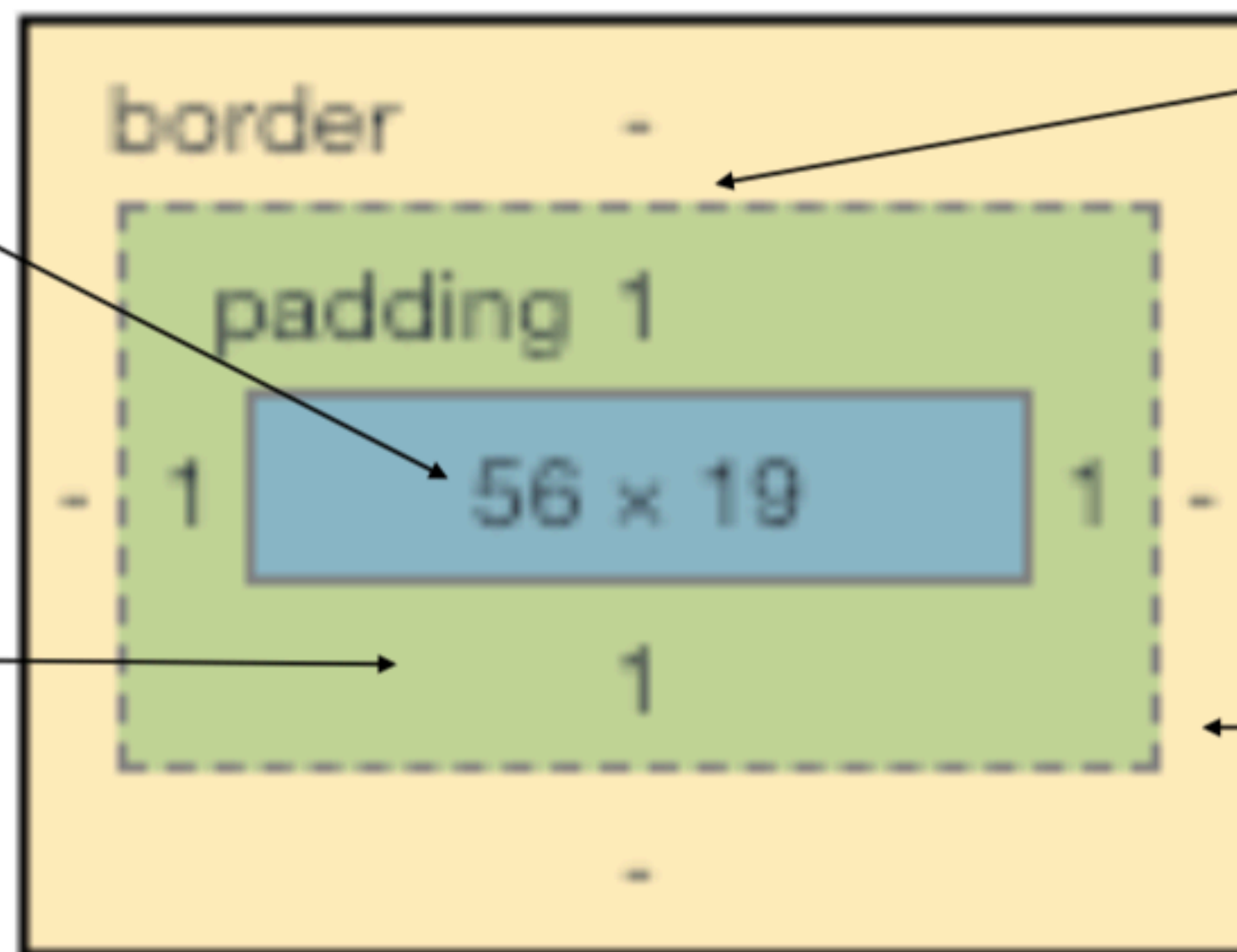


**모든것을 네모로 바라보자**  
네모의꿈

**Content**  
실제 내용이 위치

**Border**  
테두리 영역

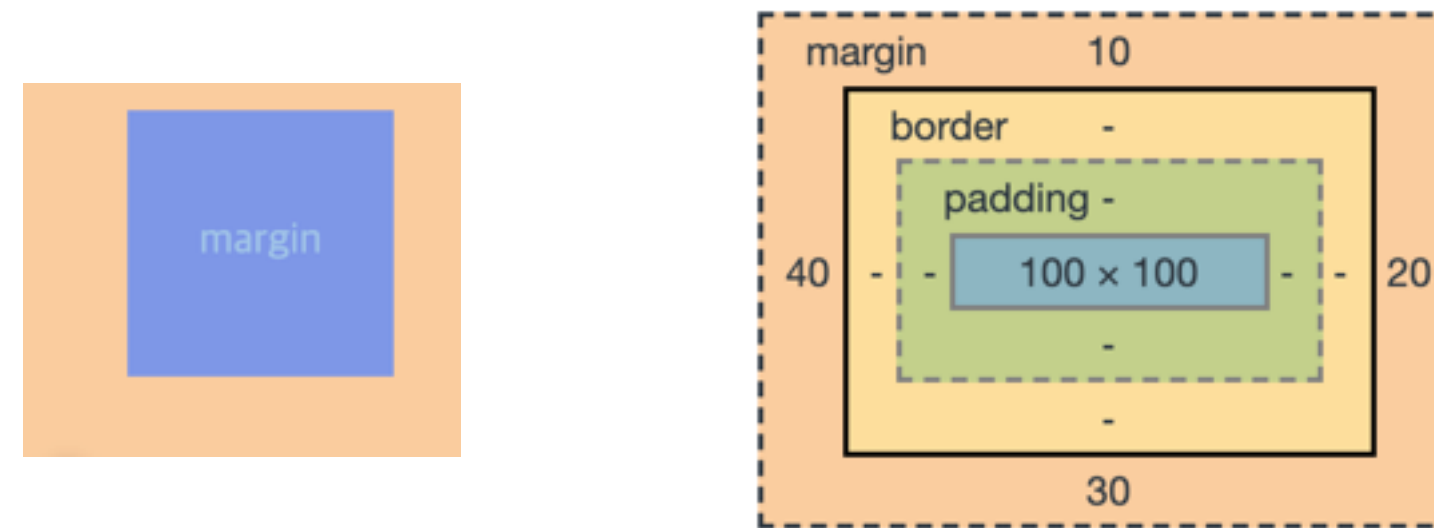
**Padding**  
테두리 안쪽의 내부 여백  
요소에 적용된 배경의  
컬러, 이미지는 패딩까지 적용



