기초웹개발론

kakao

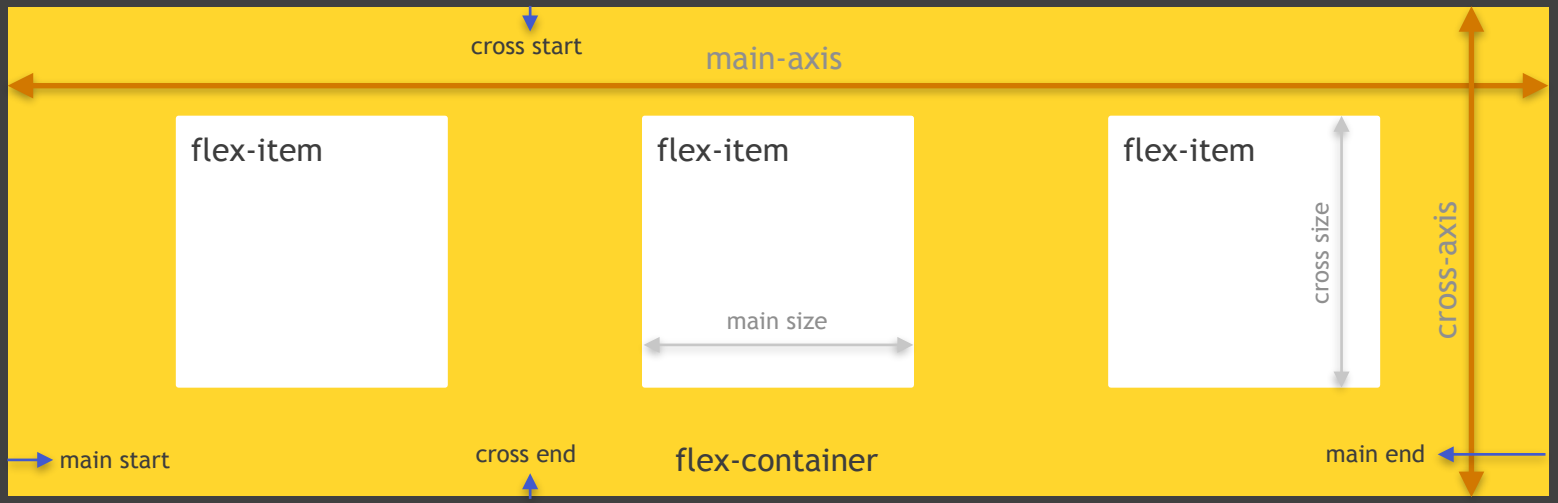
Responsive Web
Part.2

kakao

Flexbox layout

- 뷰포트나 요소의 크기가 불명확하거나 동적으로 변할 때에도 효율적으로 요소를 배치, 정렬, 분산할 수 있는 방법을 제공하는 방식
- 복잡한 계산 없이 요소의 크기와 순서를 유연하게 배치 할수 있다.
- CSS 만으로 다양한 레이아웃을 구현할수 있다.
- 브라우저 지원여부 : https://caniuse.com/#feat=flexbox

Flexbox 구성



- flexbox 레이아웃은 flex-item 으로 불리는 복수의 자식 요소와 이들을 내포 하는 flex-container 부모 요소로 구성된다.
- flex-item은 main-axis에 따라 정렬 되며 main-axis 의 방향은 flex-container 의 flex-direction속성으로 결정한다.

kakao

Flexbox 정의

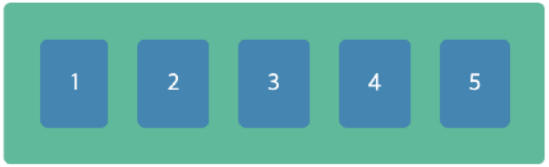
```
.flex-container {
  display: flex;
  /* or */
  display: inline-flex;
}
```

flex 속성은 부모요소에 반드시 적용해야 하는 유일한 속성이며, 적용된 요소는 flex container가 되고, flex container 의 자식 요소는 자동으로 flex item이 된다.

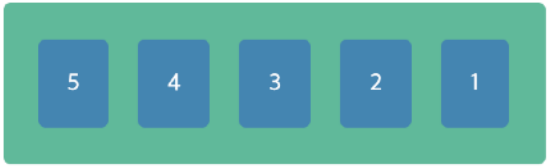
flex container 속성

- flex-direction
  - flex-container 의 주축(main-axis)을 설정함
  - row I row-reverse I column I column-reverse
  - default : row

