

15

## 반응형 웹 디자인



11-1

자유자재로 변하는 반응형 웹 디자인

11-2

미디어쿼리

11-3

미디어쿼리를 적용하는 방법

# 자유자재로 변하는 반응형 웹 디자인

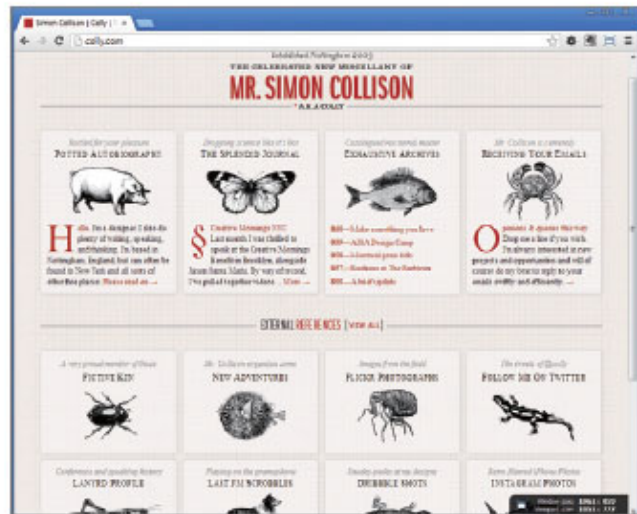
## 반응형 웹 디자인

- 스마트폰이나 태블릿, 스마트 TV처럼 다양해지는 사용자 브라우저 환경에 따라 그때마다 웹사이트를 따로 제작하는 데는 한계가 있다.
- 여러 크기의 브라우저 창에 맞게 사이트를 따로 제작하는 일은 비효율적입니다
- 원래 웹사이트 내용을 그대로 유지하면서 다양한 화면 크기에 맞게 웹사이트를 표시하도록 해 보자

### → 반응형 웹 디자인(responsive web design)

- 반응형 웹 디자인은 화면 크기에 맞게 화면 요소들을 재배치하고 각 요소의 표시 방법만 다르게 해서 사이트를 구현

# 반응형 웹 디자인



반응형 웹 디자인(<http://colly.com>)

## 뷰포트

- 스마트폰 화면에서 실제 내용이 표시되는 영역
- 320px 너비로 맞춰 모바일 사이트를 제작해도 스마트폰을 보면 아주 작게 표시됨
  - 모바일 브라우저의 기본 뷰포트 너비 980px
  - 웹 페이지를 무조건 980px 너비로 표시하려고 하기 때문

**해결)** 뷰포트 크기나 배율을 조절해야 한다



## 뷰포트

```
<meta name="viewport"
content="width=device-width">
```

- 뷰포트의 가로 너비와 단말기의 가로 너비를 맞춘다.
- 웹 페이지 배율을 조정하지 않아도 단말기 가로 너비에 맞춰 표시됨



# 뷰포트

기본형:

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

width	뷰포트의 너비를 지정합니다. 기본 값은 device-width입니다.
height	뷰포트의 높이를 지정합니다. 기본 값은 device-height입니다.
initial-scale	초기 배율을 나타내며, 기본 값은 1입니다. 1보다 작은 값을 사용하면 축소된 페이지를 표시하고, 1보다 큰 값을 사용하면 확대된 페이지를 표시합니다.
user-scalable	사용자가 페이지를 확대/축소할 수 있는지의 여부를 지정합니다. 기본 값은 'yes'인데, 'no'로 지정하면 사용자가 화면을 확대하거나 축소할 수 없습니다.
minimum-scale	사용자가 축소할 수 있는 최솟값을 지정합니다. 기본 값은 0.25이며, 가로 값을 기준으로 합니다.
maximum-scale	사용자가 확대할 수 있는 최댓값을 지정합니다. 기본 값은 5.0입니다.



## 가변 그리드 레이아웃

- 그리드 : 화면을 몇 개의 단위로 쪼개 요소를 배치하는 방식
- 화면을 단순하게 만들면서 규칙적으로 배열하기 때문에 레이아웃을 일관성 있게 유지할 수 있다는 장점이 있다.
- 기기가 바뀔 경우 가변 레이아웃 방식을 이용해 작업의 양을 줄일 수 있다.