**Margin**  
테두리 바깥의 외부 여백  
배경색을 지정할 수 없다

## box model의 구성

# 1.1. 기본 박스모델 활용 - margin

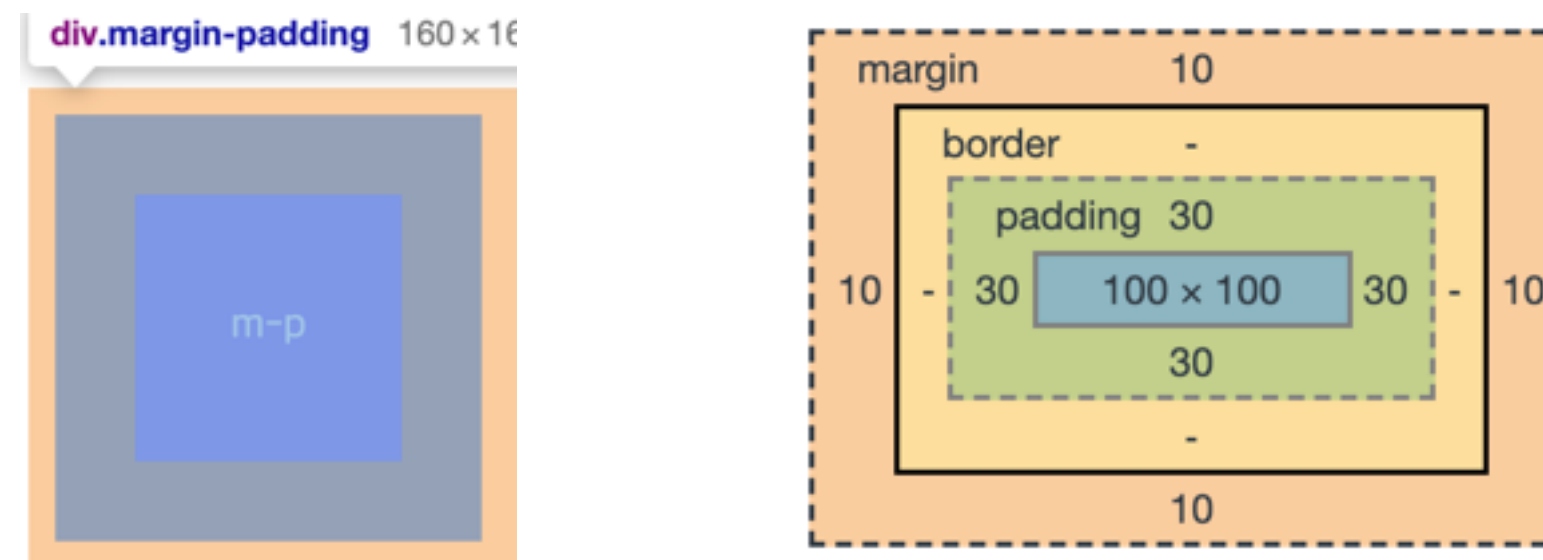


```
.margin {  
  margin-top: 10px;  
  margin-right: 20px;  
  margin-bottom: 30px;  
  margin-left: 40px;  
}
```

**상하좌우!**

Padding도 동일하다!

## 1.2. 기본 박스모델 활용 - padding



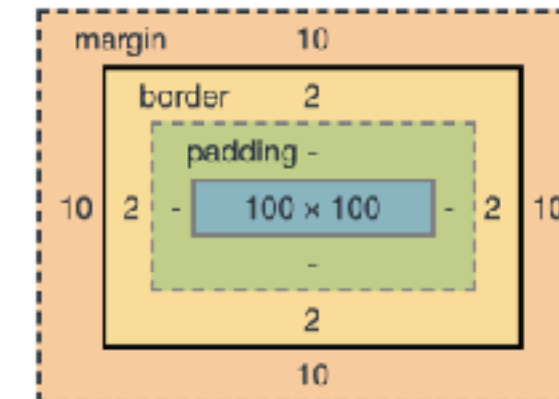
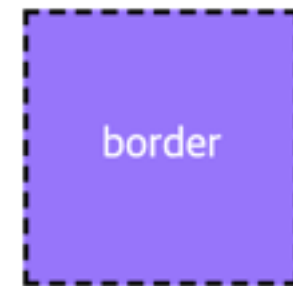
```
.margin-padding {  
  margin: 10px;  
  padding: 30px;  
}
```

**상하좌우!**

Padding도 동일하다!