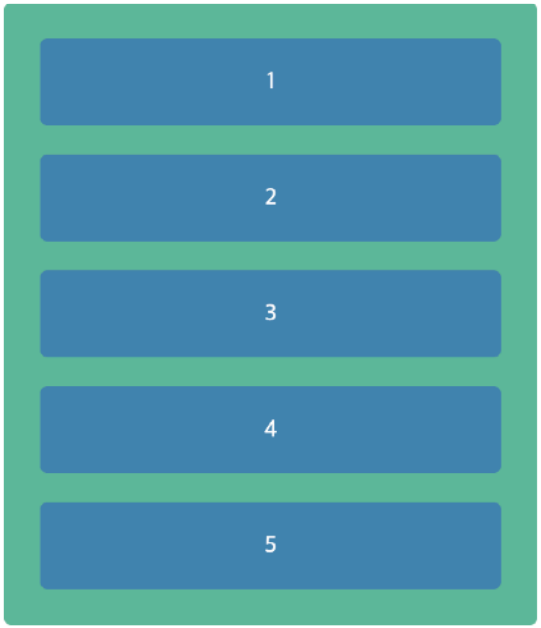
flex-direction: row



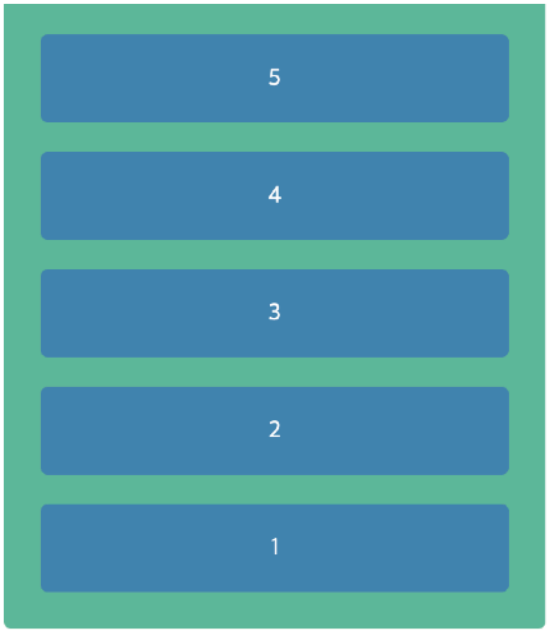
flex-direction: row-reverse



flex-direction: column



flex-direction: column-reverse

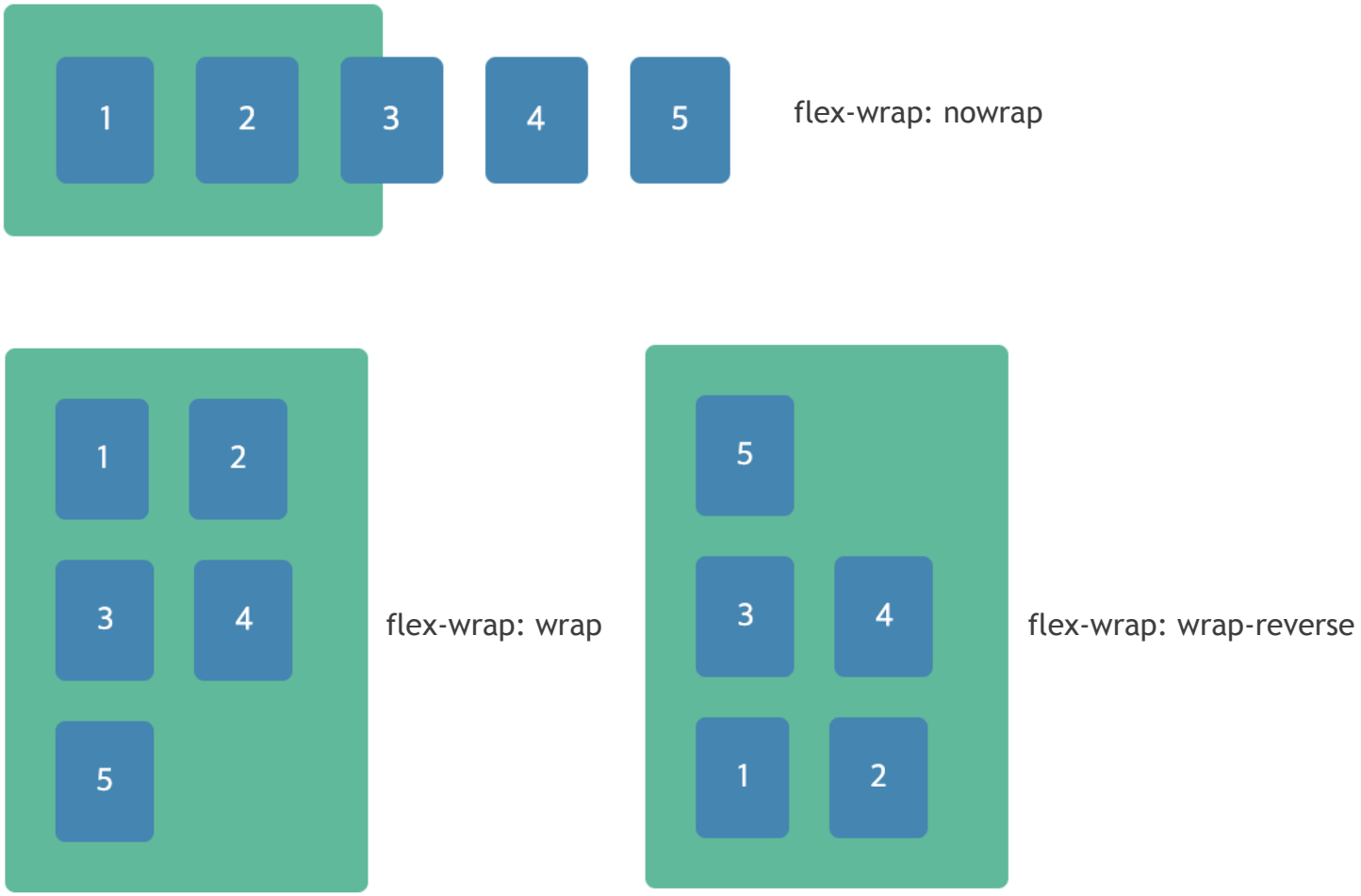


kakao

flex container 속성

- flex-wrap
  - flex-container 의 복수 flex-item을 1행 또는 복수 행으로 배치한다.
  - flex-container 의 width 보다 flex-item 들의 width 합계가 큰 경우에 적용된다
  - nowrap I wrap I wrap-reverse
  - default : nowrap

kakao


1
2
3
4
5
flex-wrap: nowrap
1
2
3
4
5
flex-wrap: wrap
5
3
4
1
2
flex-wrap: wrap-reverse

kakao

flex container 속성

- flex-flow
  - flex-direction 과 flex-wrap 속성을 위한 shorthand
  - <flex-direction> II <flex-wrap>
  - default : row nowrap

```
.flex-container {
  flex-flow: row nowrap;
}
```