## 가변 그리드 레이아웃

- 고정 그리드 레이아웃에서 가변 그리드 레이아웃으로 변경 시 고려해야 할 점

1. 전체를 감싸는 요소 확인하기 : 전체 감싸는 요소의 수치를 백분율로 바꾼다.
2. 하위의 각 요소의 너비 값을 백분율로 계산한다.

(요소의 너비 / 콘텐츠 전체를 감싸는 요소의 너비) \* 100

# 가변 그리드 레이아웃

```
#wrapper {  
  width:960px;  
  margin:0 auto;  
}  
header { /* 헤더 */  
  width:960px;  
  height:120px;  
  background-color:#066cfa;  
  border-bottom:1px solid black;  
}  
.header-text{  
  font-size:40px;  
  color:white;  
  text-align:center;  
  line-height:120px;  
}  
.content { /* 본문 */  
  float:left;  
  width:600px;  
  height:400px;  
  padding:15px;  
  background-color:#ffd800;  
}  
.right-side { /* 사이드 바 */  
  float:right;  
  width:300px;  
  height:400px;  
  padding:15px;  
  background-color:#00ff90;  
}  
footer { /* 푸터 */  
  clear:both;  
  height:120px;  
  background-color:#c3590a;  
}
```



```
#wrapper {  
  width:96%;  
  margin:0 auto;  
}  
header { /* 헤더 */  
  width: 100%;  
  height: 120px;  
  background-color: #066cfa;  
  border-bottom: 1px solid black;  
}  
.header-text{  
  font-size:32px;  
  color:white;  
  text-align:center;  
  line-height:120px;  
}  
.content { /* 본문 */  
  float:left;  
  width:62.5%;  
  height:400px;  
  padding:1.5625%;  
  background-color:#ffd800;  
}  
.right-side { /* 사이드 바 */  
  float:right;  
  width:31.25%;  
  height:400px;  
  padding:1.5625%;  
  background-color:#00ff90;  
}  
footer { /* 푸터 */  
  clear:both;  
  width:100%;  
  height:120px;  
  background-color:#c3590a;  
}
```



# 가변 레이아웃과 가변 요소

## 가변글꼴

- 글자 크기도 바뀌는 글꼴을 의미.
- 대표적으로 em과 rem 단위가 있다.
- em : 부모 요소에서 지정한 폰트의 대문자 M의 너비를 1em으로 지정한 것. (1em = 16px)
- rem : em에서 부모요소에 따른 자식요소의 글자 크기 변화를 없애기 위해 부모요소가 아닌 기본 크기를 기준으로 글자크기를 구하는 방식

```
.header-text{  
  font-size:2em;
```

```
.header-text{  
  font-size:2rem;
```

# 가변 레이아웃과 가변 요소

## 가변이미지

- 브라우저 크기가 작아질 경우 이미지 일부가 가려지는 현상을 방지하기 위함.
- 방법은 css를 이용하는 방법과 <img> 태그의 srcset 속성을 사용하는 방법이 있다.

### 1. css 이용 :

- max-width의 속성 값을 퍼센트로 지정한다
- 화면이 작아져도 그림이 깨지는 것을 방지할 수 있다.

```
.content img {  
    max-width:100%;  
    height:auto;  
}
```

# 가변 레이아웃과 가변 요소

## 가변이미지

2. <img> 태그와 srcset 속성 :

- 화면이 작아질 때 고화질 해상도 그림의 다운로드 문제는 여전히 존재
- 텍스트를 포함한 그림일 경우 그림이 잘 안보일 수 있음
- srcset 속성을 이용하면 화면 너비 값이나 픽셀 밀도에 따라 고 해상도의 이미지 파일을 지정할 수 있음
- 형식 : [, <이미지2>...]">
- srcset에 들어갈 이미지는 이미지 경로 뿐만이 아니라 너비와 함께 값도 입력이 가능하다.

```

```

# 가변 레이아웃과 가변 요소

## 가변이미지

3. <picture> 태그와 <source> 태그 속성 :

