

<h1>

D.com

WEB - 스타디

</h1>

#front-end편



---

3강.CSS 레이아웃

-보기 좋은 떡이 먹기도 좋다!

# 강의 자료 개요

별표 친 페이지는 반드시 읽어주세요!



## 이론

이론 내용입니다. 가벼운 마음으로 읽어주세요!

## 실습

百聞이 不如一打!  
직접 따라해보세요!

## 읽을거리

읽어두면 배운 것을 심층적으로 이해하는데 있어 도움이 되는  
내용입니다! 여유가 있으면 읽어주세요!



# 지금까지 배운 내용은?

---

1. Basic HTML Structure, <h1>, <a>, <div>, <p>, <table>, <li>, <img> ...

2. CSS Selector, Property, Box-Model

#쪽 훑어보고 모르는 내용이 있으면 복습하도록 합시다!

# 오늘의 학습 목표!

---

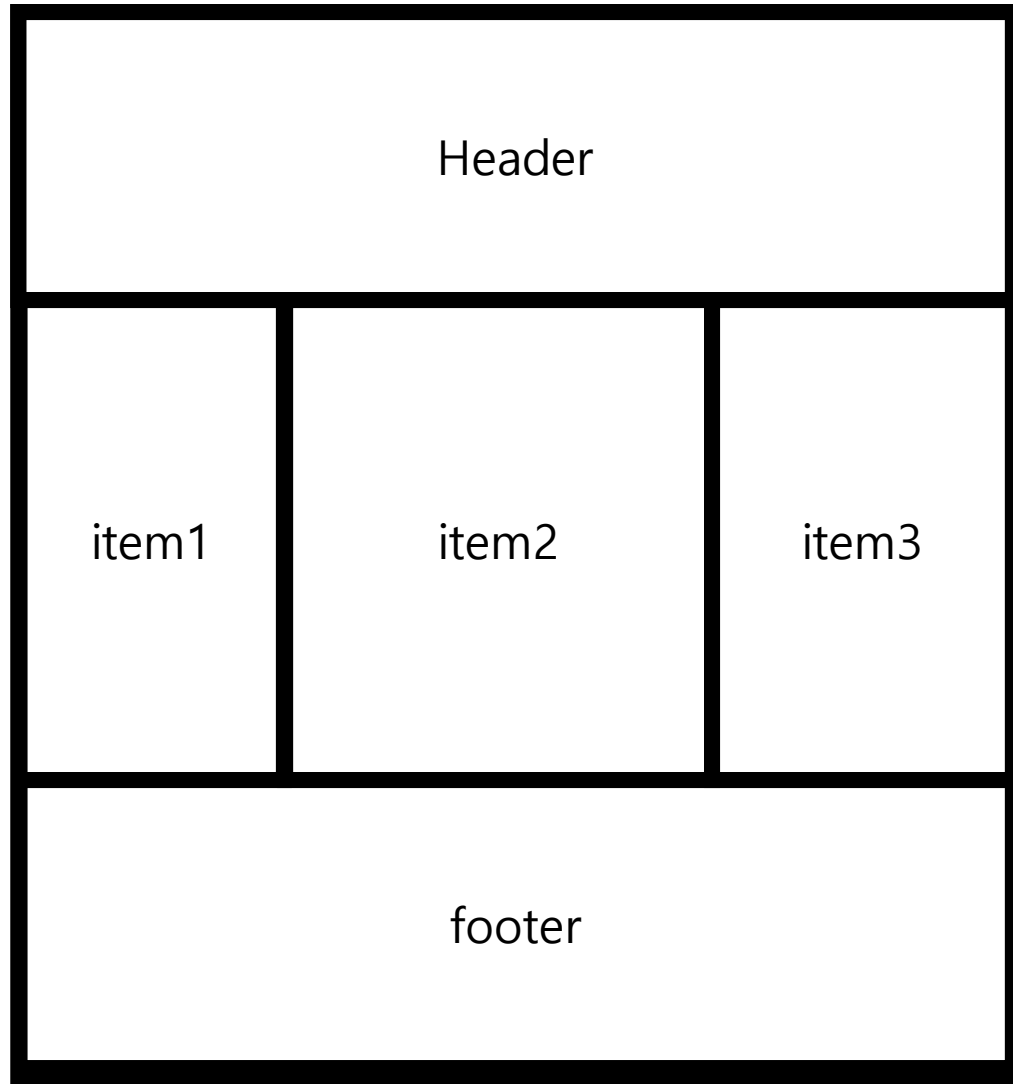
CSS로 Layout을 다루기 위해  
Position, Z-index,  
Display, Flexbox 등을 알아봅시다!

