

14. 반응형 웹이란

14-1 모바일 기기와 웹 디자인

14-2 가변 그리드 레이아웃

14-3 가변 요소

14-4 미디어 쿼리

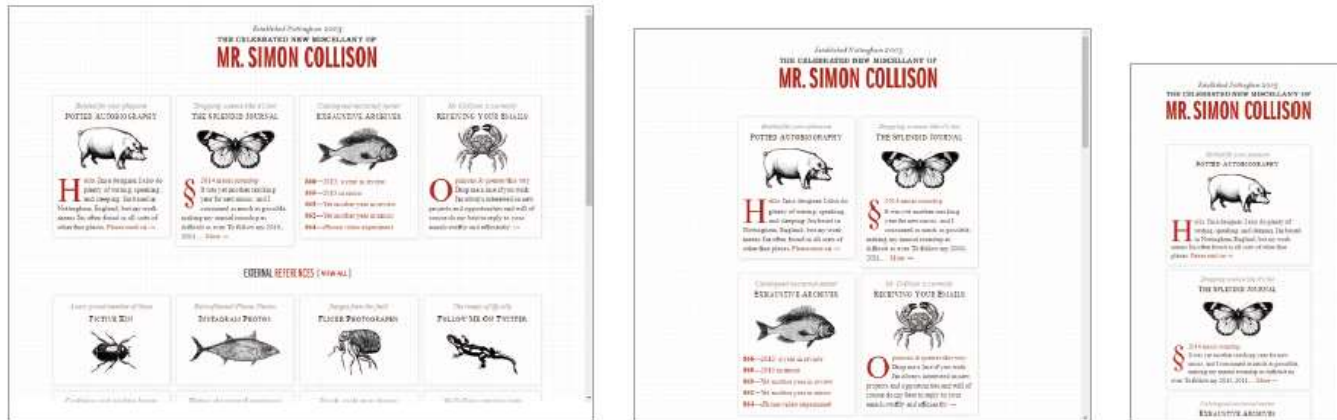
14-5 미디어 쿼리를 이용한 사이트 구성

14-6 플렉서블 박스 레이아웃



반응형 웹 디자인

- 웹 사이트의 내용을 그대로 유지하면서 다양한 화면 크기에 맞게 웹 사이트를 표시하는 방법
- 다양한 화면 크기의 모바일 기기들이 계속 쏟아져 나오는데 그 때마다 그 크기에 맞춘 사이트를 별도로 제작하는 것은 비효율적
→ 화면 크기에 '반응'해 화면 요소들을 자동으로 바꾸어 사이트를 구현하는 것이 바로 반응형 웹 디자인



반응형 웹 사이트(<http://colly.com>)

반응형 웹 디자인의 장단점

장점

- 모든 스마트 기기에서 접속 가능
- 가로 모드에 맞춘 레이아웃 변경 가능
- 사이트 유지,관리 용이

단점

반응형 웹 기술이 최신 웹 표준인 CSS3의 일부

→ 최신 모던 웹 브라우저에서만 지원됨

뷰포트(viewport)

- 뷰포트 : 실제 내용이 표시되는 영역
- PC 화면과 모바일 화면의 픽셀 표시 방법이 다르기 때문에 모바일 화면에서 의도한대로 표시되지 않음
→ 뷰포트를 지정하면 기기 화면에 맞춰 확대/축소해서 내용 표시

뷰포트 지정하기

- <head> 태그 안에서 <meta> 태그를 이용해 뷰포트 지정

기본형 `<meta name="viewport" content="속성1=값, 속성2=값2, ... ">`

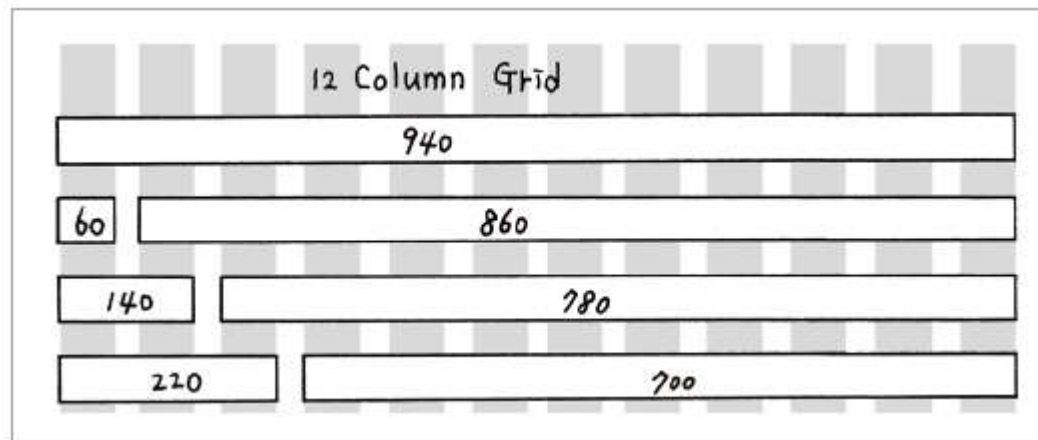
- 일반적인 사용법 : 뷰포트의 너비를 스마트폰 화면 너비에 맞추고 초기 화면 배율을 1로 지정

