

<목차>

- CSS Layout
 - Float
 - Flexbox
- Bootstrap
 - Bootstrap grid system
- Responsive web

[CSS layout techniques]

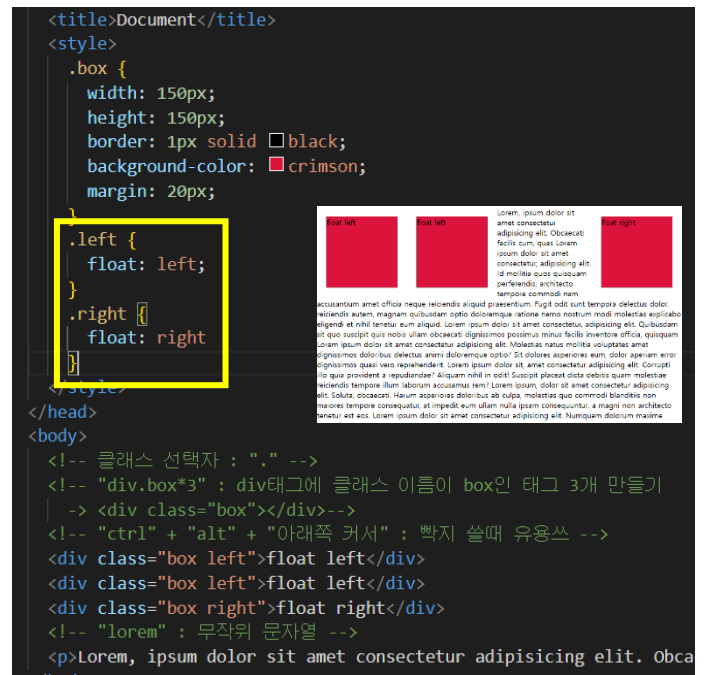
- **Display:** none, inline, block, flex
- **Position:** static, relative, absolute(부모: relative), fixed, sticky
- **Float :**
- **Flexbox**
- **Grid**

<Float>

: 박스를 왼쪽 혹은 오른쪽으로 이동시켜 텍스트를 포함한 인라인 요소들이 주변을 wrapping 하도록 함

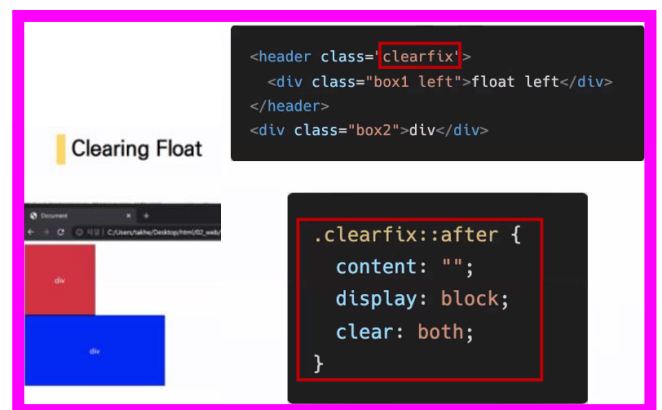
: 요소가 Normal flow를 벗어나도록 함

- **Float 속성**
 - **none:** 기본값
 - **left:** 요소를 왼쪽으로 띄움
 - **right:** 요소를 오른쪽으로 띄움



● Clearing Float ★★★★★

: float 속성을 적용한 요소의 부모요소에 **clear**를 사용



: clearfix 태그 뒤에 내용이 빈(**content: ""**) 블록을(**display: block**) 만들고, float: left, float: right를 초기화 시키겠다는 뜻이다.

이렇게 작성 해주면 clearfix태그 뒤에 다른 내용을 작성하면 위로 떨어 올라가지 않고 clearfix태그 다음에 위치하게 된다.