# CSS Layout – Introduction



# CSS Layout

---

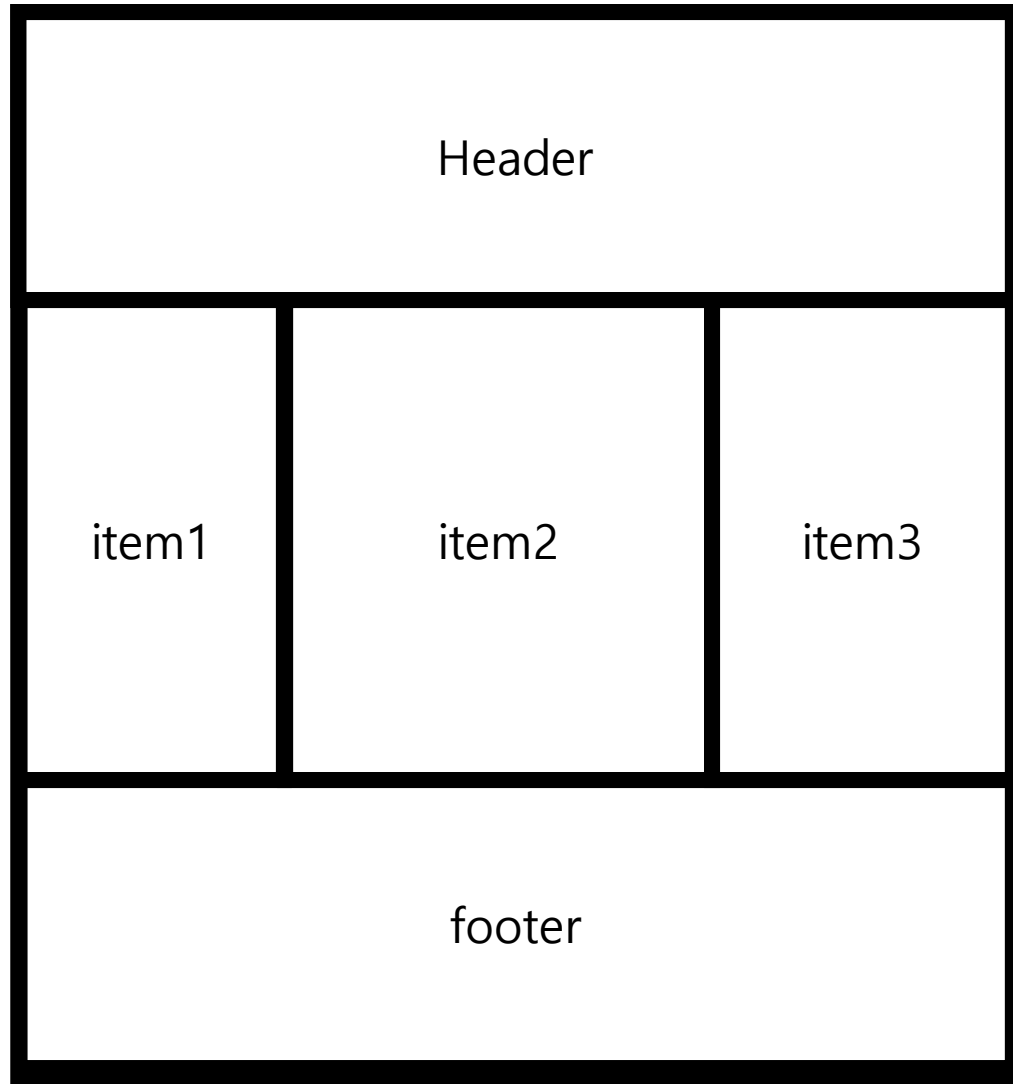


## Layout

“구획을 나누고 적절한 정보를 배치하는 것”

