

# CSS 기본 스타일

## 크기 단위

- px (픽셀)
  - 모니터 해상도의 한 화소인 '픽셀' 기준
  - 픽셀의 크기는 변하지 않기 때문에 고정적인 단위
- %
  - 백분율 단위
  - 가변적인 레이아웃에서 자주 사용

## 크기 단위

- em
  - (바로 위, 부모 요소에 대한) 상속의 영향을 받음
  - 배수 단위, 요소에 지정된 사이즈에 상대적인 사이즈를 가짐
- rem
  - (바로 위, 부모 요소에 대한) 상속의 영향을 받지 않음
  - 최상위 요소(html)의 사이즈를 기준으로 배수 단위를 가짐

## vscode에서 직접해보기

```
<body>
  <ul class="font-big">
    <li class="em">2em</li>
    <li class="rem">2rem</li>
    <li>no class</li>
  </ul>
</body>
```

```
<style>
  .font-big {
    font-size: 36px;
  }
  .em {
    font-size: 2em;
  }
  .rem {
    font-size: 2rem;
  }
</style>
```

## 크기 단위 em vs rem

- 2em

li.em 607.11 × 83.56

Color ■ #000000

Font 72px D2Coding

ACCESSIBILITY

Contrast Aa 21 ✓

Name

Role listitem

Keyboard-focusable ⓧ

- 2em
- 2rem

li.rem 607.11 × 37.33

Color ■ #000000

Font 32px D2Coding

ACCESSIBILITY

Contrast Aa 21 ✓

Name

Role listitem

Keyboard-focusable ⓧ

- 2em
- 2rem
- no class

li 607.11 × 41.78

Color ■ #000000

Font 36px D2Coding

ACCESSIBILITY

Contrast Aa 21 ✓

Name

Role listitem

Keyboard-focusable ⓧ

## 크기 단위 (viewport)

- 웹 페이지를 방문한 유저에게 바로 보이게 되는 웹 콘텐츠의 영역 (디바이스 화면)
- 디바이스의 viewport를 기준으로 상대적인 사이즈가 결정됨
- 예) vw, vh, vmin, vmax

## 크기 단위 (viewport)

```
<body>
  <h1 class="px">px사용</h1>
  <h1 class="vw">vw사용</h1>
</body>
```

```
<style>
  h1 {
    color: black;
    background-color: pink;
  }
  .px {
    width: 200px;
  }

  .vw {
    width: 50vw;
  }
</style>
```

## 색상 단위

- 색상 키워드(`background-color: red;`)
  - 대소문자를 구분하지 않음
  - red, blue, black 과 같은 특정 색을 직접 글자로 나타냄
- RGB 색상(`background-color: rgb(0, 255, 0);`)
  - 16진수 표기법 혹은 함수형 표기법을 사용해서 특정 색을 표현하는 방식
- HSL 색상(`background-color: hsl(0, 100%, 50%);`)
  - 색상, 채도, 명도를 통해 특정 색을 표현하는 방식



## 색상 단위

- 색상 키워드
- RGB 색상
  - '#' + 16진수 표기법
  - rgb() 함수형 표기법
- HSL 색상
  - 색상, 채도, 명도
- a는 alpha(투명도)

```
p { color: black; }  
p { color: #000; }  
p { color: #000000; }  
p { color: rgb(0, 0, 0); }  
p { color: hsl(120, 100%, 0); }  
  
p { color: rgba(0, 0, 0, 0.5); }  
p { color: hsla(120, 100% 0.5);}
```

모두 black !

## CSS 문서 표현

- 텍스트
  - 서체(font-family), 서체 스타일(font-style, font-weight 등)
  - 자간(letter-spacing), 단어 간격(word-spacing), 행간(line-height) 등
- 컬러(color), 배경(background-image, background-color)
- 기타 HTML 태그별 스타일링
  - 목록(li), 표(table)

# CSS Selectors



## 선택자(Selector) 유형

- 기본 선택자
  - 전체 선택자, 요소 선택자
  - 클래스 선택자, 아이디 선택자, 속성 선택자
- 결합자(Combinators)
  - 자손 결합자, 자식 결합자
  - 일반 형제 결합자, 인접 형제 결합자
- 의사 클래스/요소(Pseudo Class)
  - 링크, 동적 의사 클래스
  - 구조적 의사 클래스, 기타 의사 클래스, 의사 엘리먼트, 속성 선택자

## CSS 선택자 정리

- 요소 선택자
  - HTML 태그를 직접 선택
- 클래스(class) 선택자
  - 마침표(.)문자로 시작하며, 해당 클래스가 적용된 항목을 선택
- 아이디(id) 선택자
  - # 문자로 시작하며, 해당 아이디가 적용된 항목을 선택
  - 일반적으로 하나의 문서에 1번만 사용
  - 여러 번 사용해도 동작하지만, 단일 id를 사용하는 것을 권장