- HTML5.1에서 <picture>가 표준으로 잡히면서 <source>태그를 함께 사용해 해상도 뿐만 아니라 화면 너비에 따라 다른 이미지 파일을 표시할 수 있음

속성	설명
srcset	이미지 파일의 경로
media	srcset에 지정한 이미지를 표시하기 위한 조건
type	파일 유형
sizes	파일의 크기

# 가변 레이아웃과 가변 요소

## 가변비디오

- 화면의 크기가 달라질 때마다 비디오 화면의 너비가 조절될 수 있도록 css를 사용해 max-width 속성의 값을 퍼센트로 지정한다.

```
video {  
    max-width: 100%;  
}
```





# 미디어쿼리

## 미디어쿼리(Media Query)

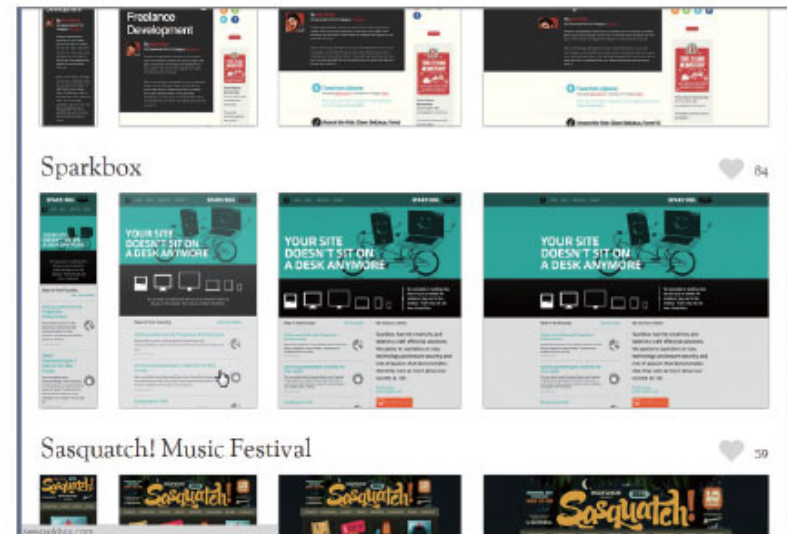
- 사이트에 접속하는 장치에 따라 특정한 CSS 스타일을 사용하도록 한다.
- 뷰포트 너비 같은 여러 장치들의 조건에 맞춰 특정한 CSS 스타일을 지정한다.

➤ W3C의 미디어 쿼리 모듈 스펙(<http://www.w3.org/TR/css3-mediaqueries/>)에서의 정의

HTML4와 CSS2는 미디어 유형마다 각각의 스타일시트를 지원하고 있습니다. 예를 들어, 화면(screen)에 표시할 때는 산세리프(sans-serif) 글꼴을 사용하고, 인쇄(print)할 때는 세리프(serif) 글꼴을 사용하도록 2개의 미디어 유형이 정의된 문서를 만들 수 있습니다. 미디어 쿼리(media query)는 스타일시트에 정밀하게 라벨을 붙여서 미디어 유형을 더욱 넓게 확장한 것입니다.

# http://mediaqueri.es 사이트

- 미디어 쿼리를 이용해 제작된 사이트들을 모아놓은 곳
- 사이트 섬네일 이미지를 클릭하면 해당 사이트로 이동
- 미디어 쿼리를 이용해 웹사이트를 제작하면, PC나 태블릿, 스마트폰의 웹 브라우저 화면 크기에 따라 사이트 레이아웃이 바뀐다.



미디어 쿼리를 사용한 사이트 갤러리(<http://mediaqueri.es>)

## 미디어쿼리 구문

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

- 미디어 쿼리 구문은 대·소문자를 구별하지 않는다.
- 기본으로 미디어 유형이 지정되어야 하고, 필요할 경우 AND 연산자로 조건을 적용한다.

```
@media screen and (max-width:1024px) {  
  body {  
    background: url(images/bg1.jpg) no-repeat fixed;  
    background-size: cover;  
  }  
}
```

## 미디어쿼리 구문

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

- 미디어 쿼리에서 사용할 수 있는 연산자는 아래와 같다.