`<meta name="viewport" content="width=device-width, initial-scale=1">`

속성	설명	사용 가능한 값	기본 값
width	뷰포트 너비	device-width 또는 크기	브라우저 기본 값
height	뷰포트 높이	device-height 또는 크기	브라우저 기본 값
user-scalable	확대/축소 가능 여부	yes 또는 no	yes
initial-scale	초기 확대/축소 값	1~10	1
minimum-scale	최소 확대/축소 값	0~10	0.25
maximum-scale	최대 확대/축소 값	0~10	1.6

그리드 시스템(grid system)

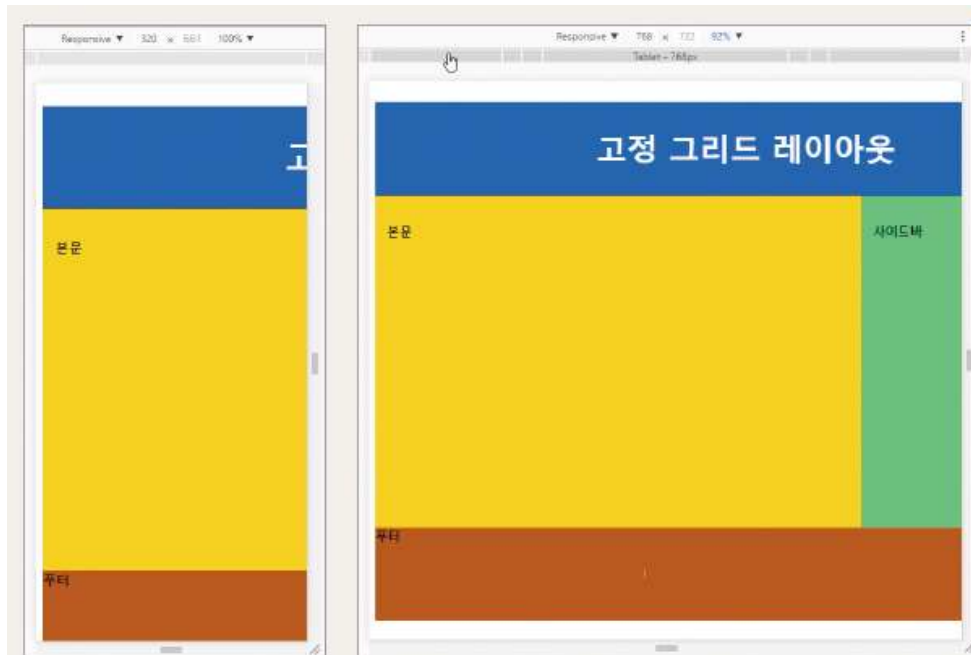
- 화면을 여러 개의 칼럼(column)으로 나누어, 필요할 때마다 칼럼들을 묶어 배치하는 방법
- 화면 너비 값에 따라 '960 그리드 시스템', '1200 그리드 시스템' 등으로 나뉨
- 칼럼 개수에 따라 '12 칼럼 그리드 시스템', '16 칼럼 그리드 시스템', '24 칼럼 그리드 시스템' 등으로 나뉨
- 주로 960 픽셀 12 칼럼의 그리드 시스템 사용
- 고정 그리드 : 화면 너비를 일정하게 고정하고 레이아웃 만듦.
- 가변 그리드 : 화면 너비를 % 같은 가변 값으로 지정.
- 가변 그리드 레이아웃을 사용할 경우, 너비 값이 줄어들면 실제 콘텐츠를 확인하기 불편하므로 가능하면 간결한 디자인을 사용하는 것이 좋음.



문서 좌우에 10px씩의 패딩이 있다고 가정한 레이아웃 →

고정 그리드 레이아웃일 경우

- 문서의 맨 바깥 부분을 #wrapper 요소로 묶고 너비를 960px로 지정
- 헤더와 본문, 사이드 바, 푸터를 배치. 이 때 너비는 px 값.
- 화면 너비가 좁아질 경우 내용의 일부가 가려질 수 있음.