## CSS 적용 우선순위 (cascading order)

### 1. 중요도 (Importance) : 사용시 주의

- !important

### 2. 우선 순위 (Specificity)

- 인라인 > id > class, 속성, pseudo-class > 요소, pseudo-element

### 3. CSS 파일 로딩 순서

## CSS 적용 우선순위 (cascading order)

<b>CSS SPECIFISHITY</b> <small>WITH PLANKTON, FISH AND SHARKS</small>			
<b>*</b>  universal selector 0-0-0	<b>div</b>  1 element 0-0-1	<b>li &gt; ul</b>  2 elements 0-0-2	<b>body div ...ul li p a</b>  12 elements 0-0-12
<b>.myClass</b>  1 class 0-1-0	<b>*.myClass</b>  1 universal selector 1 class 0-1-0	<b>[type=checkbox]</b>  1 attribute selector 0-1-0	<b>:only-of-type</b>  1 pseudo-class 0-1-0
<b>li.myClass</b>  1 element 1 class 0-1-1	<b>li[attr]</b>  1 element 1 attribute 0-1-1	<b>li:nth-of-type(3n)~li</b>  2 elements 1 pseudo-class 0-1-2	<b>form input[type=email]</b>  2 elements 1 attribute 0-1-2
<b>li.class:nth-of-type(3n)</b>  1 element 1 class 1 pseudo-class 0-2-1	<b>input[type]:not(.class)</b>  1 element 1 class 1 attribute 0-2-1	<b>ol:nth-child(4n)~ol:nth-child(2n)</b>  10 class/attribute/pseudo-classes 0-10-0	<b>#myDiv</b>  ID Selector 1-0-0
<b>#myDiv li.class a[href]</b>  2 types 2 class/attribute 1 ID Selector 1-2-2	<b>#divitis #myDiv a</b>  2 ID Selectors 1 type selector 2-0-1	<b>style=""</b>  inline style 1-0-0-0	<b>!important</b>  important 1-0-0-0

X-0-0: The number of ID selectors  
 0-Y-0: The number of class selectors, attributes selectors, and pseudo-classes  
 0-0-Z: The number of type selectors and pseudo-elements  
 \*, +, >, ~: Universal selector and combinators do not increase specificity  
 :not(x): Negation selector has no value. Argument increases specificity

## CSS 상속

- CSS는 상속을 통해 부모 요소의 속성을 자식에게 상속한다.
  - 속성(프로퍼티) 중에는 상속이 되는 것과 되지 않는 것들이 있다.
  - 상속 되는 것 예시  
예) Text 관련 요소(font, color, text-align), opacity, visibility 등
  - 상속 되지 않는 것 예시  
예) Box model 관련 요소(width, height, margin, padding, border, box-sizing, display), position 관련 요소(position, top/right/bottom/left, z-index) 등



## CSS 상속 – MDN에서 확인하기

초기값	as each of the properties of the shorthand: <ul style="list-style-type: none"><li><code>margin-bottom: 0</code></li><li><code>margin-left: 0</code></li><li><code>margin-right: 0</code></li><li><code>margin-top: 0</code></li></ul>
적용대상	all elements, except elements with table <code>display</code> types other than <code>table-caption</code> , <code>table</code> and <code>inline-table</code> . It also applies to <code>::first-letter</code> .
상속	no
Percentages	refer to the width of the containing block
Media	visual
Computed value	as each of the properties of the shorthand: <ul style="list-style-type: none"><li><code>margin-bottom</code>: the percentage as specified or the absolute length</li><li><code>margin-left</code>: the percentage as specified or the absolute length</li><li><code>margin-right</code>: the percentage as specified or the absolute length</li><li><code>margin-top</code>: the percentage as specified or the absolute length</li></ul>
Animation type	a <code>length</code>
Canonical order	the unique non-ambiguous order defined by the formal grammar

## CSS 상속 – MDN에서 확인하기

```
<body>
  <p>안녕하세요! <span>테스트</span> 입니다.</p>
</body>
```

```
<style>
  p {
    /* 상속됨 */
    color: red;
    /* 상속 안됨 */
    border: 3px solid black;
  }
  span {
  }
</style>
```

## CSS 상속 – MDN에서 확인하기

- 상속이 되지 않아서 span에는 border가 없음

안녕하세요! 테스트입니다.