연산자	
and	조건을 계속 추가할 때 사이에 넣어서 씀
,[선표]	동일한 스타일 유형을 사용할 미디어의 유형과 조건이 있다면 선표를 이용해 추가
only	미디어 쿼리를 지원하는 웹 브라우저에서만 조건을 인식하게 한다. 즉 미디어 쿼리가 지원되지 않는 웹 브라우저는 미디어 쿼리가 실행되지 않는다.
not	not 다음에 지정하는 미디어 유형을 제외한다.

# 미디어쿼리 구문

## 미디어 유형의 종류

미디어유형	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(px 단위 사용 불가)
embossed	점자 프린터

# 미디어쿼리에서 사용하는 조건

실제 웹 문서 내용이 화면에 보여지는 영역

가로 너비와 세로 높이 : 뷰포트의 너비와 높이

width, height	웹 페이지의 가로 너비, 세로 높이
min-width, min-height	최소 너비, 최소 높이
max-width, max-height	최대 너비, 최대 높이

예)

```
@media screen and (max-width:960px) {  
  body { background-color:green; }  
}
```

단말기의 가로 너비와 세로 높이 : 뷰포트의 너비와 높이

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

예)

```
@media all and (device-width:320px) and  
(device-height:480px) { ... }
```

## 미디어쿼리에서 사용하는 조건

**화면 회전** : 디바이스를 세로로 또는 가로로 보기

orientation: portrait	단말기의 세로 방향
orientation: landscape	단말기의 가로 방향

예)

```
@media all and (orientation:portrait) {  
    ...  
}
```

**화면 비율** : 브라우저 화면 너비 값(width)을 높이 값(height)으로 나눈 것

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

예)

```
@media all and (aspect-ratio:16/9) {  
    ...  
}
```



## 미디어쿼리에서 사용하는 조건

**단말기의 물리적 화면 비율** : 단말기 너비 값(device-width)을 높이로 나눈 것

device-aspect-ratio	단말기 화면 비율(단말기 너비/단말기 높이)
min-device-aspect-ratio	단말기 최소 화면 비율
max-device-aspect-ratio	단말기 최대 화면 비율

예)

```
@media all and (aspect-ratio:16/9) {  
    ...  
}
```

**색상당 비트 수** : 단말기에서 사용하는 최대 색상 비트 수

color	비트 수
min-color	최소 비트 수
max-color	최대 비트 수

예)

```
@media all and (color) { ... }  
@media all and (color:0) { ... }
```

## 미디어쿼리 적용하기 - css 파일 링크

각 조건별로 스타일시트 파일을 따로 저장한 후, <link> 태그나 @import 문을 사용해서 CSS 파일 연결

기본형:

```
<link href="css 파일 경로" rel="stylesheet" type="text/css" media="조건">
```

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

## 미디어쿼리 적용하기

예) 너비가 321px 이상 768px 이하일 때 태블릿 PC용 tablet.css를 사용하도록

```
<link href="css/tablet.css" rel="stylesheet" type="text/css" media="screen  
and (min-width:321px) and (max-width:768px)">
```

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

## 미디어쿼리 적용하기 – 직접 정의하기

<style> 태그 안에서 media 속성을 사용하여 조건과 그에 맞는 스타일 정의

기본형:

```
<style media="미디어쿼리 조건">  
    스타일 규칙들  
</style>
```

예)

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

너비가 320px 이하일 때  
배경색을 orange로

## 미디어쿼리 적용하기 – 직접 정의하기

기본형:

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

예)