# 1.3. 기본 박스모델 활용 - border



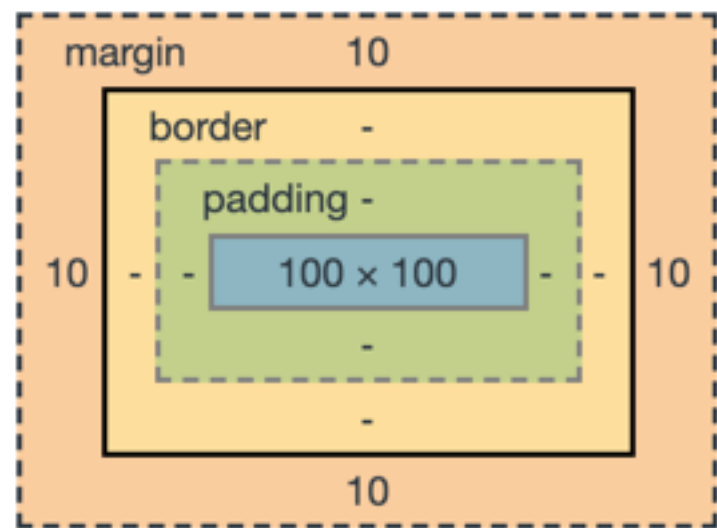
```
.border {  
  border-width: 2px;  
  border-style: dashed;  
  border-color: black;  
}
```

Filter	Show all
background-color	rgb(151, 117, 250)
border-bottom-color	rgb(0, 0, 0)
border-bottom-style	dashed
border-bottom-width	2px
border-left-color	rgb(0, 0, 0)
border-left-style	dashed
border-left-width	2px
border-right-color	rgb(0, 0, 0)
border-right-style	dashed
border-right-width	2px
border-top-color	rgb(0, 0, 0)
border-top-style	dashed
border-top-width	2px

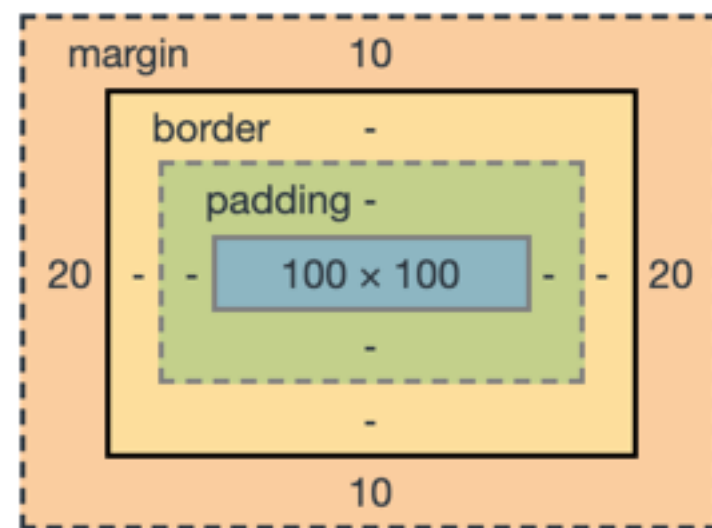
상하좌우!

border도 상하좌우 따로 설정도 가능하다!