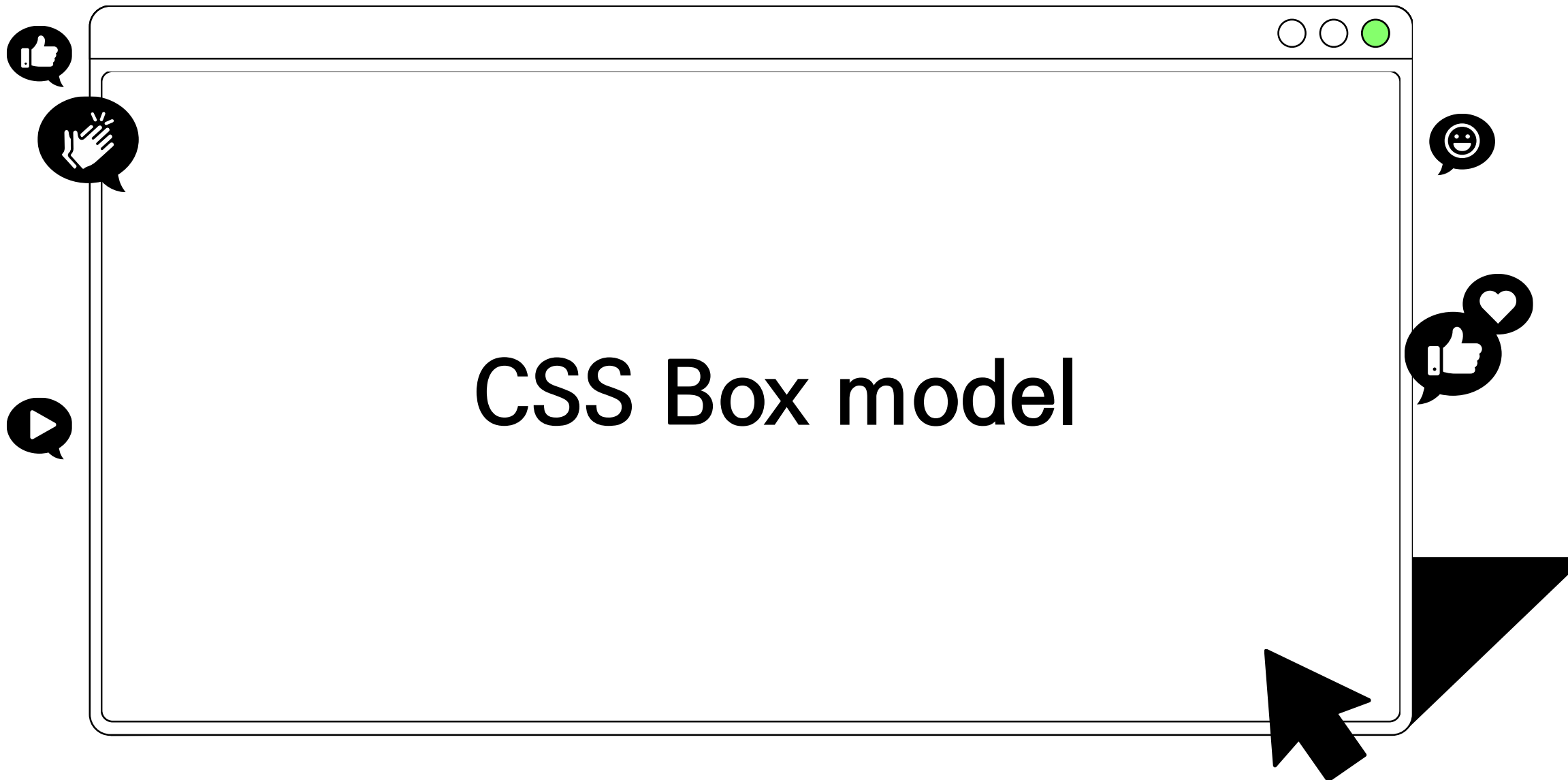
- 상속이 되었다면 span에도 border가 적용되어야 함

안녕하세요! 테스트입니다.

```
<body>
  <p>안녕하세요! <span>테스트</span> 입니다.</p>
</body>
```

```
<style>
  p {
    /* 상속됨 */
    color: red;
    /* 상속 안됨 */
    border: 3px solid black;
  }
  span {
  }
</style>
```

# CSS Box model



## Box model

This is my site

h1 tag in div

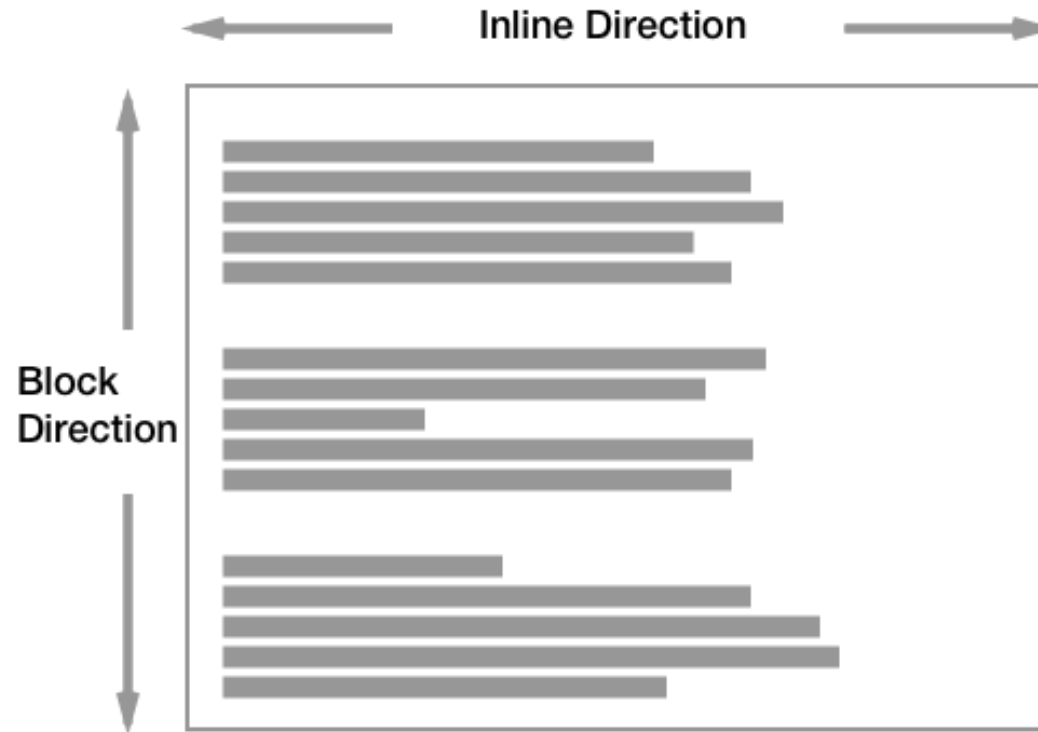
`div#circle` | 50×50



## CSS 원칙 I

모든 요소는 네모(박스모델)이고,  
위에서부터 아래로, 왼쪽에서 오른쪽으로 쌓인다.  
(좌측 상단에 배치)