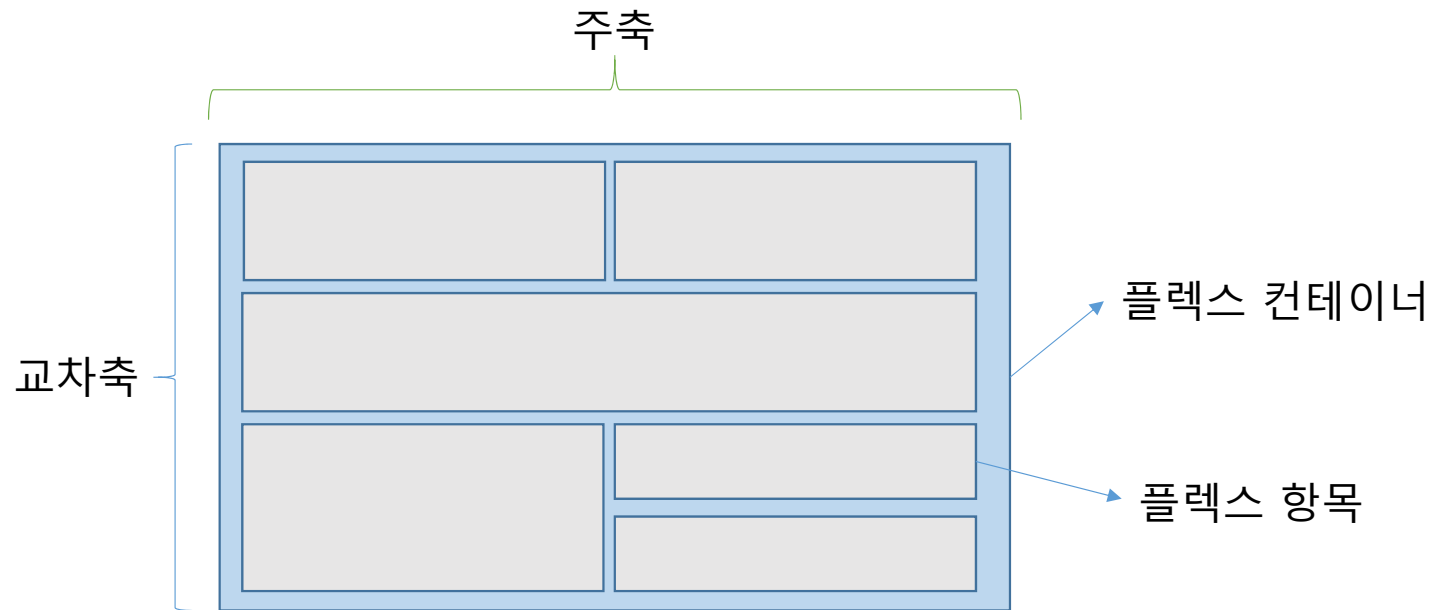
```
<style>  
  @media screen and (max-width:320px) {  
    body {background-color:orange}  
  }  
</style>
```

너비가 320px 이하일 때  
배경색을 orange로

# **플렉서블 박스 레이아웃**

## 플렉서블 박스 레이아웃

- 플렉서블 박스 레이아웃 : 그리드 레이아웃을 기본으로 해 플렉스 박스를 원하는 위치에 배치하는 것.
- 여유공간에 따라 너비나 높이, 위치를 자유롭게 변형할 수 있다.



# 플렉서블 박스 레이아웃 기본 속성

display 속성 – 플렉스 컨테이너 저장하기

**display : flex | inline-flex**

- 플렉서블 박스 레이아웃을 만들려면 먼저 웹 콘텐츠를 플렉스 컨테이너로 묶어주어야 한다.
- 특정 요소가 플렉스 컨테이너로 동작하려면 display 속성을 이용해 플렉스 박스 형태를 지정해야 한다.