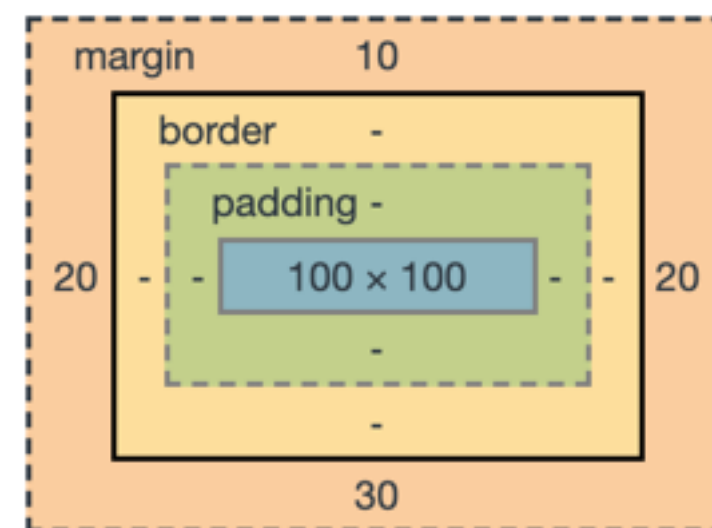
# 1.4. 기본 박스모델 활용 - shorthand



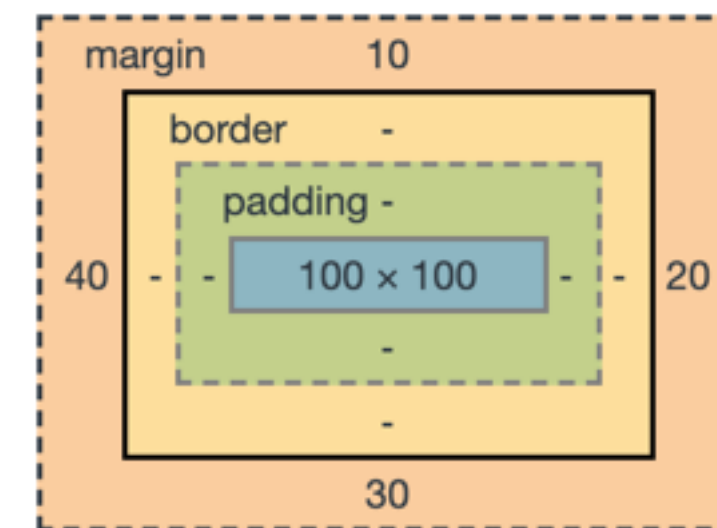
```
.margin-1 {  
  margin: 10px;  
}
```



```
.margin-2 {  
  margin: 10px 20px;  
}
```



```
.margin-3 {  
  margin: 10px 20px 30px;  
}
```

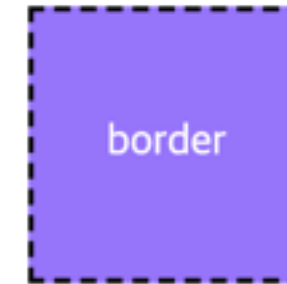


```
.margin-4 {  
  margin: 10px 20px 30px 40px;  
}
```

다양한 shortcut 활용도 가능하다.

Padding도 동일하다!

# 1.4. 기본 박스모델 활용 - shorthand



<pre>.border {   border-width: 2px;   border-style: dashed;   border-color: black; }</pre>	<pre>.border {   border: 2px dashed black; }</pre>
--	--

**border 역시도 단축어가 있다.**

## **2. display 속성**

**1. block**

**2. inline**

**3. inline-block**

**4. none**

## **2.1. block**

**항상 새로운 라인에서 시작한다.**

**화면 크기 전체의 가로폭을 차지한다. (width: 100%)**

**block 레벨 요소 내에 inline 레벨 요소를 포함할 수 있다**

# 2.1. block

기본적으로 너비의 100%!

## Block 요소

첫 번째 블록 요소에서 시작한다.

**h1** 824 x 39

StylesEvent ListenersDOM BreakpointsPropertiesAccessibility

Filter: :hov .cls +

element.style {  
}  
**h1** { user agent stylesheet  
display: block;  
font-size: 2em;  
margin-block-start: 0.67em;  
margin-block-end: 0.67em;  
margin-inline-start: 0px;  
margin-inline-end: 0px;  
font-weight: bold;  
}  
Inherited from **html**  
**html** { user agent stylesheet  
color: -internal-root-color;  
}

margin21.440  
border-  
padding-  
824 x 39  
21.440

Filter: Show all  
▶ color: rgb(0, 0, 0)  
▶ display: block  
▶ font-size: 32px  
▶ font-weight: 700

# 2.1. block

너비가 정해지면 나머지를 margin으로!

div 100×100에서 시작한다.



StylesEvent ListenersDOM BreakpointsPropertiesAccessibility

Filter: :hov .cls +

element.style {  
}  
  
div {03\_css.html:9  
height: 100px;  
width: 100px;  
background-color: #9775fa;  
}  
  
div {user agent stylesheet  
display: block;  
}  
  
Inherited from html  
html {user agent stylesheet  
color: -internal-root-color;  
}

FilterShow all

▶ background-colorrgb(151, 117, 250)

▶ colorrgb(0, 0, 0)

▶ displayblock

▶ height100px

▶ width100px

## 2.1. block

**margin-right: auto;**



**margin-left: auto;**



**margin-right: auto;  
margin-left: auto;**



## block의 가로정렬 활용법



## **2.1. block**

**block 레벨 요소 예**

**div, h1 ~ h6, p, ol, ul, li, hr, table, form**

## 2.2. inline

새로운 라인에서 시작하지 않으며 문장의 중간에 들어갈 수 있다.

content의 너비만큼 가로폭을 차지한다.

width, height, margin-top, margin-bottom 프로퍼티를 지정할 수 없다.

상, 하 여백은 line-height로 지정한다.

## 2.2. inline

인라인 요소

`<span>인라인 요소</span>`

`<input type="text" placeholder="텍스트를 입력하세요.">`

`<input type="date">`

## **2.2. inline**

**inline 레벨 요소 예**

**span, a, strong, img, br, input, select, textarea, button**

## 2.3. inline-block

**block과 inline 레벨 요소의 특징을 모두 갖는다.**

**inline 레벨 요소처럼 한 줄에 표시 되면서**

**block에서의 width, height, margin(top, bottom) 속성을 모두 지정할 수 있다.**

## 2.4. None

해당 요소를 화면에 표시하지 않는다. (**공간조차 사라진다**)