## CSS 원칙 I



Normal Flow

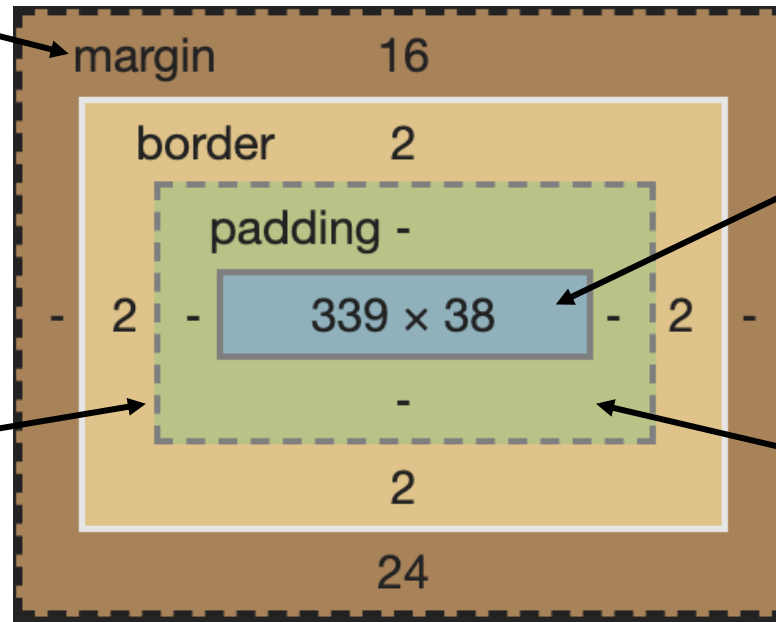
## Box model

- 모든 HTML 요소는 box 형태로 되어있음
- 하나의 박스는 네 부분(영역)으로 이루어짐
  - margin
  - border
  - padding
  - content



Margin  
테두리 바깥의 외부 여백  
배경색을 지정할 수 없다

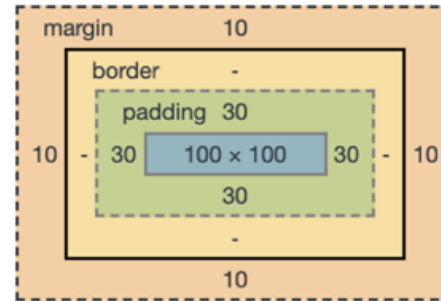
Border  
테두리 영역



Content  
글이나 이미지 등  
요소의 실제 내용

Padding  
테두리 안쪽의 내부 여백  
요소에 적용된 배경색,  
이미지는 padding까지 적용

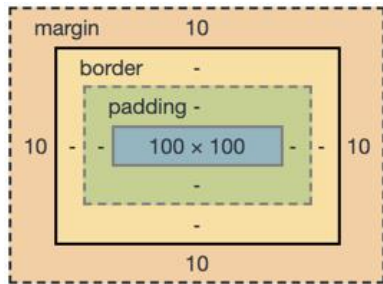
## Box model 구성 (padding)



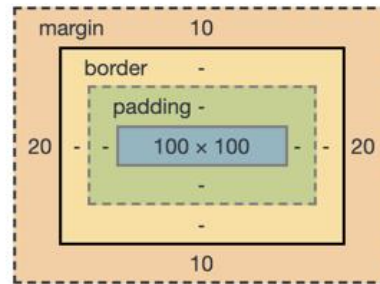
```
.margin-padding {  
  margin: 10px;  
  padding: 30px;  
}
```

상하좌우!

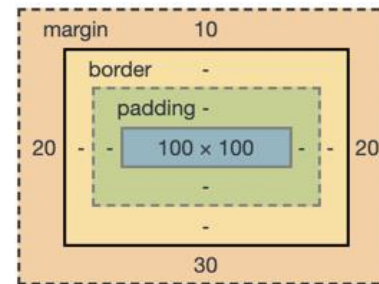
## Box model 구성 (margin/padding)



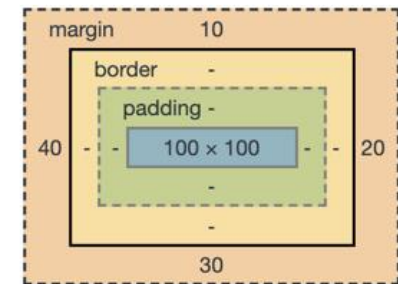
```
.margin-1 {  
  margin: 10px;  
}
```



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

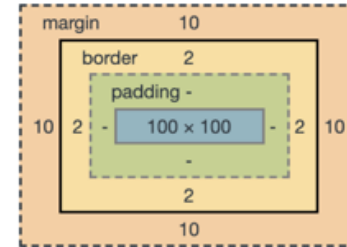


```
.margin-3 {  
  margin: 10px 20px 30px;  
}
```



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

## Box model 구성 (border)

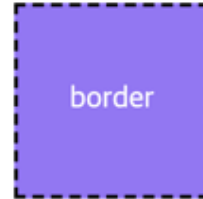


```
.border {  
  border-width: 2px;  
  border-style: dashed;  
  border-color: black;  
}
```

Filter <span>Show all</span>	
▶ 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!