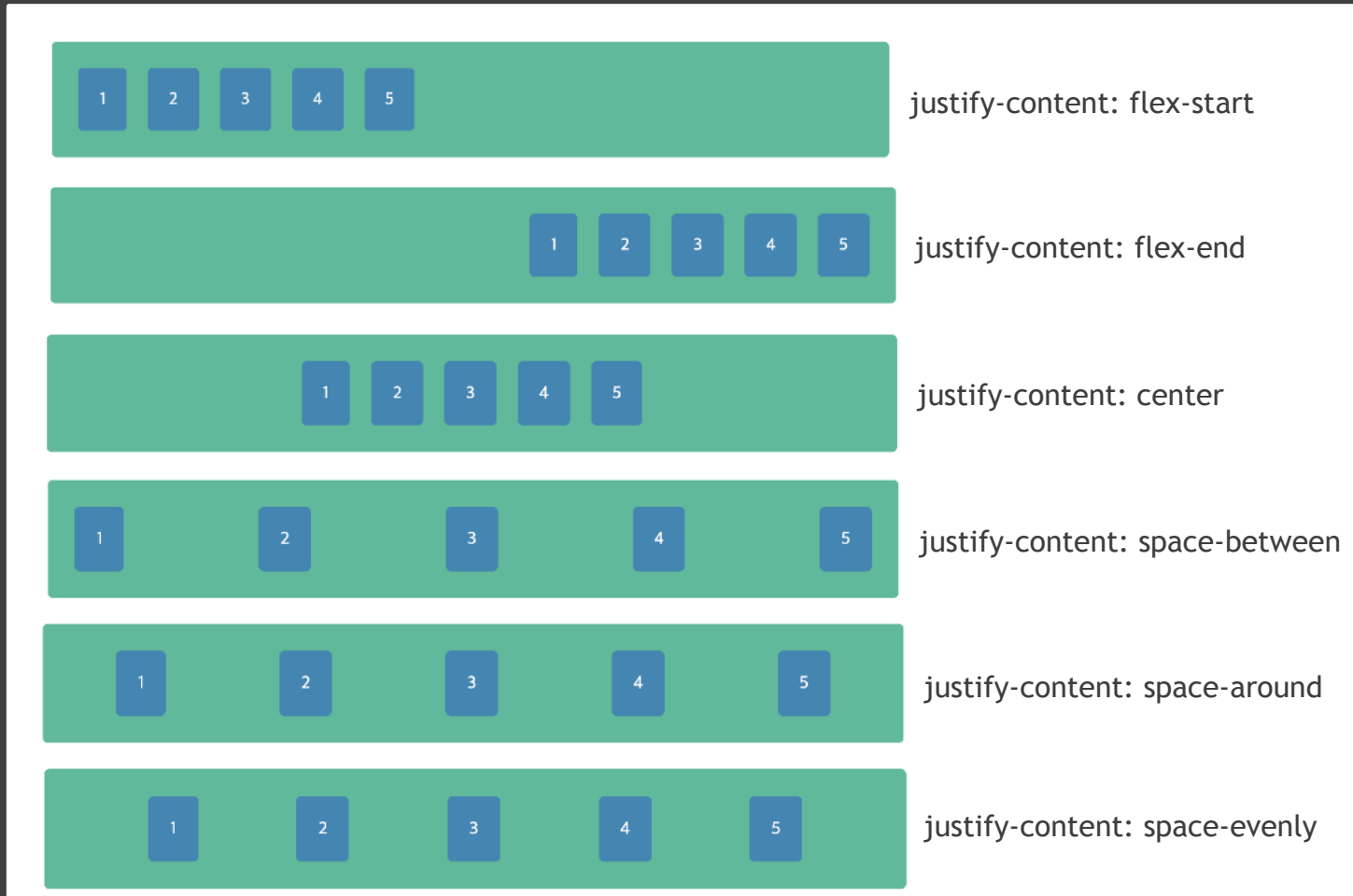
kakao

flex container 속성

- justify-content
  - main-axis 를 기준으로 flex-item 을 수평 정렬한다.
  - flex-start I flex-end I center I space-between I space-around I space-evenly
  - default : flex-start

```
.flex-container {
  justify-content: flex-start;
}
```

kakao


1 2 3 4 5
justify-content: flex-start
1 2 3 4 5
justify-content: flex-end
1 2 3 4 5
justify-content: center
1 2 3 4 5
justify-content: space-between
1 2 3 4 5
justify-content: space-around
1 2 3 4 5
justify-content: space-evenly

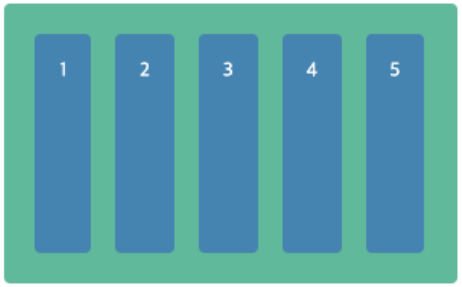
flex container 속성

- align-items
  - cross-axis 를 기준으로 flex-item 을 정렬한다.
  - stretch I flex-start I flex-end I center
  - default : stretch