# **3. visibility 속성**

**1. visible**

**2. hidden**

## **3.1. visible**

**해당 요소를 보이게 한다.(기본값)**



## **3.2. hidden**

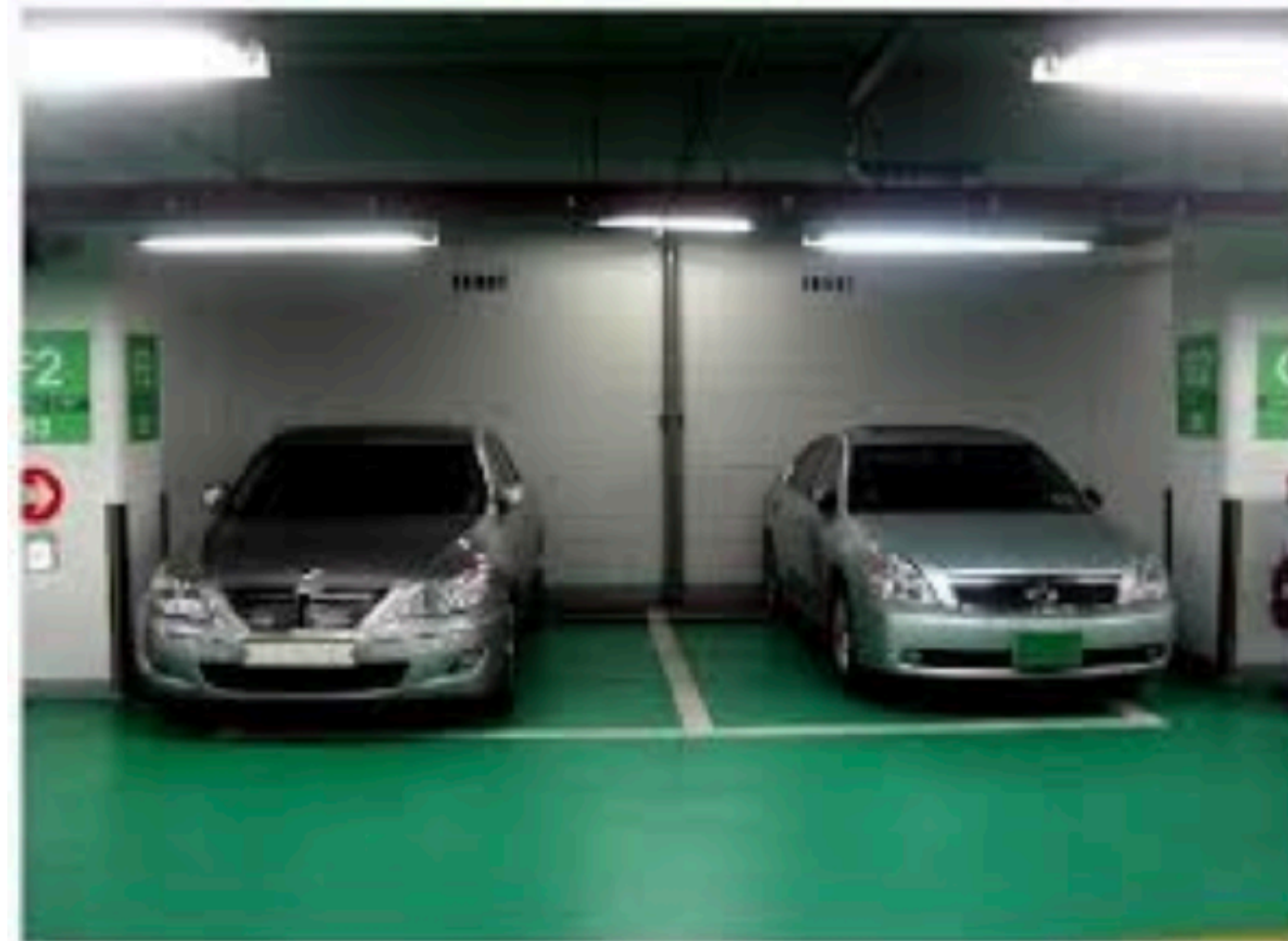
**해당 요소를 안 보이게 한다.**

# display: none vs visibility: hidden



```
<div class="block">block</div>  
<div class="none">none</div>  
<div class="hidden">hidden</div>  
<div class="block">visible</div>
```