## Box model 구성 (border)



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

```
.border {  
  border: 2px dashed black;  
}
```

---

vscode에서 직접해보기 – box model

이제 박스를 직접 만들어보자

## vscode에서 직접해보기 – box model

```
<body>
  <div class="box1">div</div>
  <div class="box2">div</div>
</body>
```

```
<style>
  .box1 {
    width: 500px;
    border-width: 2px;
    border-color: black;
    border-style: dashed;
    padding-left: 50px;
    margin-bottom: 30px;
  }

  .box2 {
    width: 500px;
    border: 2px solid black;
    padding: 20px 30px;
  }
</style>
```

## vscode에서 직접해보기 – box model





## vscode에서 직접해보기 - box model

```
<body>
  <div class="box">content-box</div>
  <div class="box box-sizing">border-box</div>
</body>
```

```
<style>
  .box {
    width: 100px;
    margin: 10px auto;
    padding: 20px;
    border: 1px solid black;
    color: white;
    text-align: center;
    background-color: blueviolet;
  }

  .box-sizing {
    box-sizing: border-box;
    margin-top: 50px;
  }
</style>
```

## vscode에서 직접해보기 – box sizing



보라색 영역의 너비는 얼마인가?

## vscode에서 직접해보기 – box sizing

```
.box {                                04_box_sizing.html:8
  width: 100px;
  margin: ▶ 10px auto;
  padding: ▶ 20px;
  border: ▶ 1px solid black;
  background-color: blueviolet;
  color: white;
  text-align: center;
}
```



142px…?

## box-sizing

content-box

border-box

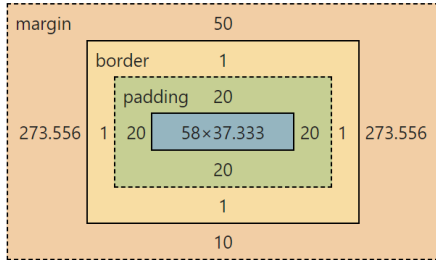
DevTools is now available in Korean!  
Always match Chrome's language | Switch DevTools to Korean  
Don't show again

Elements | Console | Sources | >> | ⚙️

<!DOCTYPE html>  
<html lang="en">  
 <head>...</head>  
 <body>  
 <div class="box">content-box</div>  
 ... <div class="box box-sizing">border-box</div> == \$0

html | body | div.box.box-sizing

Styles | Computed | Layout | Event Listeners | DOM Breakpoints



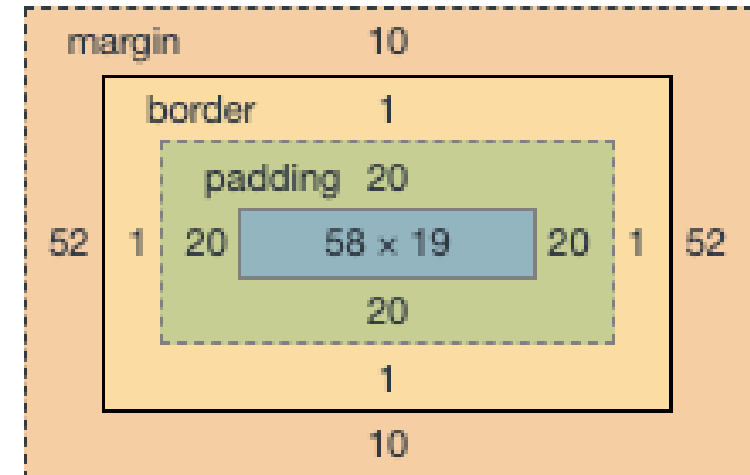
The diagram illustrates the box model with the following dimensions:

- margin: 50 (top), 10 (bottom), 273.556 (left), 273.556 (right)
- border: 1 (top), 1 (bottom), 1 (left), 1 (right)
- padding: 20 (top), 20 (bottom), 20 (left), 20 (right)
- content: 58x37.333

내가 원하는 너비는 100px!

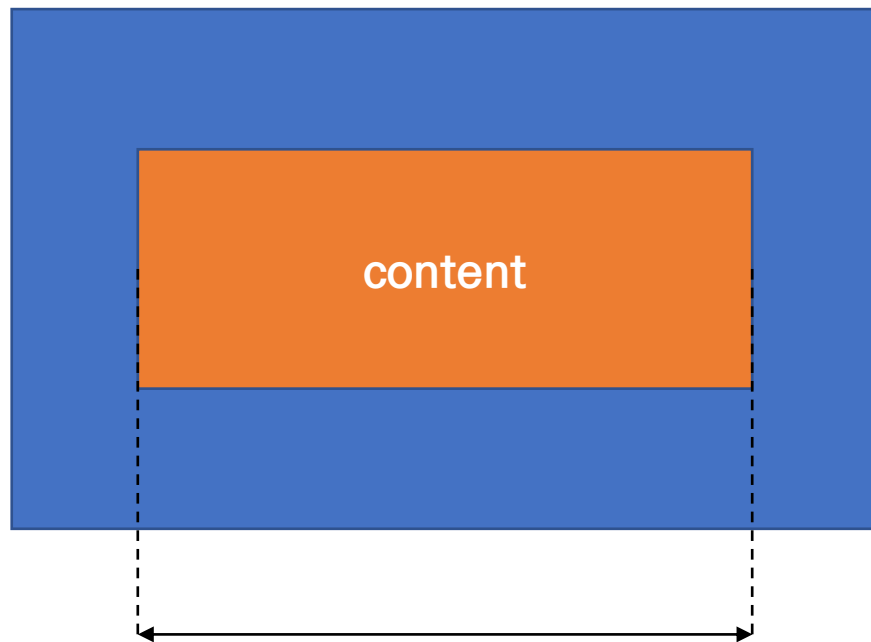
## box-sizing

- 기본적으로 모든 요소의 box-sizing은 content-box
  - Padding을 제외한 순수 contents 영역만을 box로 지정
- 다만, 우리가 일반적으로 영역을 볼 때는 border까지의 너비를 100px 보는 것을 원함
  - 그 경우 box-sizing을 border-box로 설정