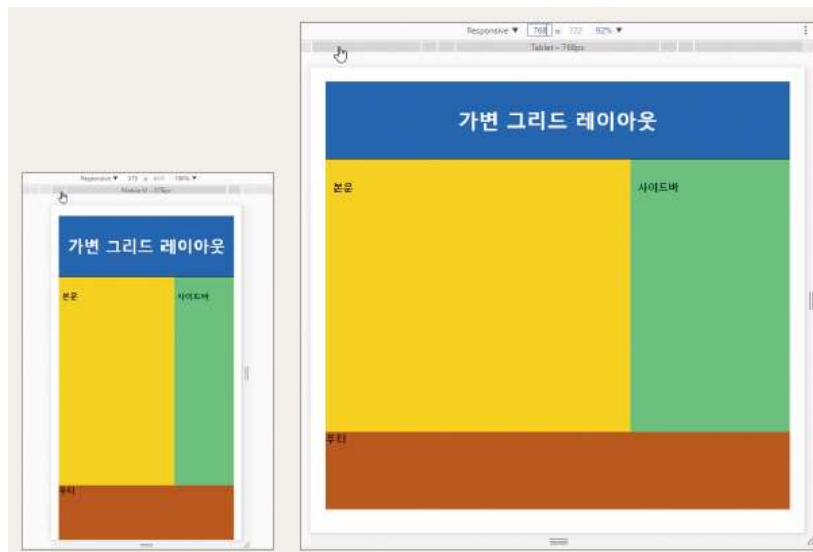


```
<style>
#wrapper {
  width:960px;
  margin:0 auto;
}
header {
  width:960px;
  height:120px;
}
.content {
  float:left;
  width:600px;
}
.right-side {
  float:right;
  width:300px;
}
footer {
  clear:both;
  width:960px;
}
</style>
```

가변 그리드 레이아웃 만들기

- ① 전체를 감싸는 요소의 너비를 %로 변환 (화면에 딱 차게 하고 싶다면 100%, 여유를 두려면 적당히)
- ② 전체를 감싸는 요소의 너비를 기준으로 각 요소의 너비를 계산

$(\text{요소의 너비} / \text{콘텐츠 전체를 감싸는 요소의 너비}) * 100$



```
<style>
#wrapper {
  width:96%;
  margin:0 auto;
}
header {
  width:100%;
  height:120px;
}
.content {
  float:left;
  width:62.5%;
  height:400px;
  padding:1.5625%;
  background-color:#ffd800;
}
.right-side {
  float:right;
  width:31.25%;
  height:400px;
  padding:1.5625%;
  background-color:#00ff90;
}
footer {
  clear:both;
  width:100%;
  height:120px;
  background-color:#c3590a;
}
```

가변 글꼴

em 단위

부모 요소 폰트의 대문자 M 너비를 1em으로 지정. 1em=16px

$$\text{글자 크기(em)} = \frac{\text{글자 크기(px)}}{16\text{px}}$$

rem 단위

- em 단위는 부모 요소가 중첩될 경우 글자 크기가 계속 달라짐
- rem은 처음부터 기본 크기를 지정하고 그것을 기준으로 글자 크기 지정

```
<style>
  .header-text{ font-size:2em; }
  .content { font-size: 1.5em; }
  .right-side { font-size: 1.5em; }
  footer { font-size: 1.5em; }
</style>
```

```
<style>
  body { font-size:16px; }
  .header-text{ font-size:2rem; }
  .fluid-text { font-size:1.5rem; }
</style>
```


가변 이미지

- 브라우저 창의 너비가 변하더라도 이미지 너비 값은 변하지 않음 → 브라우저 화면 너비를 줄일 경우 이미지 일부가 가려짐
- 가변 이미지(fluid image)로 만들면 창의 너비에 따라 이미지 너비도 조절됨

1) CSS를 이용한 방법

이미지를 감싸고 있는 부모 요소만큼만 커지거나 작아지도록 max-width 속성 값을 100%로 지정

```
<style>
  .content img {
    max-width:100%;
    height:auto;
  }
</style>
```



가변 이미지

2) 태그와 srcset 속성

화면 너비 값이나 픽셀 밀도에 따라 고해상도의 이미지
파일 지정 가능

기본형 [, <이미지2>, <이미지3>, ...]">

예)

```

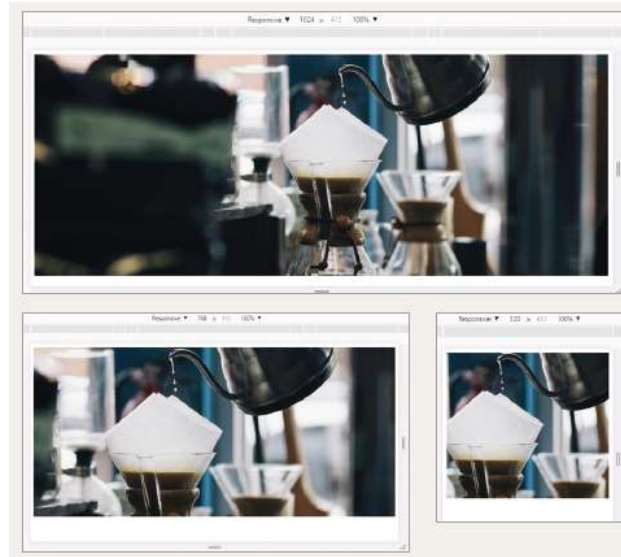
```

```
<picture>
  <source srcset="images/shop-large.jpg" media="(min-width:1024px)">
  <source srcset="images/shop-medium.jpg" media="(min-width:768px)">
  <source srcset="images/shop-small.jpg" media="(min-width:320px)">
  