# CSS Layout

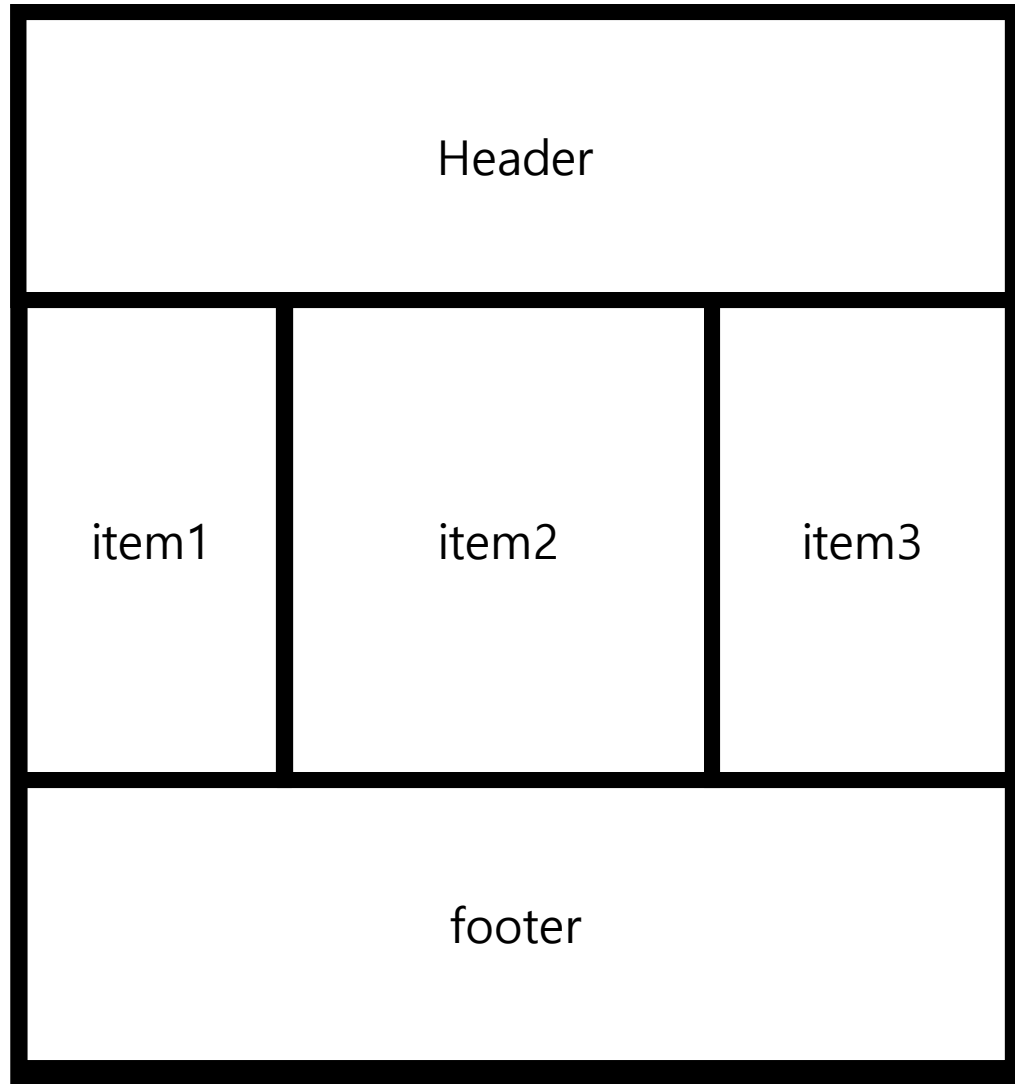
---



“사용자가 이용하기 쉽게  
Layout 구성하는 것은  
매우 중요합니다!”

# CSS Layout

---



“따라서 이번 시간에는  
CSS를 이용하여  
Layout을 구성하는  
방법을 배웁니다!”