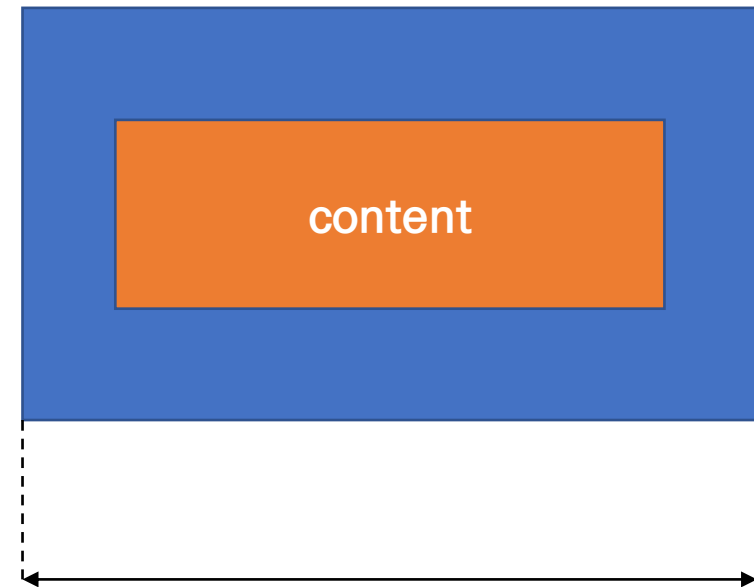


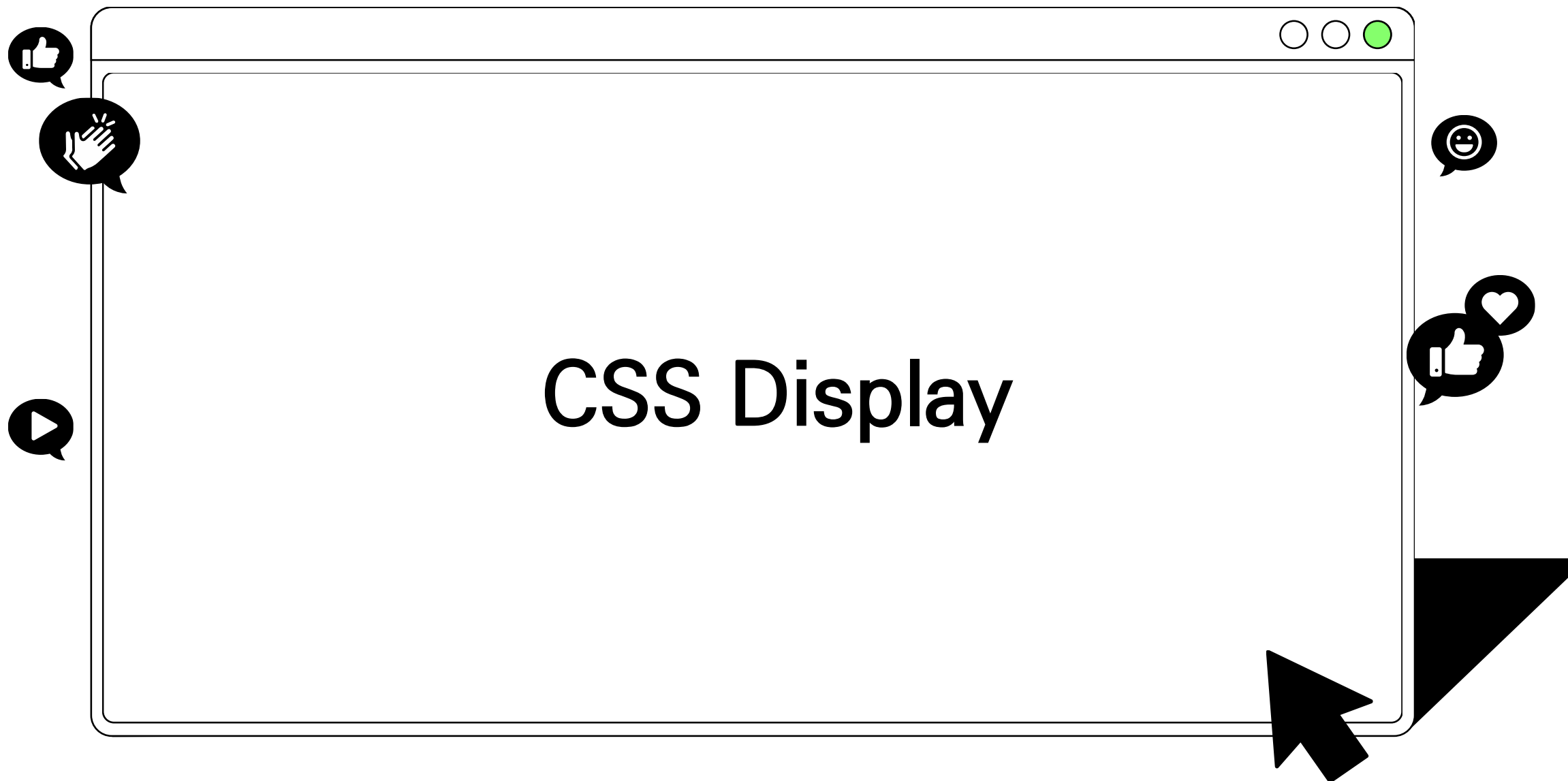
## box-sizing

content-box



border-box