```

3) <picture> 태그와 <source> 태그

화면 해상도뿐만 아니라 화면 너비에 따라 다른 이미지 파일 표시

속성	설명
srcset	이미지 파일의 경로
media	srcset에 지정한 이미지를 표시하기 위한 조건(속성 값은 14-4 미디어 쿼리 참고).
type	파일 유형
sizes	파일의 크기



가변 비디오

CSS를 사용해 max-width 속성을 100%로 지정

```
<style>  
  video { max-width: 100%; }  
</style>
```

```
<video autoplay loop src="assets/cars.mp4"> </video>
```



접속하는 장치(미디어)에 따라 특정한 CSS 스타일을 사용하도록 함.

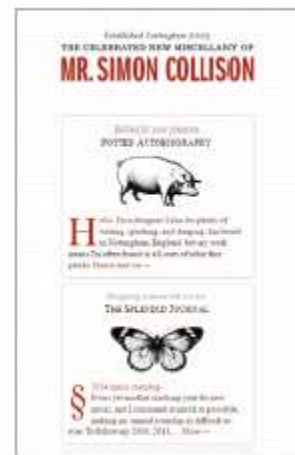
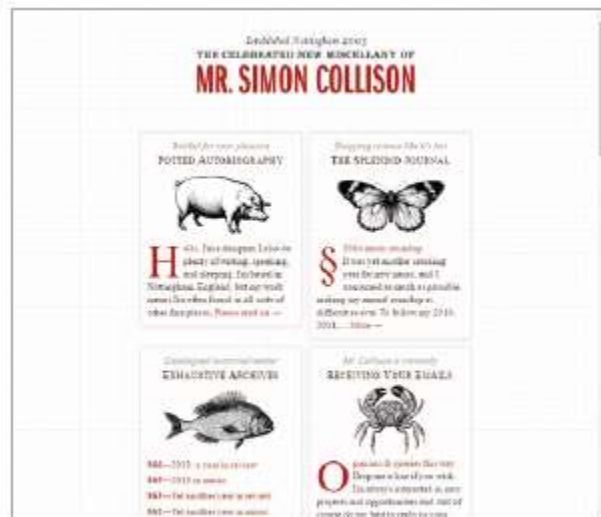
예) colly.com

브라우저 창의 너비를 조절할 때마다 화면에 표시되는 칼럼 개수가 달라짐
PC나 태블릿, 스마트폰의 웹 브라우저 화면 크기에 따라 사이트 레이아웃이 바뀜.

미디어 쿼리를 이용해 제작
된 사이트들을 모아놓은
<http://mediaqueri.es> 참고



colly.com



미디어 쿼리 구문

기본형 @media [only | not] 미디어 유형 [and 조건] * [and 조건]

연산자	설명
and	조건을 계속 추가할 수 있습니다.
,(심표)	동일한 스타일 유형을 사용할 미디어의 유형과 조건이 있다면 심표를 이용해 추가합니다.
only	미디어 쿼리를 지원하는 웹 브라우저에서만 조건을 인식하게 합니다. 이 키워드를 사용하면 미디어 쿼리를 지원하지 않는 웹 브라우저에서는 미디어 쿼리를 무시하고 실행하지 않습니다. IE에서는 미디어 쿼리를 제대로 인식하지 못하기 때문에 only 키워드를 사용하더라도 큰 의미가 없습니다.
not	not 다음에 지정하는 미디어 유형을 제외합니다. 예를 들어 'not tv' 라고 지정한다면 TV를 제외한 미디어 유형에만 적용합니다.

미디어 유형의 종류

미디어 유형	사용 가능한 미디어
all	모든 미디어 유형
print	인쇄 장치
screen	컴퓨터 스크린(스마트폰 스크린 포함)
tv	음성과 영상이 동시 출력되는 TV
aural	음성 합성 장치(주로 화면을 읽어 소리로 출력해 주는 장치)
braille	점자 표시 장치
handheld	패드(pad)처럼 손에 들고 다니는 장치
projection	프로젝터
tty	디스플레이 기능이 제한된 장치(픽셀(px) 단위를 사용할 수 없음)
embossed	점자 프린터

예) 미디어 유형이 'screen'이면서 최소 너비가 '200px'이고 최대 너비가 '360px'일 경우에 적용할 CSS를 정의하는 구문

```
@media screen and (min-width:200px) and (max-width:360px) {
    ...
}
```

미디어 쿼리 조건

웹 문서의 가로 너비와 세로 높이(뷰포트)