# CSS Layout – Box model



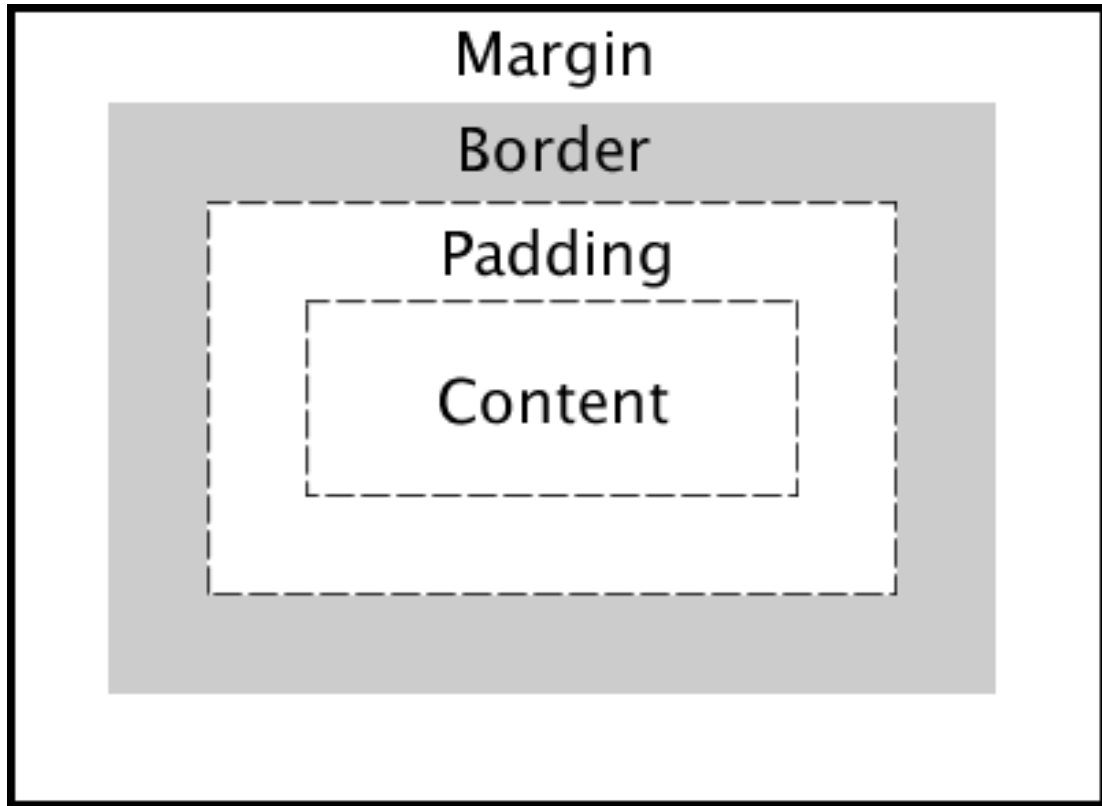
# CSS Layout – Box Model

---

“2강에서 배운 Box-Model 또한 Layout을 조정하는 방법입니다.”

# Box-Model 복습!

---



Q. 각 영역의 역할과  
조정하는 방법은?

#꼭 훑어보고 모르는 내용이 있으면 복습하도록 합시다!

# CSS Layout – Display



# CSS Layout – Display

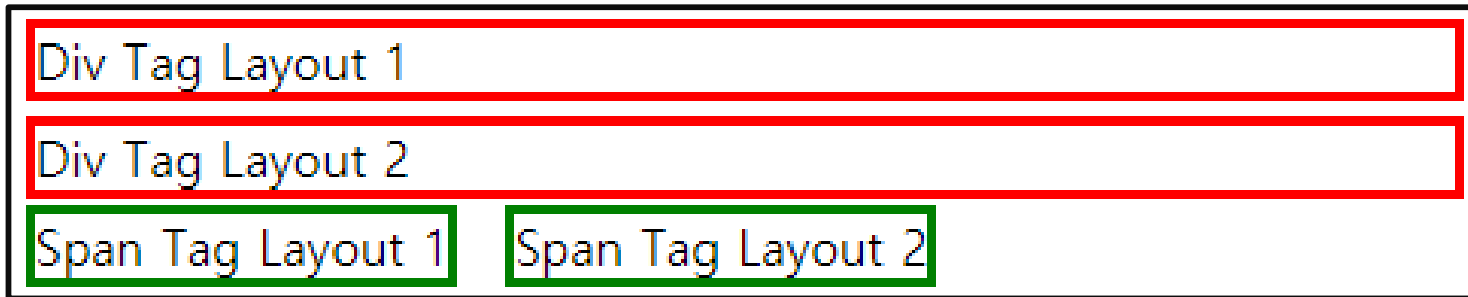
---

2강 내용 중에서 ..