**display: none**  
**vs**  
**visibility: hidden**



**background-image**

# Font & Text

몇 가지만 적용해봅시다

**font-size**

**font-family**

**letter-spacing**

**text-align**

**white-space**

# Position

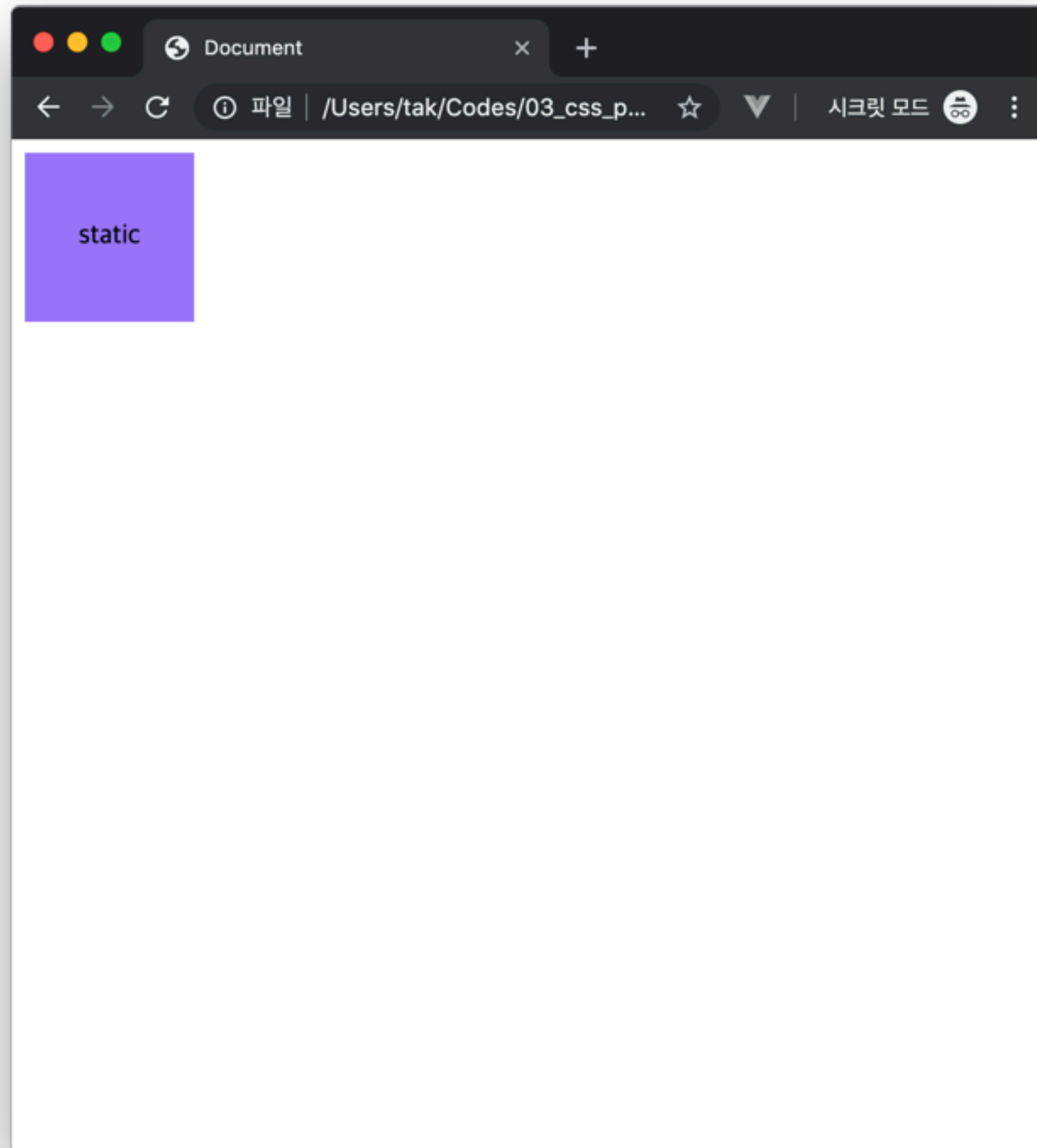
**이제 네모를 위치 시켜보자!**



# **1. static (기본위치)**

**기본적인 요소의 배치 순서에 따라  
위에서 아래로, 왼쪽에서 오른쪽으로 순서에 따라 배치되며  
부모 요소 내에 자식 요소로서 존재할 때는  
부모 요소의 위치를 기준으로 배치된다.**

# 1. static (기본위치)

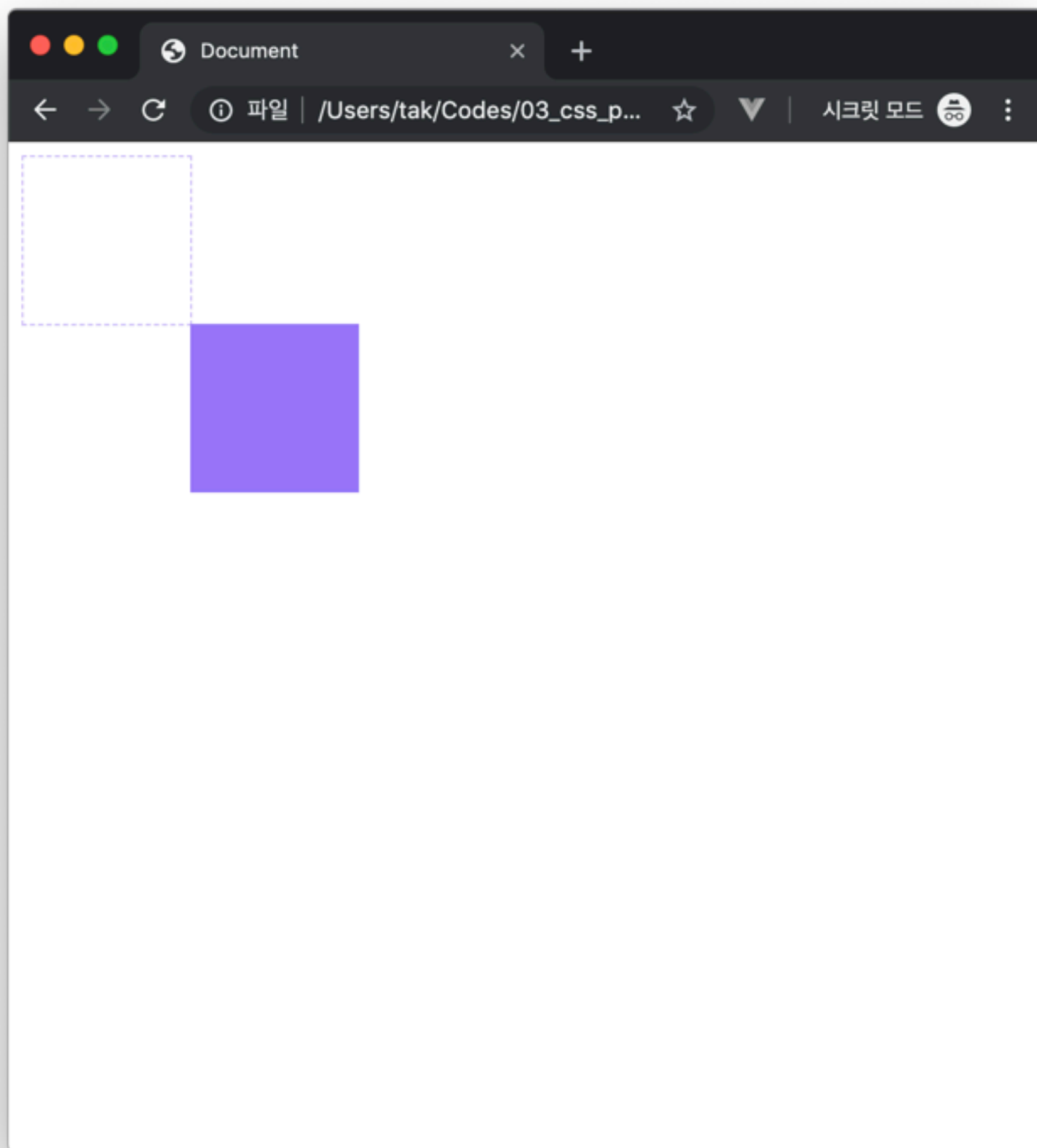


```
div {  
  height: 100px;  
  width: 100px;  
  background-color: #9775fa;  
  color: black;  
  line-height: 100px;  
  text-align: center;  
}
```