가로, 세로 값 설정하는 속성	설명
width, height	웹 페이지의 가로 너비, 세로 높이
min-width, min-height	최소 너비, 최소 높이
max-width, max-height	최대 너비, 최대 높이



```

<style>
body {
    background: url(images/bg0.jpg) no-repeat
    fixed;
    background-size: cover;
}
@media screen and (max-width:1024px) {
    body {
        background: url(images/bg1.jpg) no-
        repeat fixed;
        background-size: cover;
    }
}
@media screen and (max-width:768px) {
    body {
        background: url(images/bg2.jpg) no-
        repeat fixed;
        background-size: cover;
    }
}
@media screen and (max-width:320px) {
    body {
        background: url(images/bg3.jpg) no-
        repeat fixed;
        background-size: cover;
    }
}
</style>

```

미디어 쿼리 조건

단말기의 가로 너비와 세로 높이

단말기의 가로, 세로 값을 설정하는 속성	설명
device-width, device-height	단말기의 가로 너비, 세로 높이
min-device-width, min-device-height	단말기의 최소 너비, 최소 높이
max-device-width, max-device-height	단말기의 최대 너비, 최대 높이

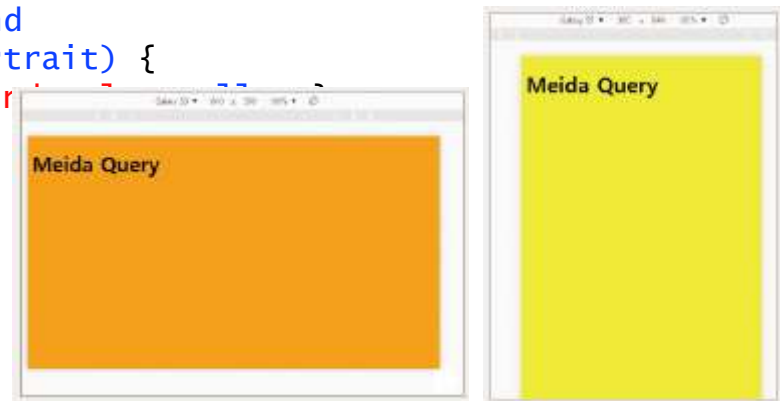
단말기 크기와 뷰포트 크기를 하나로 통일해 사용하기 위해 뷰포트를 지정할 때 width="device-width"로 놓고 사용

화면 회전

orientation 속성을 사용해서 화면 방향 체크

속성	설명
orientation: portrait	단말기 세로 방향
orientation: landscape	단말기 가로 방향

```
<style>
@media screen and
(orientation:landscape) {
    body {background-color:orange;}
}
@media screen and
(orientaion :portrait) {
    body {background-color:yellow;}
}
</style>
```



미디어 쿼리 조건

화면 비율

단말기 브라우저 화면의 너비 값을 높이 값으로 나눈 것

속성	설명
aspect-ratio	화면 비율(width 값/ height 값)
min-aspect-ratio	최소 화면 비율
max-aspect-ratio	최대 화면 비율

단말기 화면 비율

단말기의 너비 값을 높이 값으로 나눈 것

속성	설명
device-aspect-ratio	단말기 화면 비율(width 값/ height 값)
min-device-aspect-ratio	단말기 최소 화면 비율
max-device-aspect-ratio	단말기 최대 화면 비율

색상당 비트 수

단말기의 최대 색상 비트 수를 조건으로 사용

color:1 → $2^1 = 2$

color:2 → $2^2 = 4$

color:3 → $2^3 = 8$

속성 값	설명
color	비트 수
min-color	최소 비트 수
max-color	최대 비트 수

미디어 쿼리 중단점 만들기

- 중단점(breakpoint) : 서로 다른 CSS를 적용할 화면 크기
- 대부분 기기의 화면 크기 기준.
- 모든 기기를 반영할 수 없기 때문에 스마트폰과 태블릿, 데스크톱 정도로 구분
- 모바일 퍼스트(mobile first) : 모바일 기기 레이아웃을 기본으로 작성 → 태블릿 & PC 레이아웃 작성
- 미디어 쿼리 중단점은 개발자나 작업 조건에 따라 달라질 수 있다.

/ 스마트폰 세로 */*

```
@media only screen and (min-width: 320px) { ..... }
```

/ 스마트폰 가로 */*

```
@media only screen and (min-width: 480px) { ..... }
```

/ 태블릿 세로 */*

```
@media only screen and (min-width: 768px) { ..... }
```

/ 태블릿 가로 / 데스크톱 */*

```
@media only screen and (min-width: 1024px) { ..... }
```

미디어 쿼리 사용한 사이트 구성

외부 CSS 파일 연결

특정 조건에 맞을 경우, 지정한 css 파일을 가져와 적용

<link> 태그 사용하기

<head>와 </head> 사이에 삽입

기본형 `<link rel="stylesheet" media="미디어 쿼리 조건" href="css 파일 경로">`

예) 화면 너비가 768px 이하일 때 미리 만들어 놓은
태블릿용 스타일 시트 파일을 적용하려면

```
<link rel="stylesheet" media="screen and (max-width:768px)" href="css/tablet.css">
```