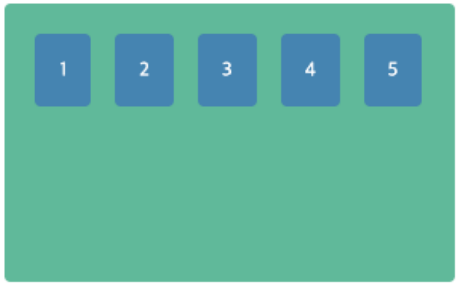
```
.flex-container {
  align-items : stretch;
}
```

kakao

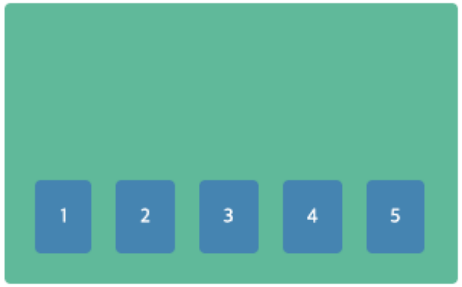
align-items: stretch



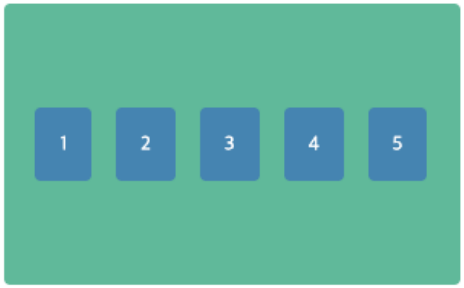
align-items: flex-start



align-items: flex-end



align-items: center



flex container 속성

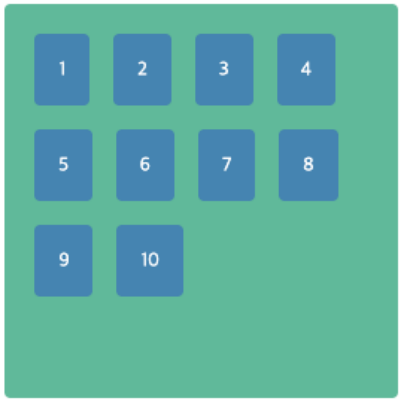
- align-content
  - cross-axis 를 기준으로 여러줄의 flex-item 을 정렬한다.
  - flex-start I flex-end I center I space-between I space-around
  - default : stretch

```
.flex-container {
  align-content : stretch;
}
```

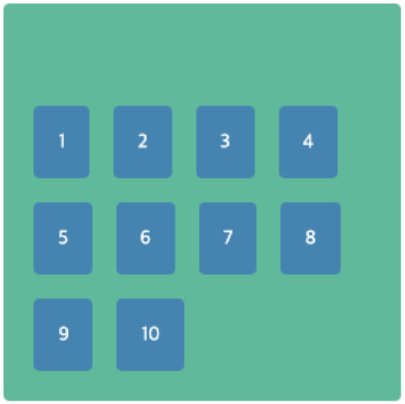
kakao

align-content: stretch

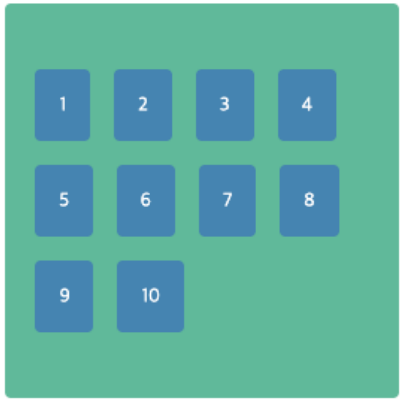
align-content: flex-start



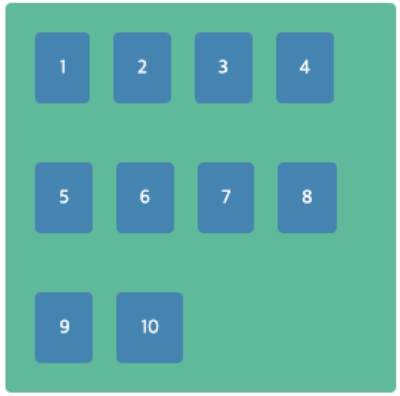
align-content: flex-end



align-content: center



align-content: space-between



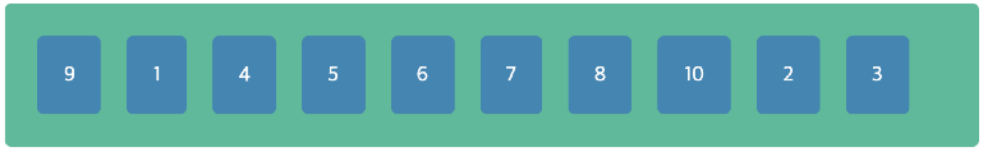
align-content: space-around

kakao

flex item 속성

- order
  - HTML 코드를 수정하지 않고 order 속성값을 지정하는 것으로 간단하게 재배치가 가능하다. 기본배치 순서는 flex-container 에 추가된 순서이다.
  - default : 0

```
.flex-item {
  order : 1;
}
```




```html
<div class="flex-container">
  <div class="flex-item n1">1</div>
  <div class="flex-item n2">2</div>
  <div class="flex-item n3">3</div>
  <div class="flex-item n4">4</div>
  <div class="flex-item n5">5</div>
  <div class="flex-item n6">6</div>
  <div class="flex-item n7">7</div>
  <div class="flex-item n8">8</div>
  <div class="flex-item n9">9</div>
  <div class="flex-item n10">10</div>
</div>
```

```css
.flex-container {
  display: flex;
  margin: 10px;
  padding: 15px;
  border-radius: 5px;
  background: #60B99A;
}

.flex-item {
  margin: 10px;
  padding: 20px;
  color: #fff;
  text-align: center;
  border-radius: 5px;
  background: #4584b1;
}

.n2 {
  order: 1
}

.n3 {
  order: 2
}

.n9 {
  order: -1
}
```

flex item 속성

- flex-grow
  - flex-item의 확장에 관련된 속성
  - 0일경우 flex-item 은 flex-container 의 크기가 변경되어도 커지지 않는다.
  - 속성값이 1이 상일경우 원래 크기와는 상관없이 flex container 를 채우도록 크기가 커진다.
  - flex-item 이 동일한 값을 가지면 모든 flex-itemn은 동일한 너비를 갖는다.
  - default: 0 (음수는 무시)

```
.flex-item {
  flex-grow : 1;
}
```

kakao

flex item 속성

- flex-shrink
  - flex-item의 축소에 관련된 속성
  - 0일경우 flex-item 은 flex-container 의 크기가 flex-item 의 크기보다 작아져도 크기가 유지된다.
  - 속성값이 1이 상일경우 flex-container 의 크기가 flex-item 보다 작아질 때 flex item의 크기가 flex-container 의 크기에 맞추어 줄어든다
  - default: 1

```
.flex-item {
  flex-shrink : 0;
}
```