여기서 display: table 대신 display: block을 쓰는 것도 가능하다.

clear: left 를 쓰면 float: left 를, clear: right 를 쓰면 float: right 를 초기화 한다는 뜻인데

굳이 하나하나 신경쓰기 싫다면 항상 **clear: both** 를 사용해도 상관이 없다.

overflow 속성 사용

위와 마찬가지로 float 속성을 가진 요소의 부모 요소에

overflow: hidden 또는 **overflow: auto** 를 적용시켜주면 된다.

overflow: auto 사용시 자식의 너비가 부모의 너비보다 크다면 가로 스크롤바가 생기고, overflow: hidden 의 경우 넘치는 부분이 잘리기 때문에 사용하지 않는 것이 좋음.

```

display: inline;
}

.detail {
border: 2px dashed skyblue;
text-align: right;
display: inline;
float: right;
margin: 0px;
}

img {
float: left;
}

.clearfix:after {
/* overflow: auto; */
content: "";
display: block;
clear: both;
}
}

</style>
</head>
<body>
<div class="clearfix">
<div class="title">HPHK</div>
<div class="detail">해피해킹을 소개합니다!</div>
</div>

<span>
기업의 기술적 변화를 함께 합니다.
B2B IT/SW 교육

```

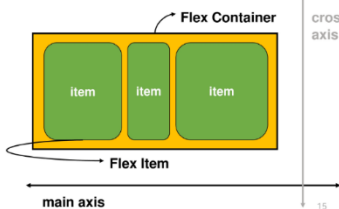
[Flex]

왜 flexbox를 사용해야 할까?

: 이전까지 Normal Flow를 벗어나는 수단은 Float 혹은 Position

하기 어려웠던 것? : (수동 값 부여 없이) 1. 수직 정렬 2. 아이템의 너비와 높이 혹은 간격을 동일하게 배치

● CSS Flexible Box Layout



: 행과 열 형태로 아이템들을 배치하는 1차원 레이아웃 모델

- 축

- main axis (메인 축)
- cross axis (교차 축)

- 구성요소

- flex **Container** (부모 요소)
- flex **Item** (자식 요소)

: **flex를 부모요소에 적용시켜야 함(first!)**

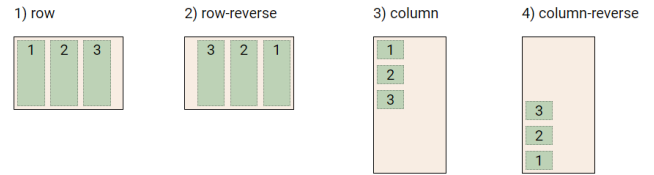
- **display : flex** (box 형태로 컨테이너)

- display : **inline-flex** (box가 아이템 크기에 맞게 줄어듦)

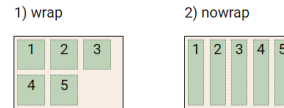
● flex 속성



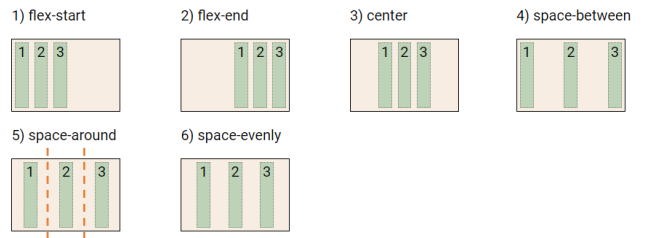
flex-direction



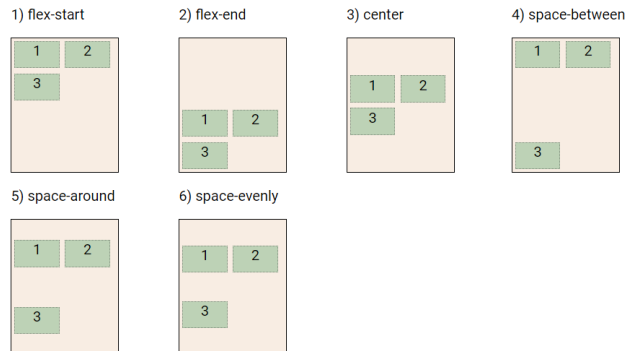
flex-wrap



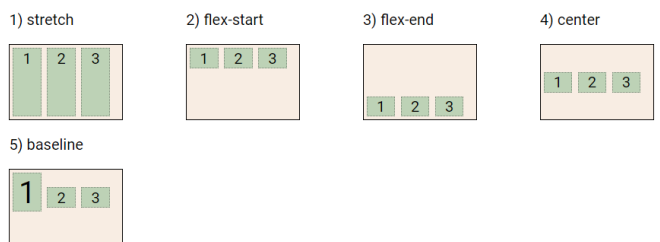
justify-content



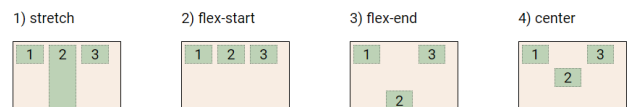
align-content



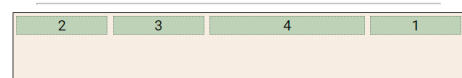
align-items



align-self



flex-grow & order



1. 배치 설정

- flex-direction

: main axis 기준 방향 설정 (꼬지)