@import 구문 사용하기

<style>와 </style> 사이에 삽입

기본형 `@import url(css 파일 경로) 미디어 쿼리 조건`

예) 화면 너비가 321px 이상 768px 이하일 때 미리 만들어
놓은 태블릿용 스타일 시트 파일을 적용하려면

```
@import url("css/tablet.css") only screen and (min-width:321px) and (maxwidth:768px);
```

미디어 쿼리 사용한 사이트 구성

웹 문서에서 직접 정의하기

<style> 태그 안에 media 속성 사용

```
기본형 <style media="미디어 쿼리 조건">
    스타일 규칙들
</style>
```

예) 최대 너비가 320px일 때 즉 너비가 320px 이하인 경우

```
<style media="screen and (max-width:320px)">
  body {
    background-color: orange;
  }
</style>
```

<style> 태그 안에 @media 구문 사용하기

```
기본형 <style>
    @media 미디어 쿼리 조건{
        스타일 규칙들
    }
</style>
```

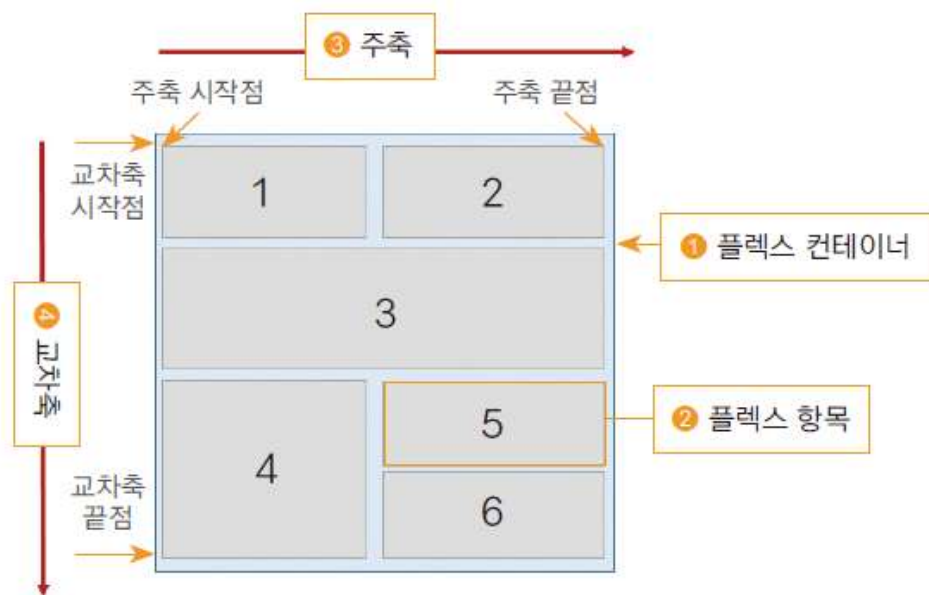
예) 화면 너비가 320px 이하일 때 배경 색을 주황색으로 바꾸는 미디어 쿼리

```
<style>
  @media screen and (max-width:320px) {
    body {
      background-color: orange;
    }
  }
</style>
```

플렉서블 박스 레이아웃

플렉서블 박스 레이아웃(flexible box layout)

- 그리드 레이아웃을 기본으로, 플렉스 박스를 원하는 위치에 배치하는 것.
- 여유 공간에 따라 너비나 높이, 위치를 자유롭게 변형할 수 있음



① 플렉스 컨테이너(flex container) - 웹 문서에 텍스트나 이미지, 표 등 웹 요소들을 플렉서블하게 사용하려면 먼저 플렉스 컨테이너로 묶어 주어야 합니다.

② 플렉스 항목(flex item) - 플렉스 컨테이너에 담기는 웹 요소입니다. 위의 그림에서 1~6번 요소들은 모두 플렉스 항목입니다.

③ 주축(main axis) - 플렉스 컨테이너 안에서 플렉스 항목을 배치하는 기본 방향입니다. 기본적으로 주축은 왼쪽에서 오른쪽으로 수평 방향으로 배치합니다. 주축에서 플렉스 항목이 배치되기 시작하는 지점을 '주축 시작점', 배치가 끝나는 지점을 '주축 끝점'이라고 합니다.

④ 교차축(cross axis) - 교차축은 주축과 교차되는 방향입니다. 기본적으로 위에서 아래로 배치합니다. 교차축에서 배치가 시작되는 지점을 '교차축 시작점', 배치가 끝나는 지점을 '교차축 끝점'이라고 합니다.