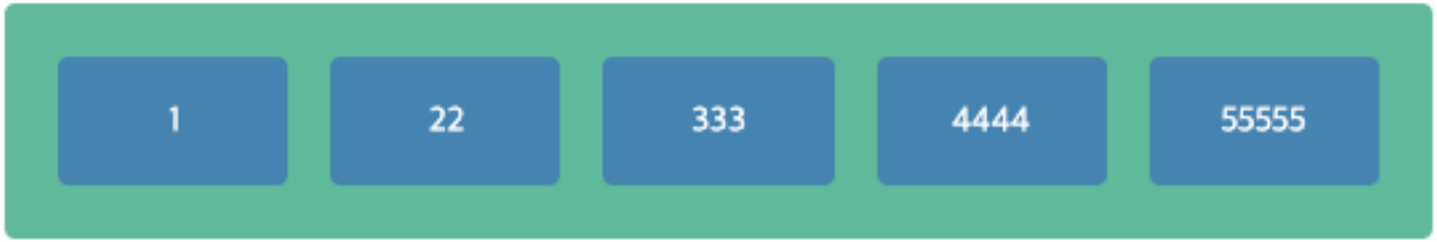
kakao

flex item 속성

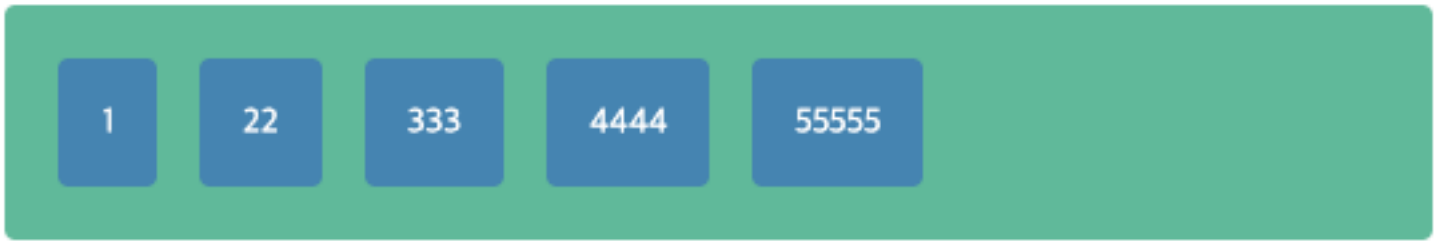
- flex-basis
  - flex-item의 기본크기를 결정하는 속성
  - 속성의 auto으로 설정하면 flex-container 기준으로 크기가 결정된다.
  - 0 로 설정하면 컨텐츠의 크기를 기준으로 크기가 결정된다.
  - 그 외 px, % 등으로 지정가능하다.
  - default: auto

```
.flex-item {
  flex-basis : 0px;
}
```

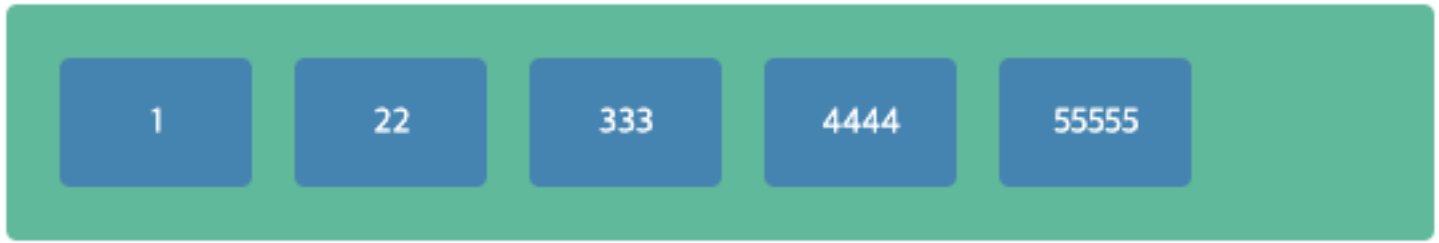
flex-basis: auto



flex-basis: 0



flex-basis: 50px



flex item 속성

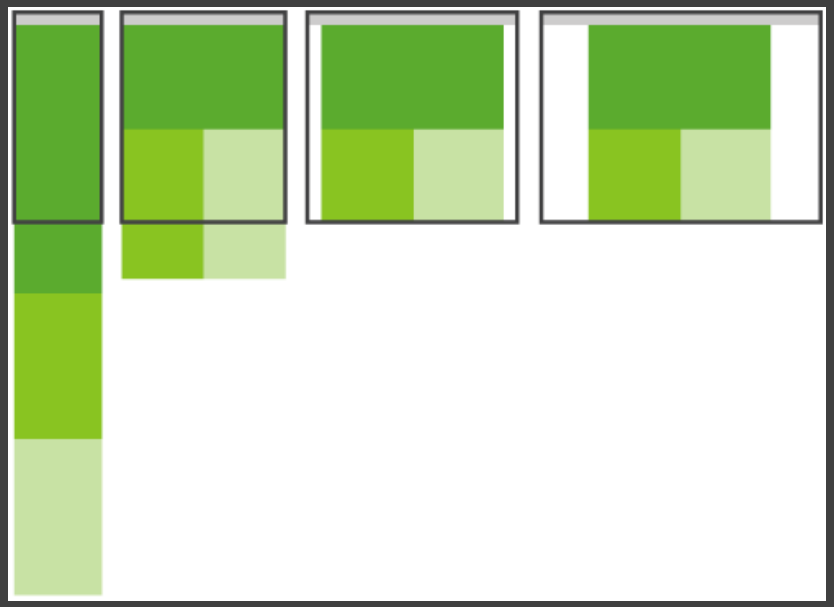
- flex
  - flex-grow, flex-shrink, flex-basis 속성을 위한 shorthand
  - 값이 1개일경우 number 이면 flex-grow, length 나 % 를 지정하면 flex-basis
  - 값이 2개일경우 첫번째 값은 number 이어야 하며 flex-grow 값이됨. 두번째 값은 number 이면 flex-shrink, length 나 % 를 지정하면 flex-basis
  - 값이 3개일경우 <number> <number> (<length>|<%>)
  - initial - flex: 0 1 auto
  - auto - flex: 1 1 auto
  - none - flex: 0 0 auto