## **2. relative (상대위치)**

**기본 위치(static으로 지정 되었을 때의 위치)를 기준으로  
좌표 프로퍼티(top, bottom, left, right)를 사용하여  
위치를 이동 (음수도 가능)**

## 2. relative (상대위치)

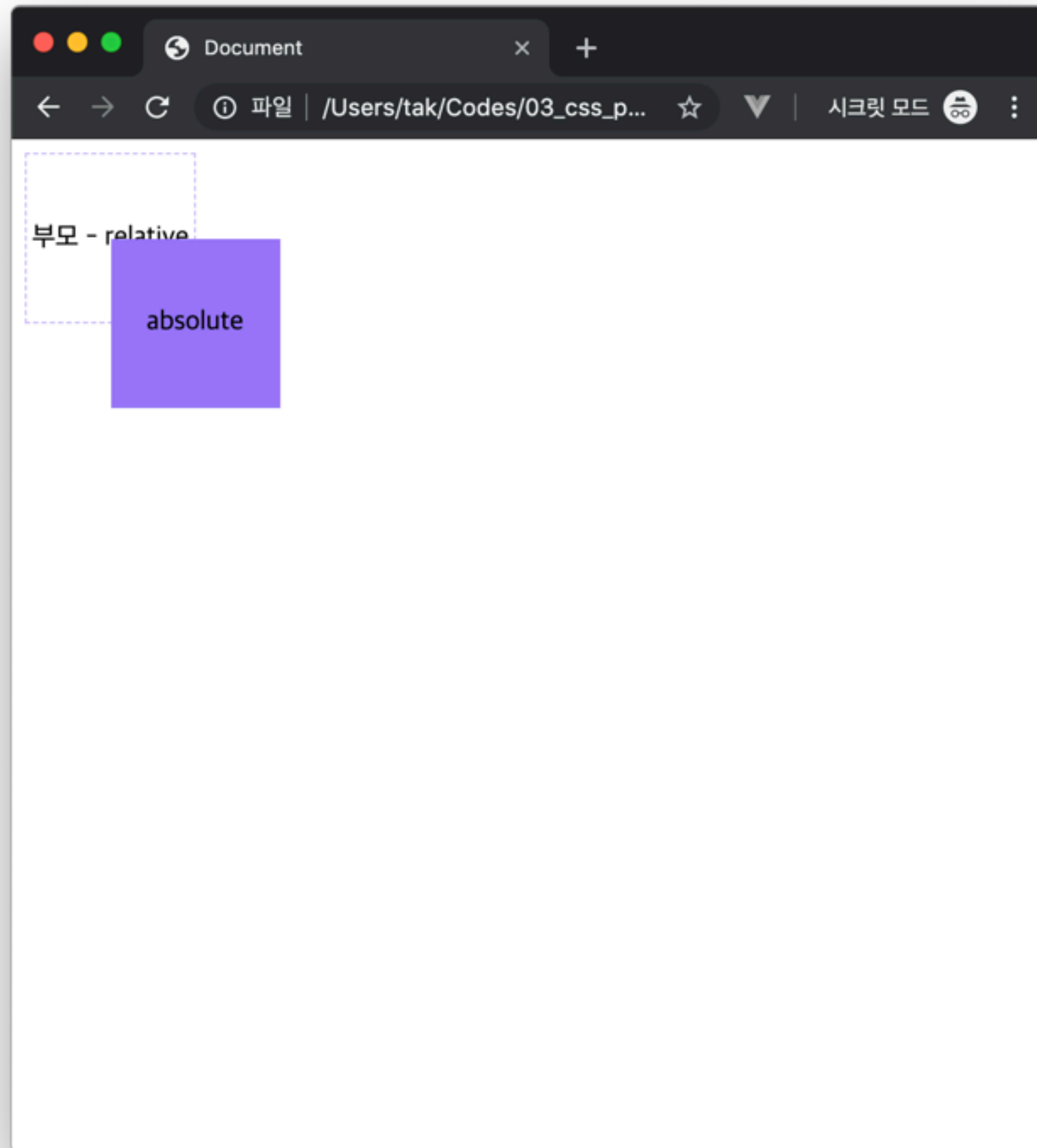


```
.relative {  
  position: relative;  
  top: 100px;  
  left: 100px;  
}
```