플렉서블 박스 레이아웃 기본 속성

display 속성

배치 요소들을 감싸는 부모 요소를 플렉스 컨테이너로 지정

기본형 display: flex | inline-flex

속성 값	설명
flex	플렉스 박스를 박스 레벨 요소로 정의합니다.
inline-flex	플렉스 박스를 인라인 레벨 요소로 정의합니다.

```
<style>
#container {
  display: flex;
}
</style>
```

```
<div id="container">
  <div></div>
  <div></div>
</div>
```

display 속성과 브라우저 접두사

- 최신 모던 브라우저에서는 모두 지원되고 대부분의 구식 버전에서도 지원됨
- 하지만**, 브라우저마다 플렉스 박스를 지원하는 방법이 달라 브라우저 접두사를 붙여야 함

```
.wrapper {
  display: -webkit-box; /* iOS 6 이하, 사파리 3.1 */
  display: -moz-box; /* 파이어폭스 19 이하 */
  display: -ms-flexbox; /* IE 10 */
  display: -webkit-flex; /* 웹킷 구 버전 */
  display: flex; /* 표준 스펙 */
}
```

플렉서블 박스 레이아웃 기본 속성

flex-direction 속성

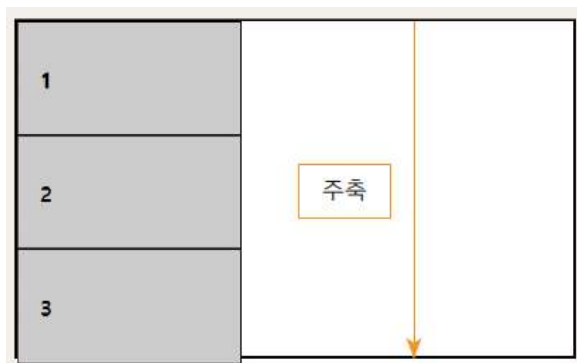
플렉스 항목 배치 방향 지정

기본형 flex-direction: row | row-inverse | column | column-inverse

속성 값	설명
row	주축을 가로로 교차축을 세로로 지정합니다. 플렉스 항목은 주축 시작점에서 끝점으로(왼쪽에서 오른쪽으로) 배치됩니다.★
row-inverse	주축을 가로로 교차축을 세로로 지정합니다. 플렉스 항목은 주축 끝점에서 시작점으로(오른쪽에서 왼쪽으로) 배치됩니다.
column	주축을 세로로 교차축을 가로로 지정합니다. 플렉스 항목은 주축 시작점에서 끝점으로(위쪽에서 아래쪽으로) 배치됩니다.
column-inverse	주축을 세로로 교차축을 가로로 지정합니다. 플렉스 항목은 주축 끝점에서 시작점으로(아래쪽에서 위쪽으로) 배치됩니다.

```
<style>
#container {
  display: flex;
  flex-direction: column;
}
#container div {
  width: 200px;
  border: 1px solid black;
  background: #ccc;
}
</style>
```

```
<div id="container">
  <div id="box1"><h2>1</h2></div>
  <div id="box2"><h2>2</h2></div>
  <div id="box3"><h2>3</h2></div>
</div>
```



플렉서블 박스 레이아웃 기본 속성

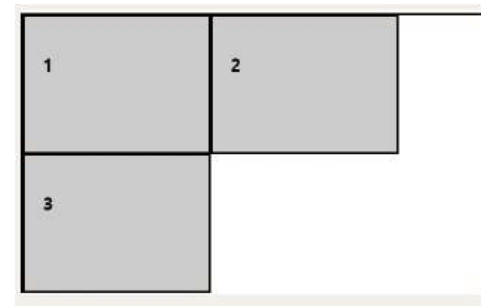
flex-wrap 속성

플렉스 항목을 한 줄 또는 여러 줄로 배치

기본형 flex-wrap: no-wrap | wrap | wrap-reverse

속성 값	설명
no-wrap	플렉스 항목들을 한 줄에 표시합니다. 기본 값입니다.★
wrap	플렉스 항목을 여러 줄에 표시합니다.
wrap-reverse	플렉스 항목을 여러 줄에 표시하되 기존 방향과 반대로 배치합니다.

```
<style>
#container {
  display: flex;
  flex-wrap: wrap;
}
</style>
```



```
<style>
#container {
  display: flex;
  flex-wrap: wrap-reverse;
}
</style>
```



플렉서블 박스 레이아웃 기본 속성

flex-flow 속성

- 플렉스 배치 방향과 여러 줄 배치를 한꺼번에 지정
- 기본 값은 flex-flow:row no-wrap

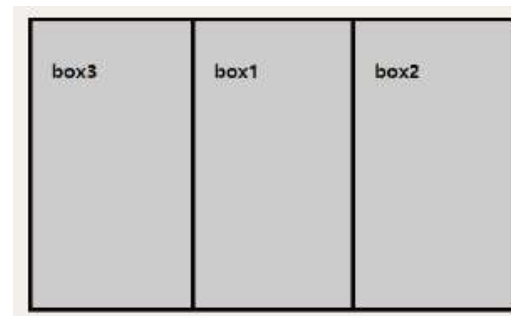
기본형 flex-flow : <플렉스 방향> <플렉스 줄 배치>

order 속성

- 플렉스 항목의 배치 순서 바꾸기
- order 값에 지정된 순서에 따라 배치됨.

```
<style>
#container {
  display:flex;
}
#box1 { order:2; }
#box2 { order:3; }
#box3 { order:1; }
</style>

<div id="container">
  <div
    id="box1"><h2>box1</h2></div>
  <div
    id="box2"><h2>box2</h2></div>
  <div
    id="box3"><h2>box3</h2></div>
</div>
```