“<div>와 <span>의 차이점은 줄 바꿈의 여부이다!”

왜 그럴까요? 직접 확인해 봅시다!

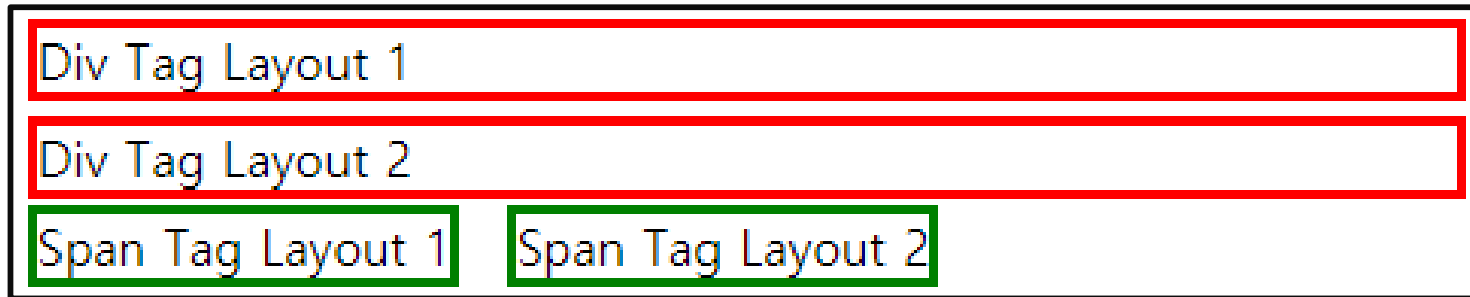
# CSS Layout – Display



〈div〉는 화면 한 줄 전체를 차지하며  
〈span〉는 화면 일부만 차지합니다.

#Border 값을 줘서 확인하면 됩니다!

# CSS Layout - Display



<div>는 화면 한 줄 전체를 차지하며 => **Block Element**

<span>는 화면 일부만 차지합니다. => **Inline Level Element**

#Border 값을 줘서 확인하면 됩니다!

# CSS Layout – Display

---

이처럼 HTML Element는 크게 두가지 속성으로 나뉩니다!

## Inline and Block

“줄을 바꾸지 않고 다른 요소와 함께  
한 행에 위치하려 합니다!”

<a> <br> <button> <img>  
<input> <span> <strong>  
<sub> <sup> <textarea>

...

“한 줄을 완전히 차지하려 합니다!”

<canvas> <div> <footer>  
<form> <h1> <header>  
    <ol> <p> <section>  
    <table> <ul> <video>

...



# CSS Layout – Display

---

또한 두 속성은 이러한 차이점을 가지고 있습니다.

## Inline vs Block

1. 상, 하단 Margin 속성이 적용되지 않습니다.  
상 하단 여백은 line-height 속성으로 적용합니다.

2. Width, Height 속성이 적용되지 않습니다.  
내부 요소의 부피에 따라 결정됩니다.

1. 기본적으로 **Width 100%**로 정의되어 있습니다.

2. Margin, Width, Height 속성 모두 **적용 가능**합니다!

#Margin, Width, Height가 변경되지 않는다면 Inline인지 의심해 주세요!

# CSS Layout – Display

---

그렇다면 이 속성을 변경시킬 수 있나요?

YES!

```
div{  
  display: inline;  
}
```

=> <div>의 Block 속성을 Inline 속성으로 변경!

#Inline속성을 Block 속성으로 변경시키는 것도 당연히 가능합니다!

# CSS Layout – Display

---

Inline-block 속성도 있습니다.

1. **Inline** 속성처럼 Element를 한 줄에 같이 표현해주며
2. Margin, Width, Height 속성 적용이 가능합니다.

```
p{  
    display: inline-block;  
}
```

# CSS Layout – Position



# CSS Layout – Position

---

Position은 Element의 **위치**를 **결정**하는 속성입니다.  
Static(기본값), Relative, Fixed, Absolute  
4가지 속성이 존재합니다!