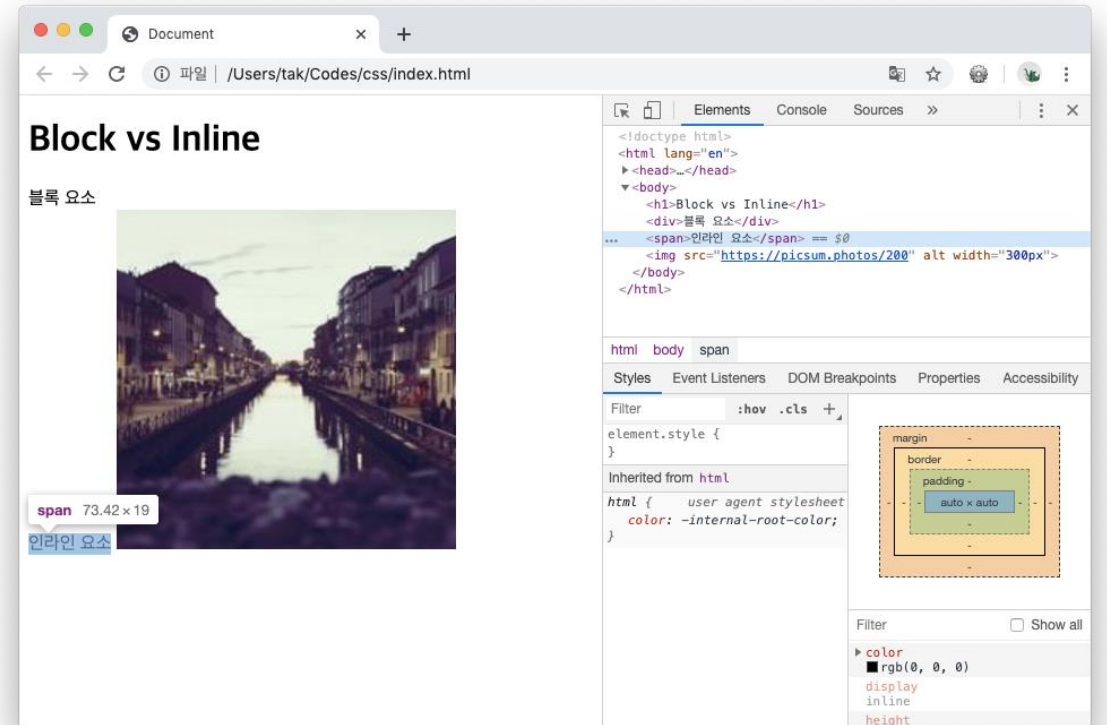
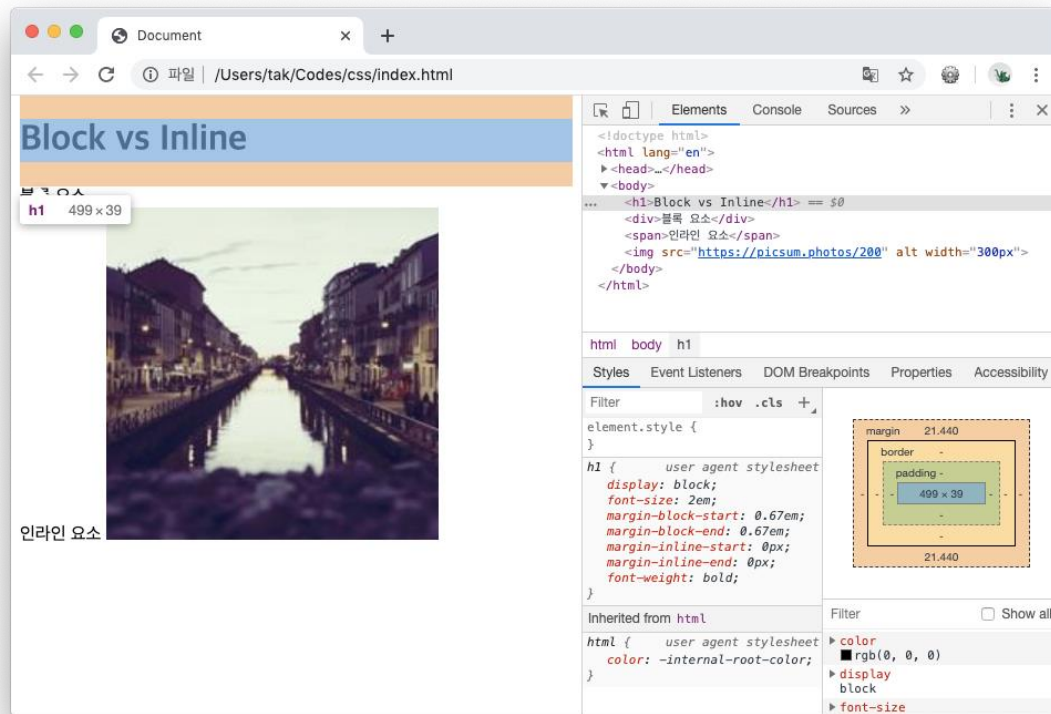


## CSS 원칙 II

- 모든 요소는 네모(박스모델)이고, 좌측상단에 배치.
- `display`에 따라 크기와 배치가 달라진다.