플렉서블 박스 레이아웃 기본 속성

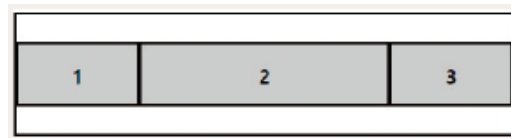
flex 속성

- 플렉스항목의 크기 조절
- flex-grow와 flex-shrink, flex-basis 속성은 별개의 속성이지만 따로 쓰지 않고 flex 속성으로 묶어 사용.

기본형 flex : [<flex-grow> <flex-shrink> <flex-basis>] | auto | initial

속성 값	설명
	플렉스 항목의 너비를 얼마나 늘일지 숫자로 지정합니다.
<flex-grow>	플렉스 항목의 너비를 얼마나 줄일지 숫자로 지정합니다.
<flex-shrink> <flex-basis>	플렉스 항목의 기본 크기를 지정합니다. width 속성처럼 너비 값을 지정할 수도 있고 0이나 auto를 지정할 수도 있습니다. 0일 경우, flex-grow와 flex-shrink의 인수 값을 함께 사용하고 auto일 경우, 플렉스 항목의 너비 값을 사용합니다.
initial	항목의 width/height 값에 의해 크기가 결정되는데 플렉스 컨테이너의 공간이 부족할 경우, 최소 크기까지 줄입니다.★
auto	항목의 width/height 값에 의해 크기가 결정되지만 플렉스 컨테이너의 공간에 따라 늘이거나 줄입니다.

```
<style>
  #box1 {
    flex:1 1 0;
  }
  #box2 {
    flex:2 2 0; /* 2배 늘이거나 2배 줄임 */
  }
</style>
```



플렉서블 박스 레이아웃

플렉스 항목 배치를 위한 속성

justify-content 속성

플렉스 항목을 주축 방향으로 배치할 때의 배치 기준

기본형 `justify-content : flex-start | flex-end | center | space-between | space-around`

속성 값	설명	미리 보기
flex-start	주축의 시작점을 기준으로 배치합니다.	
flex-end	주축의 끝점을 기준으로 배치합니다.	
center	주축의 중앙을 기준으로 배치합니다.	
space-between	첫 번째 플렉스 항목과 마지막 플렉스 항목은 시작점과 끝점에 배치한 후 중앙 항목들은 같은 간격으로 배치합니다.	
space-around	모든 플렉스 항목들을 같은 간격으로 배치합니다.	

align-items 속성, align-self 속성

교차축을 기준으로 하는 배치 방법 조절

기본형 `align-items : stretch | flex-start | flex-end | center | baseline`

기본형 `align-self : auto | stretch | flex-start | flex-end | center | baseline`

속성 값	설명	미리 보기
stretch	플렉스 항목을 확장해 교차축을 꽉 채웁니다. 기본 값입니다.★	
flex-start	교차축의 시작점을 기준으로 배치합니다.	
flex-end	교차축의 끝점을 기준으로 배치합니다.	
center	교차축의 중앙을 기준으로 배치합니다.	
baseline	시작점과 글자 기준선이 가장 먼 플렉스 항목(미리보기에서는 2번 항목의 글자 크기가 가장 크기 때문에 2번의 글자 기준선이 가장 멀리 떨어져 있음)을 시작점에 배치합니다. 그리고 그 글자의 기준선과 다른 항목의 기준선을 맞추어 배치합니다.	



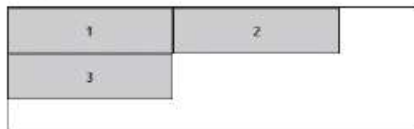
플렉스 항목 배치를 위한 속성

align-content 속성

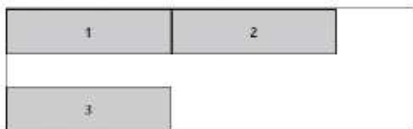
플렉스 항목이 여러 줄로 표시될 때 교차 축 기준의 배치 방법 지정

기본형 align-content : flex-start | flex-end | center | space-between | space-around

align-content : flex-start (기본 값)



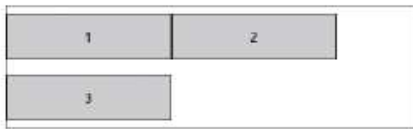
align-content : space-between



align-content : flex-end



align-content : space-around



align-content : center