: 역방향의 경우 HTML 태그 선언 순서와 시각적으로 다르니 유의 (웹 접근성에 영향)

- flex-wrap

: 아이템이 컨테이너를 벗어나는 경우 해당 영역 내에 배치되도록 설정

: 즉, 기본적으로 컨테이너 영역을 벗어나지 않도록 함

◆ nowrap (기본 값): 한 줄에 배치

◆ wrap : 넘치면 그 다음 줄로 배치

- flex-flow

: flex-direction 과 flex-wrap의 shorthand

: flex-direction 과 flex-wrap에 대한 설정 값을 차례로 작성

Ex) flex-flow: row nowrap;

2. 공간 나누기

- justify-content : main axis를 기준으로 공간 배분

◆ flex-start

◆ flex-end

◆ center

◆ space-between

◆ space-around (ex. 1 : 2 : 2 : 1)

◆ space-evenly (ex. 1 : 1 : 1 : 1)

- align-content : cross axis를 기준으로 공간 배분

(아이템이 한 줄로 배치되는 경우 확인할 수 없음)

◆ stretch : cross axis로 가득 채우기

◆ flex-start

◆ flex-end

◆ center

◆ baseline

3. 정렬

- align-items: 모든 아이템을 cross axis 기준으로

- align-self

: 개별 아이템을 cross axis 기준으로 정렬

: 주의! 컨테이너에 적용하는 것이 아니라 개별 아이템에 적용

■ stretch (기본 값): 컨테이너를 가득 채움

■ flex-start: 위

■ flex-end: 아래

■ center: 가운데

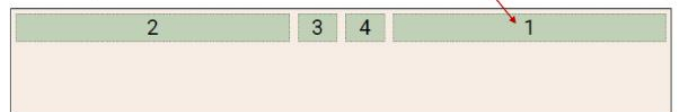
■ baseline: 텍스트 baseline에 기준선을 맞춤

4. 기타 속성

- flex-grow : 남은 영역을 아이템에 분배

- order : 배치 순서

```
<div class="flex_item" grow-1 order-3">1</div>
<div class="flex-item" grow-1">2</div>
<div class="flex-item" order-1">3</div>
<div class="flex-item" order-2">4</div>
```

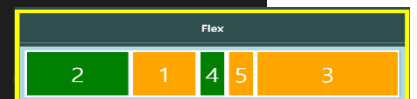


```
<style>
.flex-container {
  /* 정렬하려고하는 부모 요소에 선언 */
  display: flex;
}
```

```
.item1 {
  align-self: flex-start;
  order: 0;
  /* 주축에서 남은 공간을 항목들에게 분배 */
  flex-grow: 1;
}

.item2 {
  align-self: center;
  order: -1;
  flex-grow: 2;
}

.item3 {
  align-self: flex-end;
  order: 1;
  flex-grow: 3;
}
</style>
</head>
<body>
<div class="base">
  <h2>Flex</h2>
  <hr>
  <div class="box flex-container">
    <div class="item1">1</div>
    <div class="item2">2</div>
    <div class="item3">3</div>
```



<Bootstrap>

#부트스트랩 실행(CDN통해서 가져오기)

(CDN: content delivery(distribution) network

: 컨텐츠(CSS, JS, Image, Text등)을 효율적으로 전달하기 위해 여러 노드에 가진 네트워크에 데이터를 제공하는 시스템

- 개별 end-user의 가까운 서버를 통해 빠르게 전달 가능(지리적 이점)
- 외부 서버를 활용함으로써 본인 서버의 부하가 적어짐)