Responsive Web
디자인패턴

kakao

레이아웃 - Mostly Fluid

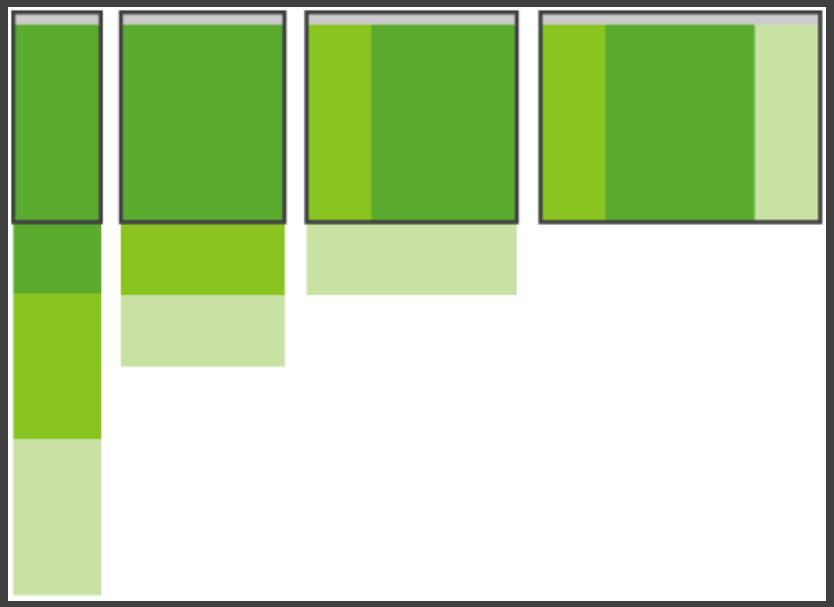
가변형 그리드를 이용해 단순하게 컨텐츠의 폭을 맞춤.



kakao

레이아웃 - Column Drop

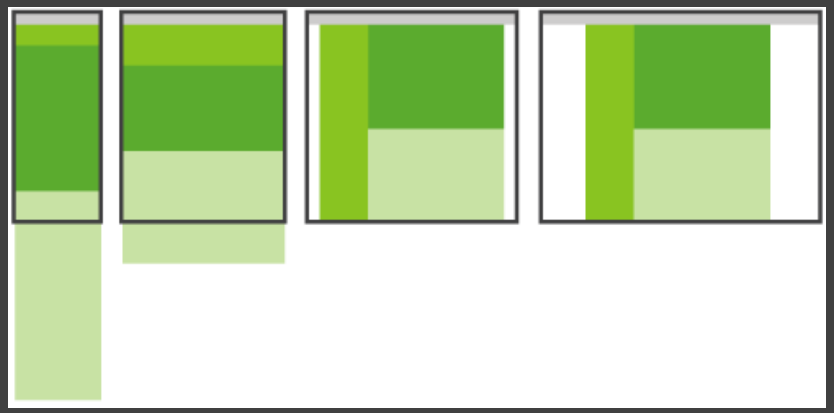
화면폭이 좁아져 더이상 컨텐츠의 정상적인 표현이 힘들때 컬럼을 하단으로 떨궈서 컨텐츠 영역을 확보



kakao

레이아웃 - Layout Shifter

스크린 크기마다 다른형태의 레이아웃을 사용, 단순히 컬럼을 내리는 패턴이 아닌 레이아웃을 바꿔 주기 때문에 단조로움에서 벗어날수는 있으나 상대적으로 관리가 힘들고 소스가 복잡해질수 있음



kakao

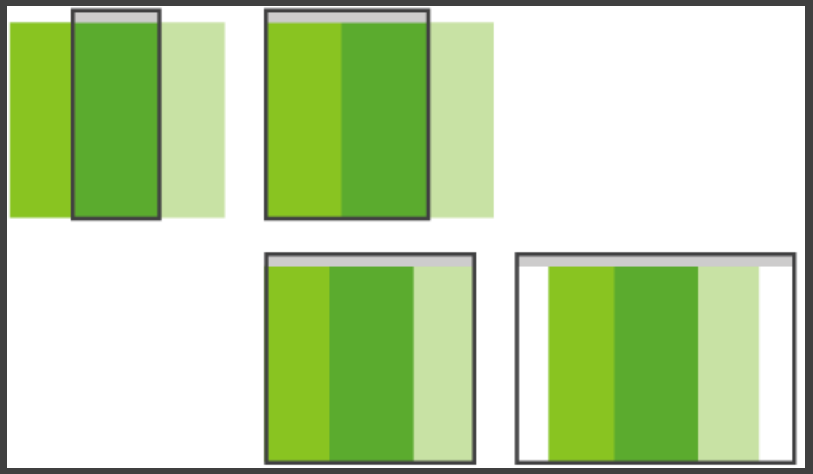
레이아웃 - Tiny tweaks

하나의 컬럼을 이용하며, 변화의 폭이 크지 않아 블로그등에서 많이 사용함.