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



# 플렉서블 박스 레이아웃 기본 속성

## display 속성과 브라우저 접두사

- 웬만한 구 브라우저에서도 지원은 되지만 지원하는 형식이 달라 접두사를 붙여야 한다.

```
.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 방향 지정하기

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

- 플렉스 항목의 주 축을 가로(row)로 할지 세로(column)로 할지 지정한다.
- 따로 지정하지 않으면 기본값인 row로 인식한다.

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

# 플렉서블 박스 레이아웃 기본 속성

flex-wrap 속성 – 플렉스 항목을 한 줄 또는 여러 줄로 배치하기

**flex-wrap : no-wrap | wrap | wrap-reverse**

- 플렉스를 여러 줄에 걸쳐 표시할 때 쓰는 속성.

| 속성 값         | 설명                               |
|--------------|----------------------------------|
| no-wrap      | 플렉스 항목들을 한 줄에 표시                 |
| wrap         | 플렉스 항목을 여러 줄에 표시                 |
| wrap-reverse | 플렉스 항목을 여러 줄에 표시하되 기존 방향과 반대로 배치 |

## 플렉서블 박스 레이아웃 기본 속성

flex-flow 속성 – 플렉스 방향과 여러 줄의 배치를 한꺼번에 지정하기

**flex-flow** : <플렉스 방향> <플렉스 줄 배치>

- 플렉스를 배치할 때 기본이 되는 방향과 여러 줄 배치 여부를 결정하는 속성.

flex-flow : row; /\* 왼쪽에서 오른쪽으로, 한 줄에 표시 \*/

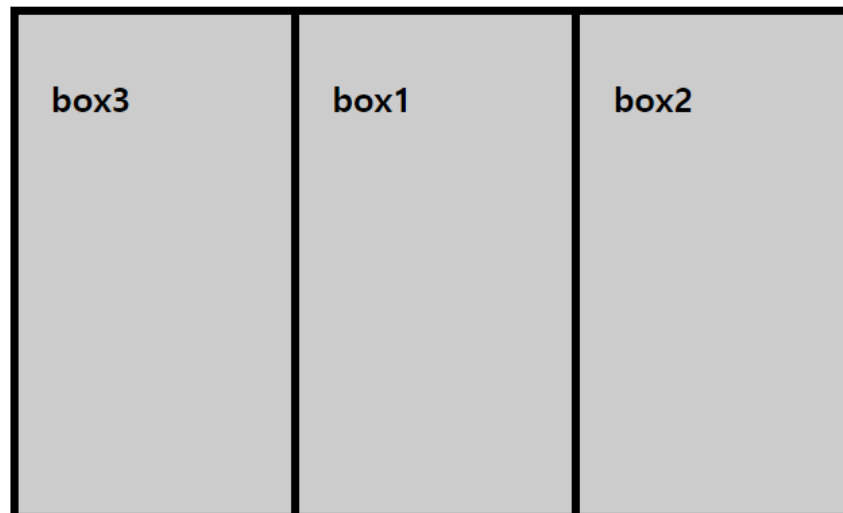
flex-flow : column wrap; /\* 위에서 아래로 여러줄에 표시 \*/

# 플렉서블 박스 레이아웃 기본 속성

order 속성 - 플렉스 항목의 배치 순서 바꾸기

**order : 0 | <숫자>**

- 플렉스의 배치 순서를 바꿀 때 사용.
- 0 : 소스에 입력한 순서대로 배치
- <숫자> : 숫자 순서에 따라 배치