## 인라인 / 블록 요소



‘CSS에서 자세한 내용을 다룹니다’라고 했던 내용!

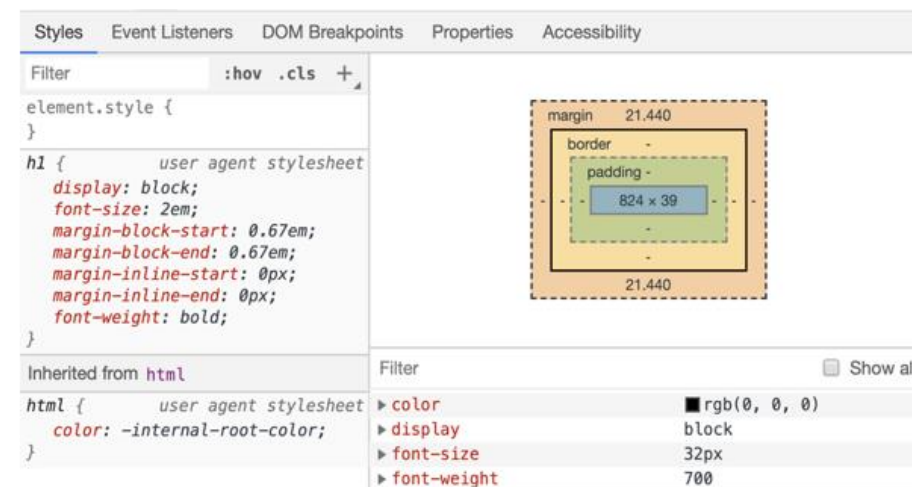
## 대표적으로 활용되는 display

- `display: block`
  - 줄 바꿈이 일어나는 요소
  - 화면 크기 전체의 가로 폭을 차지한다.
  - 블록 레벨 요소 안에 인라인 레벨 요소가 들어갈 수 있음.
- `display: inline`
  - 줄 바꿈이 일어나지 않는 행의 일부 요소
  - content 너비만큼 가로 폭을 차지한다.
  - `width`, `height`, `margin-top`, `margin-bottom`을 지정할 수 없다.
  - 상하 여백은 `line-height`로 지정한다.

## 블록 레벨 요소와 인라인 레벨 요소

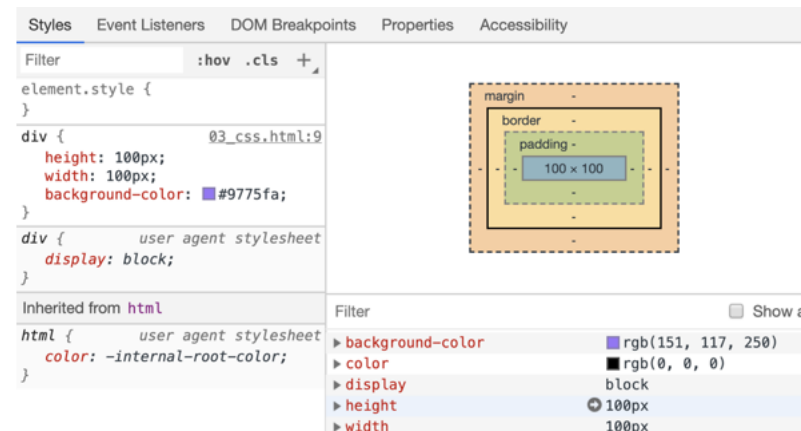
- 블록 레벨 요소와 인라인 레벨 요소 구분 (HTML 4.1까지)
- 대표적인 블록 레벨 요소  
div / ul, ol, li / p / hr / form 등
- 대표적인 인라인 레벨 요소  
span / a / img / input, label / b, em, i, strong 등

## block



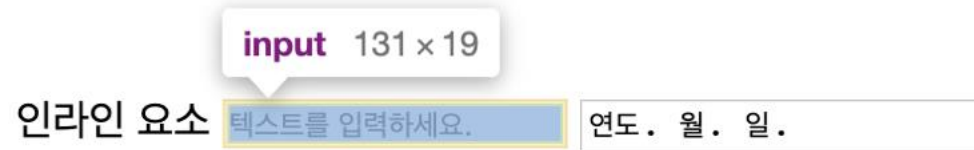
block의 기본 너비는 가질 수 있는 너비의 100%

## block



너비를 가질 수 없다면 자동으로 부여되는 margin

## inline



```
<span>인라인 요소</span>  
<input type="text" placeholder="텍스트를 입력하세요.">  
<input type="date">
```

inline의 기본 너비는 콘텐츠 영역만큼

## 속성에 따른 수평 정렬

**margin-right: auto;**



**text-align : left;**

**margin-left: auto;**



**text-align : right;**

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



**text-align : center;**

## display

- `display: inline-block`
  - block과 inline 레벨 요소의 특징을 모두 가짐
  - inline처럼 한 줄에 표시할 수 있고, block처럼 width, height, margin 속성을 모두 지정할 수 있음
- `display: none`
  - 해당 요소를 화면에 표시하지 않고, 공간조차 부여되지 않음
  - 이와 비슷한 `visibility: hidden`은 해당 요소가 공간은 차지하나 화면에 표시만 하지 않는다.
- 이외 다양한 display 속성은 <https://developer.mozilla.org/ko/docs/Web/CSS/display>