kakao

레이아웃 - Off canvas

메뉴나 서브 컬럼을 숨겨뒀다가 사용하는 방법



kakao

가변형 이미지

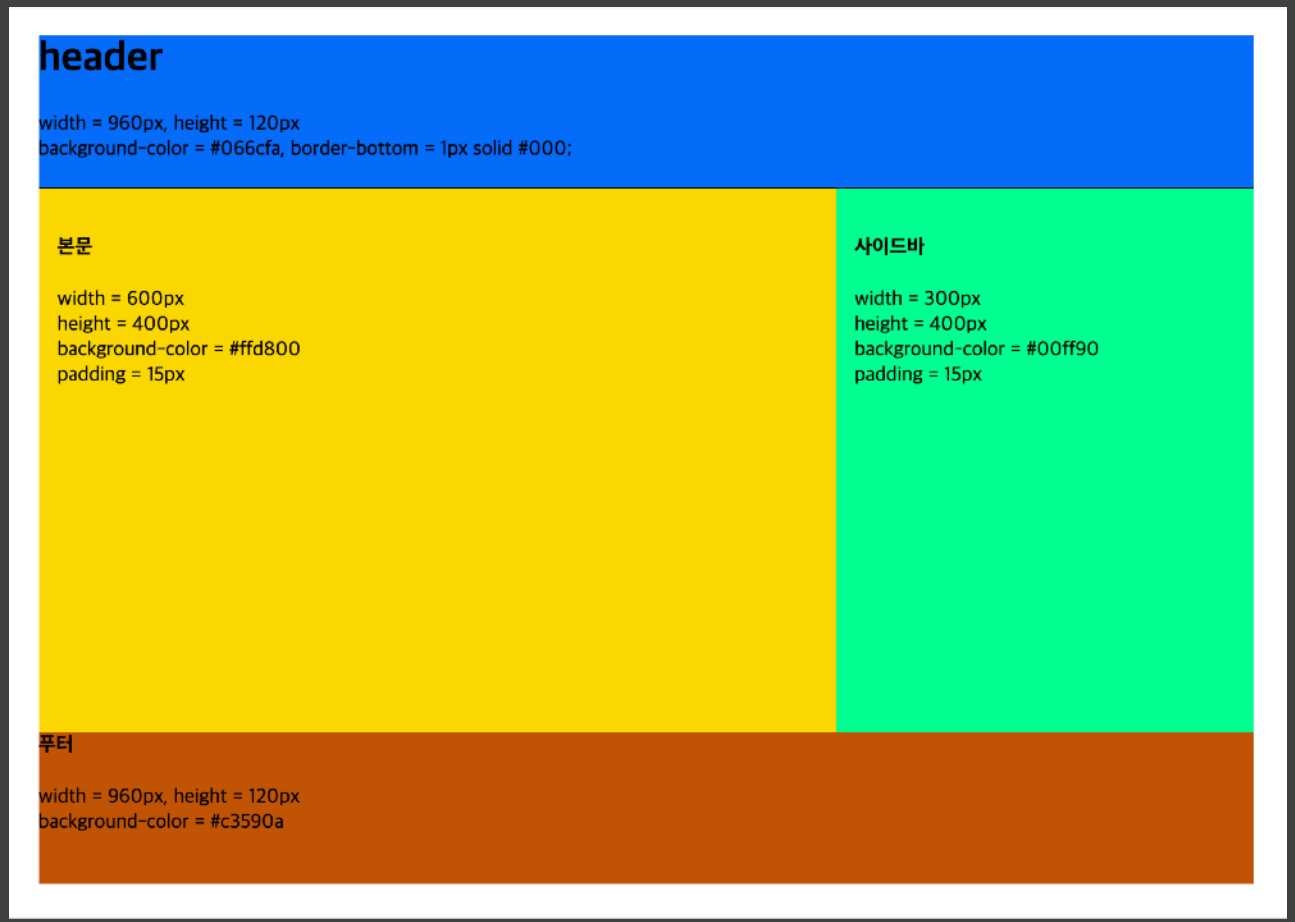
- 컬럼의 너비가 변경되어도 미디어가 컬럼을 넘어가지 않게 하는 방법
- 보통 이미지는 사이즈가 정해져 있어 뷰포트의 크기가 변경되어도 변하지 않는다.
- CSS 를 이용하는 방법
    - max-width: 100%
- <picture> <source> 태그 의 사용
- 높은 DPI 는 srcset 으로 대응.

가변형 이미지

```
<picture>
  <source media="(min-width: 800px)" srcset="head.jpg, head-2x.jpg 2x">
  <source media="(min-width: 450px)" srcset="head-small.jpg, head-small-2x.jpg 2x">
  <img src="head-fb.jpg" srcset="head-fb-2x.jpg 2x" alt="a head carved out of wood">
</picture>
```