# CSS Layout – Position

---

## Static

기본값이며, Element를 위에서 아래로  
순서대로 배열시켜줍니다!

# CSS Layout – Position

---

## Relative

**상대적인 위치를 결정할 수 있습니다.**

**따라서 Static 속성과 다르게**

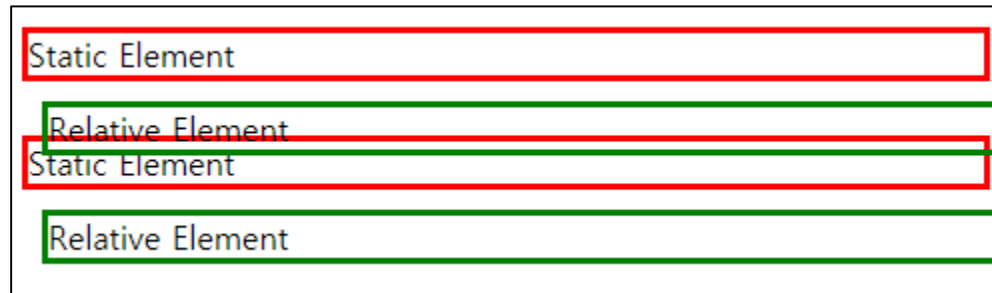
**Top, right, bottom, left 속성  
지정이 가능합니다!**

# CSS Layout – Position

## Static, Relative Example

```
.static{  
border: 3px solid red;  
}  
.relative{  
position: relative;  
border: 3px solid green;  
left: 10px;  
top: 10px;  
}
```

```
<div class="static">Static Element</div>  
<div class="relative">Relative Element</div>  
<div class="static">Static Element</div>  
<div class="relative">Relative Element</div>
```



#Relative 속성이 다른 Element에는 영향을 주지 않는다는 점에 주목해주세요!



# CSS Layout – Position

## Fixed

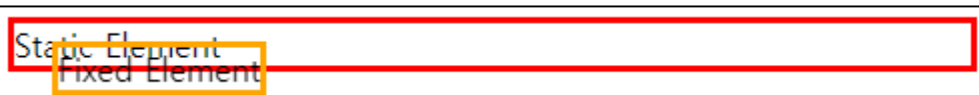
**절대적인 위치를 결정할 수 있습니다.**

#Fixed 속성은 스크롤로 화면을 내려도 사라지지 않고 같은 위치를 고수합니다!

#따라서 상단 메뉴 바 등을 구현할 때 유용합니다!

```
.fixed{  
  position: fixed;  
  border: 3px solid orange;  
  top: 20px;  
  left: 30px;  
}
```

```
<div class="static">Static Element</div>  
<div class="fixed">Fixed Element</div>
```



Static Element  
Fixed Element

# CSS Layout – Position

---

## Absolute

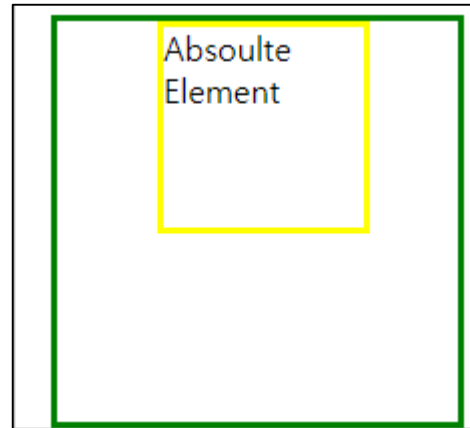
Static 속성이 아닌  
부모 Element에 대해 **상대적인 위치**를  
결정합니다.

#만약 Static 속성이 아닌 부모 Element가 없다면 위치는 <body>에 대해 결정됩니다.