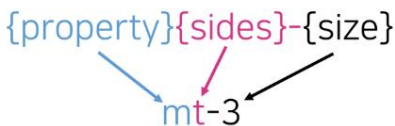
*닫는 head 태그 위에

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-gH2yJqKdNHPEq0n4Mqa/HGKIhSkIHeL5AyhkYV8i59U5AR6csBvApHHNl/v11Bx" crossorigin="anonymous">
```

*닫는 body 태그 위에

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-A3rJD856KowSb7dwlZdYEkO39Gagi7vlsF0jrRAoQmDCKtQBHUuLZ9AsSv4jD4Xa" crossorigin="anonymous"> </script>
```

[Spacing] (margin and padding)



```
<div class="mt-3 ms-5">bootstrap-spacing</div>
```

Where *property* is one of:

- **m** - for classes that set **margin**
- **p** - for classes that set **padding**

	t	top
	b	bottom
	s	left
	e	right
	x	left, right
	y	top, bottom
m		margin
p		padding

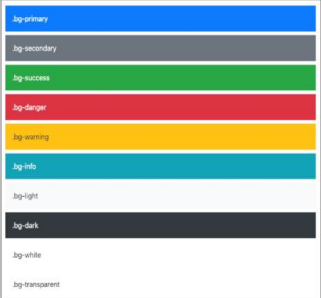
class name	rem	px
m-1	0.25	4
m-2	0.5	8
m-3	1	16
m-4	1.5	24
m-5	3	48

```
</style>
</head>
<body>
  <h2>Spacing</h2>
  <!-- "div.m-4.box" -->
  <div class="m-4 box">margin 4</div>
  <!-- div.mt-3.ms-5 -->
  <div class="mt-3 ms-5 box">margin top3 ms-5</div>
  <!-- div.mx-auto.box -->
  <div class="mx-auto box">mx-auto/가운데정렬</div>
  <div class="ms-auto box">ms-auto</div>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js">
</body>
</html>
```



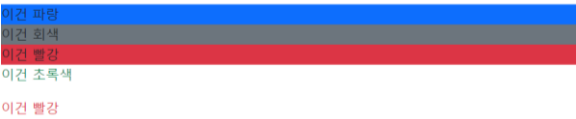
color

```
:root {
  --primary: #007bff;
  --secondary: #6c757d;
  --success: #28a745;
  --info: #17a2b8;
  --warning: #ffc107;
  --danger: #dc3545;
  --light: #f8f9fa;
  --dark: #343a40;
}
```



```
<hr>
<h2>Color</h2>
<div class="bg-primary">이건 파랑</div>
<div class="bg-secondary">이건 회색</div>
<div class="bg-danger">이건 빨강</div>
<p class="text-success">이건 초록색</p>
<p class="text-danger">이건 빨강</p>
```

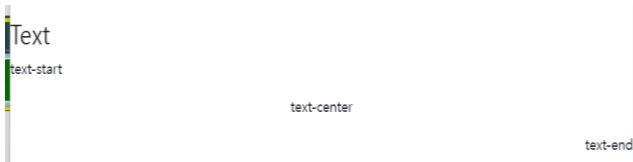
Color



```

<hr>
<h2>Text</h2>
<p class="text-start">text-start</p>
<p class="text-center">text-center</p>
<p class="text-end">text-end</p>

```



```

<a href="#" class="text-decoration-none text-dark">non-underline-link</a>

```

non-underline-link

```

<p class="fw-bold">Bold text</p>
<p class="fw-normal">Normal weight text</p>
<p class="fw-light">Light weight</p>
<p class="fst-italic">Italic text</p>

```

Bold text

Normal weight text

Light weight

Italic text

Display

d-inline d-inline
d-block
d-block

```

<h2>Display</h2>
<div class="d-inline p-2 bg-primary text-white">d-inline</div>
<div class="d-inline p-2 bg-dark text-white">d-inline</div>
<div class="d-block p-2 bg-dark text-white">d-inline</div>
<div class="d-block p-2 bg-primary text-white">d-inline</div>
<div class="box bg-warning d-sm-none d-md-block">안보이나?? 안보이나??</div>
<div class="box bg-success d-md-none d-xl-block">안보이나?? 안보이나??</div>

```

fixed-top

fixed-bottom

```

<h2>Position</h2>
<div class="box fixed-top">fixed-top</div>
<div class="box fixed-bottom">fixed-bottom</div>