```
#box1 {  
    order:2;  
}  
#box2 {  
    order:3;  
}  
#box3 {  
    order:1;  
}
```

# 플렉서블 박스 레이아웃 기본 속성

flex 속성 – 플렉스 항목 크기 조절하기

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

- 플렉스 항목의 너비를 조절할 때 쓰는 속성

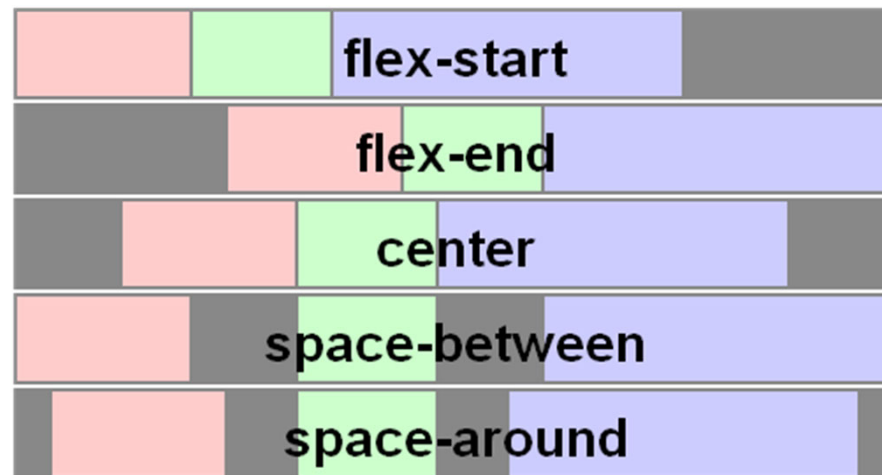
| 속성 값        | 설명   |
|-------------|--|
| flex-grow   | 플렉스 항목의 너비를 얼마나 늘릴지 배수로 지정한다   |
| flex-shrink | 플렉스 항목의 전체 너비를 기준으로 숫자만큼 전체를 나눠 부피를 축소한다.  |
| flex-basis  | 플렉스 항목의 기본 크기를 지정한다. width 속성처럼 지정할 수도 있고, 0이나 auto를 지정할 수도 있다. 0일 경우 flex-grow와 flex-shrink의 인수값을 함께 사용하고 auto일 경우 플렉스 항목의 너비값을 사용한다. |
| initial     | 항목의 width/height 값에 의해 크기가 결정되는데 플렉스 컨테이너의 공간이 부족할 경우, 최소 크기까지 줄인다.  |
| auto        | 항목의 width/height 값에 의해 크기가 결정되지만 플렉스 컨테이너의 공간에 따라 늘이거나 줄인다.  |

## 플렉스 항목 배치를 위한 속성들

justify-content 속성 – 주축 기준의 배치 방법 지정하기

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

- 플렉스 항목의 배치 기준을 지정할 때 쓰는 속성

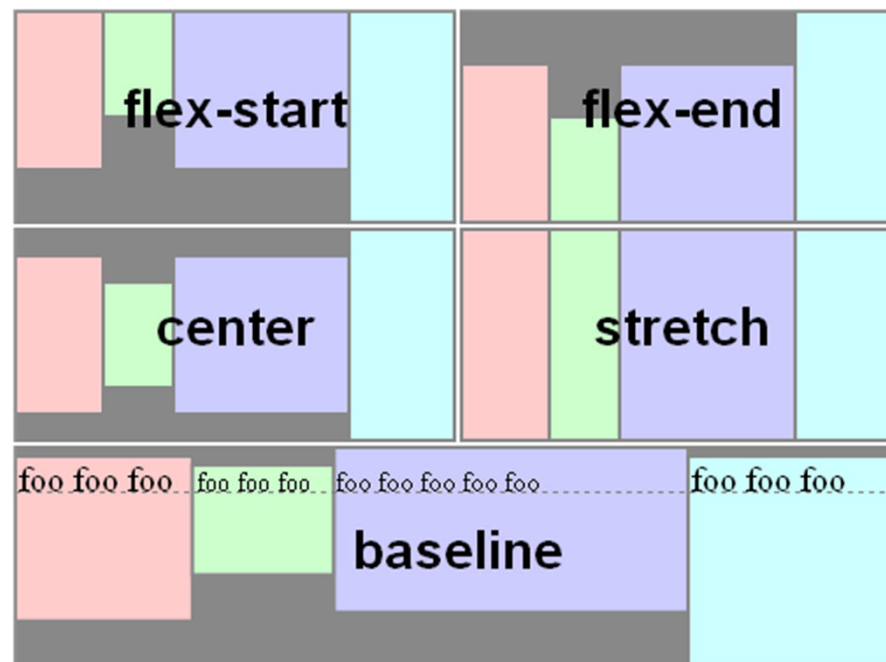


# 플렉스 항목 배치를 위한 속성들

align-item 속성 – 교차축 기준의 배치 방법 지정하기

**align-item : stretch | flex-start | flex-end | center | baseline**

- 교차축을 기준으로 배치를 조절할 때 쓰는 속성





## 플렉스 항목 배치를 위한 속성들

**align-self** 속성 – 교차축 기준의 배치 방법 지정하기

**align-item : auto | stretch | flex-start | flex-end | center | baseline**

- 교차축을 기준으로 플렉스 항목을 개별적으로 배치하는 속성
- 부모 속성 값을 상속받는 auto를 제외하면 align-item과 같다.

# 플렉스 항목 배치를 위한 속성들

align-content 속성 – 여러줄일 때의 배치 방법 지정하기

**align-item : flex-start | flex-end | center | space-between | space-around**

- 항목이 여러줄에 걸쳐 표시될 때 align-content 속성을 통해 교차축 방향의 배치 방법을 지정할 수 있다.