# CSS Layout – Position

## Absolute Example

```
<div class="relative">  
<div class="absolute">Absoulte Element</div>  
</div>
```



```
.relative{  
  position: relative;  
  border: 3px solid green;  
  left: 10px;  
  top: 10px;  
  width: 200px;  
  height: 200px;  
}  
  
.absolute{  
  position: absolute;  
  border: 3px solid yellow;  
  width: 100px;  
  height: 100px;  
  left: 50px;  
}
```

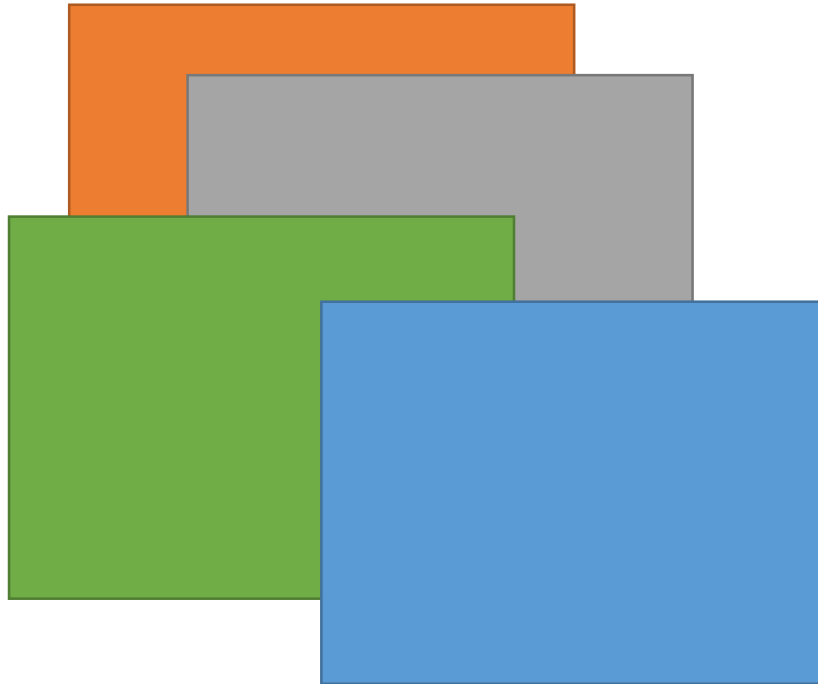
# CSS Layout Z-index



# CSS Layout Z-index

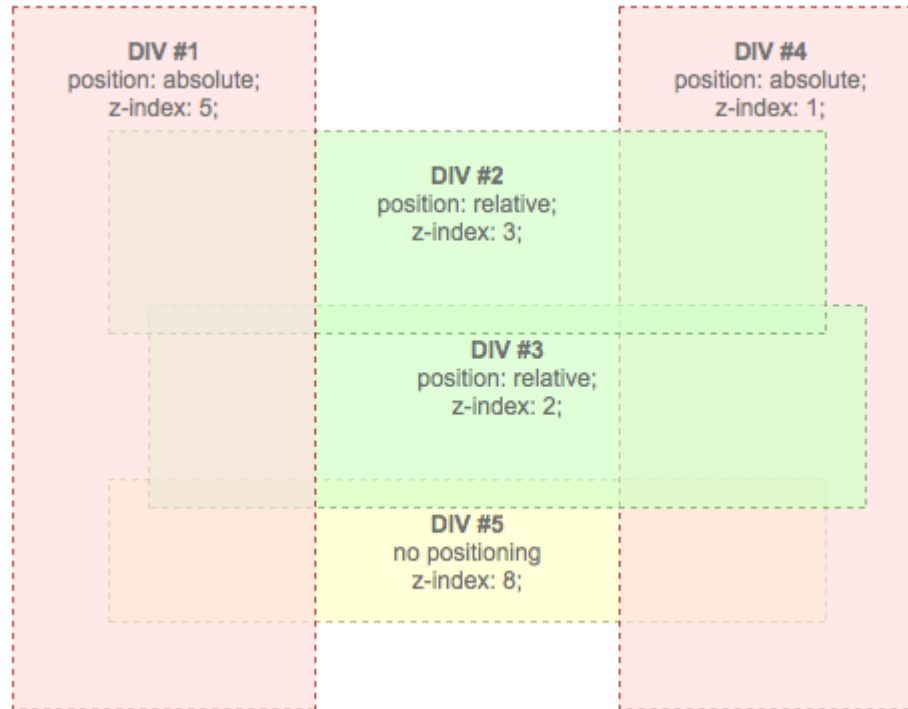
---

Z-index는 Element의 수직 위치  
즉, 배치 순서를 결정합니다!