```

<Bootstrap>

<https://getbootstrap.com/docs/5.2/components/buttons/>

- components
 - Bootstrap의 다양한 UI 요소를 활용할 수 있음
 - Components 탭 및 검색으로 원하는 UI 요소를 찾을 수 있음
 - 기본 제공된 Components를 변환해서 활용

Buttons

- 클릭 했을 때 어떤 동작이 일어나도록 하는 요소

```

<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-link">Link</button>

```



Dropdowns

- dropdown, dropdown-menu, dropdown-item 클래스를 활용해 옵션 메뉴를 만들 수 있습니다.



```

<div class="dropdown">
  <button class="btn btn-secondary dropdown-toggle" type="button" data-bs-toggle="dropdown" aria-expanded="false">
    Dropdown button
  </button>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Action</a></li>
    <li><a class="dropdown-item" href="#">Another action</a></li>
    <li><a class="dropdown-item" href="#">Something else here</a></li>
  </ul>
</div>

```

- **Grid system** (web design)

: 요소들의 디자인과 배치에 도움을 주는 시스템

: 기본요소

- **Column** : 실제 콘텐츠를 포함하는 부분
- **Gutter** : 칼럼과 칼럼 사이의 공간 (사이 간격)
- **Container** : Column들을 담고 있는 공간

<Bootstrap grid System>

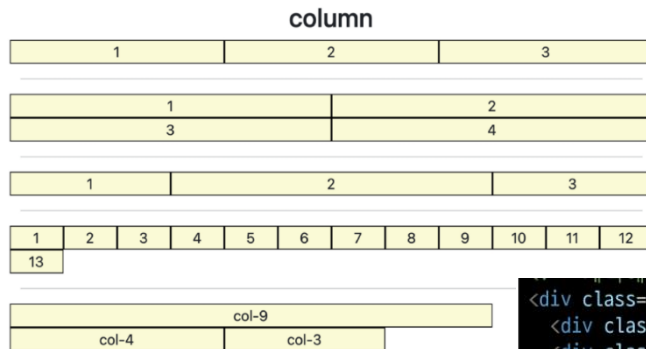
- Bootstrap Grid system은 flexbox로 제작됨
- container, rows, column으로 콘텐츠를 배치하고 정렬
- 반드시 기억해야 할 2가지 !
 1. 12개의 column
 2. 6개의 grid breakpoints

```
<div class="container">
  <div class="row">
    <div class="col"></div>
    <div class="col"></div>
    <div class="col"></div>
  </div>
</div>
```

● breakpoints

	xs ≤576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xol ≥1400px
Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-

```
<div class="container">
  <h2 class="text-center">column</h2>
  <div class="row">
    <div class="col"></div>
    <div class="col"></div>
    <div class="col"></div>
  </div>
  <hr>
  <div class="row">
    <div class="box col"></div>
    <div class="box col"></div>
    <div class="w-100"></div>
    <div class="box col"></div>
    <div class="box col"></div>
  </div>
  <hr>
</div>
```



```
<div class="row">
  <div class="box col-9">col-9</div>
  <div class="box col-4">col-4</div>
  <div class="box col-3">col-3</div>
</div>
<hr>
```

```
<h2 class="text-center">Grid breakpoints</h2>
<div class="row">
  <div class="box col-sm-8 col-md-4 col-lg-5">1</div>
  <div class="box col-8 col-sm-2 col-md-4 col-lg-2">2</div>
  <div class="box col-2 col-sm-2 col-md-4 col-lg-5">3</div>
</div>
<hr>
```

```
<h2 class="text-center">nesting</h2>
<div class="row">
  <div class="box col-6">
    <div class="row">
      <div class="box col-3">1</div>
      <div class="box col-3">2</div>
      <div class="box col-3">3</div>
      <div class="box col-3">4</div>
    </div>
  </div>
  <div class="box col-6">1</div>
  <div class="box col-6">2</div>
  <div class="box col-6">3</div>
</div>
<hr>
```

```
<h2 class="text-center">offset</h2>
<div class="row">
  <div class="box col-md-4 offset-4">.col-md-4 .offset-4</div>
  <div class="box col-md-4 offset-md-4 offset-lg-2">.col-md-4 .offset-md-4 .offset-lg-2</div>
</div>
```