### **3. absolute (절대위치)**

**부모 요소 또는 가장 가까이 있는 조상 요소(static 제외)를 기준으로 좌표 프로퍼티(top, bottom, left, right)만큼 이동한다.  
즉, relative, absolute, fixed 프로퍼티가 선언되어 있는  
부모 또는 조상 요소를 기준으로 위치가 결정된다.**

# 3. absolute (절대위치)

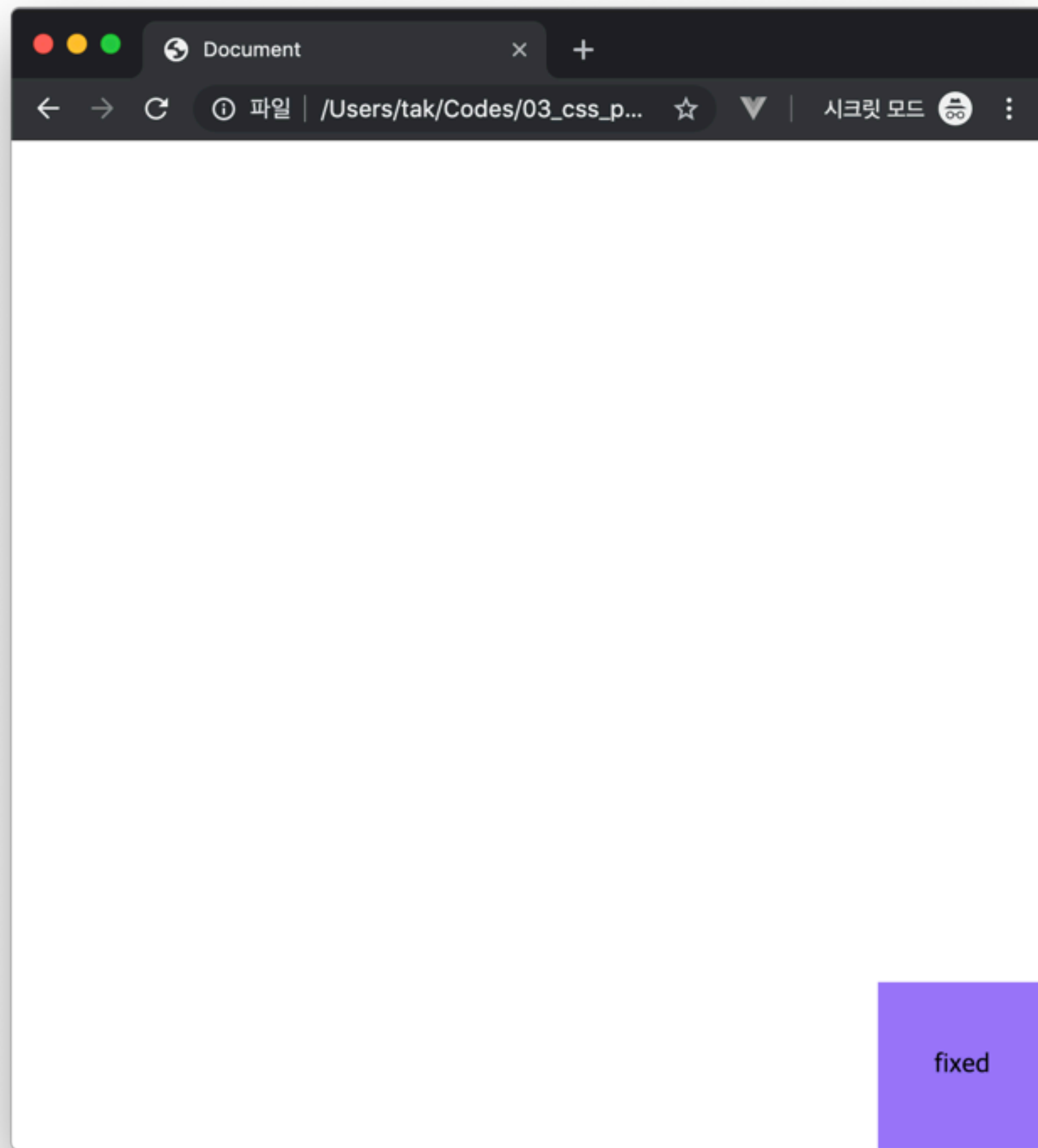


```
.parent {  
  position: relative;  
}  
  
.absolute-child {  
  position: absolute;  
  top: 50px;  
  left: 50px;  
}
```

## **4. fixed (고정위치)**

**부모 요소와 관계없이 브라우저의 viewport를 기준으로  
좌표 프로퍼티(top, bottom, left, right)을 사용하여 위치를 이동시킨다.  
스크롤이 되더라도 화면에서 사라지지 않고 항상 같은 곳에 위치한다.**

## 4. fixed (고정위치)



```
.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
}
```



# 요소정렬