# CSS Layout Z-index

Z-index가 클수록 위에(우선적으로) 표시됩니다.



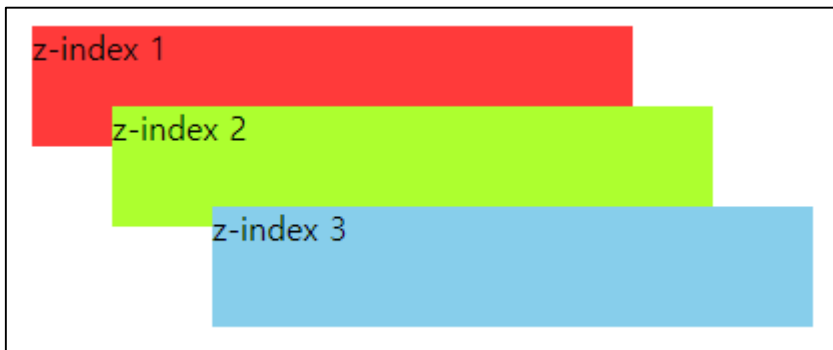
#Z-index 속성은 Static 속성이 아닌 Element에 적용 가능합니다!!

# CSS Layout Z-index

## Z-index Example

```
<div id="zindex1"> z-index 1</div>  
<div id="zindex2"> z-index 2</div>  
<div id="zindex3"> z-index 3</div>
```

```
#zindex1 {  
  position: absolute;  
  z-index: 1;  
}  
#zindex2 {  
  position: absolute;  
  z-index: 2;  
}  
#zindex3 {  
  position: absolute;  
  z-index: 3;  
}
```



#주요 코드만 표기하였습니다.

# CSS Layout Float





# CSS Layout – Float

---

Float는 한컴, 워드에서 지원하는  
**정렬**과 유사한 기능입니다!



# CSS Layout – Float

## Float으로 정렬하는 원리는 다음과 같습니다!

예를 들어 <div>는 block 이므로 한 줄 전체를 차지해야 합니다. 하지만 float 속성을 주면 요소는 다음에 올 객체를 위해 **자신의 너비를 최소화**합니다.

```
<div id="right">float left</div>  
<div id="left">float right</div>
```

```
#right{  
border: 3px solid red;  
float: right;  
}  
#left{  
border: 3px solid blue;  
float: left;  
}
```

float right

float left

# CSS Layout – Float

## Float 응용!

Float도 레이아웃을 구성하는데 유용하게 사용됩니다!  
특히 **고정된 크기**의 Element를 왼쪽 혹은 오른쪽에 구성하는데 유용합니다.

#다음과 같은 레이아웃이 대표적인 Float 응용 사례입니다!



# 오늘의 과제!



여러분은 졸업 후 놀랍게도 창업(차칸집)을 하게 되었습니다. 이래 봐도 공학과 출신인 여러분은 직접 랜딩페이지를 개발해보기로 합니다. 행운을 빕니다!

**#배우지 않은 내용들도 적극 활용하세요!**

# 과제 Example!



치킨은 과학입니다.

웹 스터디 홈페이지에서 확인해주세요!

<https://facerrain.github.io/>

**#배우지 않은 내용들도 적극 활용하세요!**

# 과제 설명!

---



예제처럼 하나의 페이지를 스크롤하는 형식이며,  
상단에는 **화면에 고정된 메뉴바(navigation)**가 존재 해야하며,  
2개 이상의 **주제별 Section**이 있어야합니다.  
이외 페이지의 주제, 디자인은 모두 **자유**입니다.  
(주제Ex. 회사, 게임, 영화 ...)

# 과제 제출!

---



개인 Git의  
“2019\_Summer\_Web\_Study”에

작업한 파일들을 “3.Homepage” 폴더로 모아  
Front-End-Assignment 폴더 안에 Push

# 읽을 거리



**랜딩 페이지란 무엇인가요?  
좋은 랜딩 페이지는 어떻게 만들죠?**

<https://ppss.kr/archives/65972>



# 다음시간!

---

JavaScript의 기초적인 문법과  
DOM에 대해 배웁니다.



**The end.**