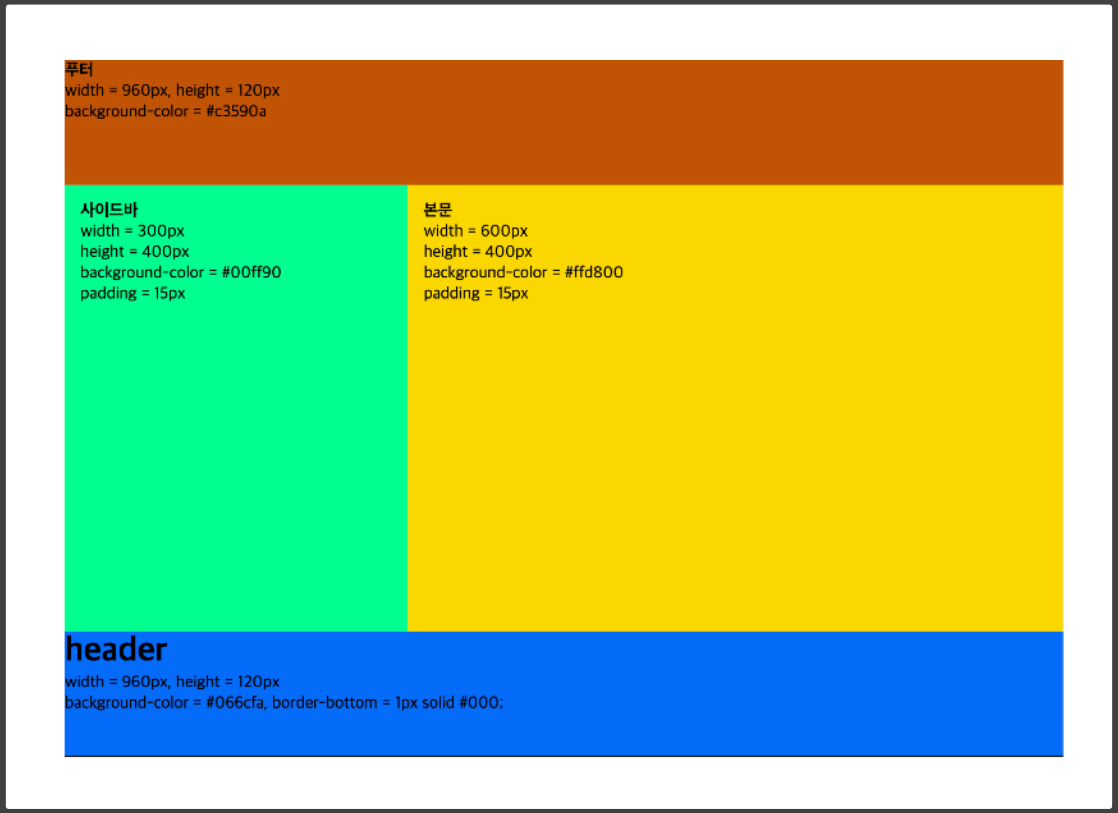
실습

상상하고 코드 보기

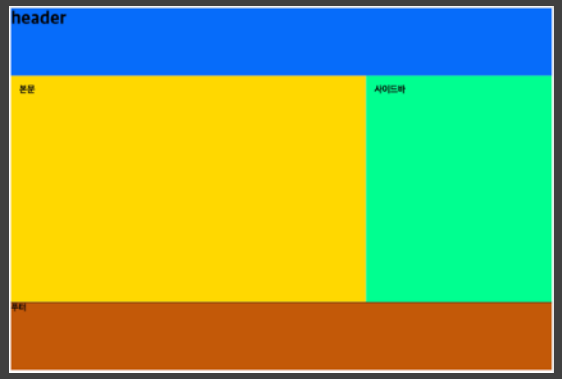

header
width = 960px, height = 120px
background-color = #066cfa, border-bottom = 1px solid #000;
본문
width = 600px
height = 400px
background-color = #ffd800
padding = 15px
사이드바
width = 300px
height = 400px
background-color = #00ff90
padding = 15px
푸터
width = 960px, height = 120px
background-color = #c3590a

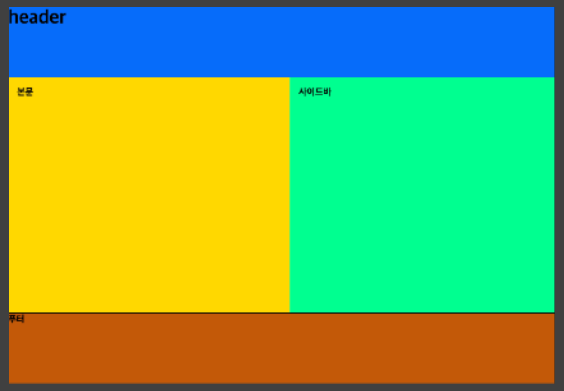
kakao

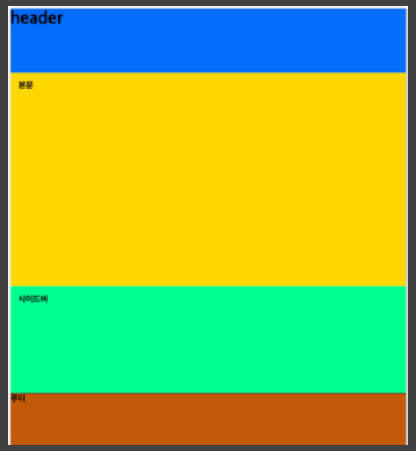
실습


푸터
width = 960px, height = 120px
background-color = #c3590a
사이드바
width = 300px
height = 400px
background-color = #00ff90
padding = 15px
본문
width = 600px
height = 400px
background-color = #ffd800
padding = 15px
header
width = 960px, height = 120px
background-color = #066cfa, border-bottom = 1px solid #000;

kakao

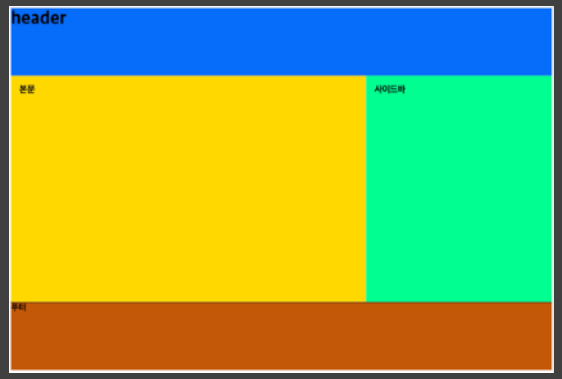


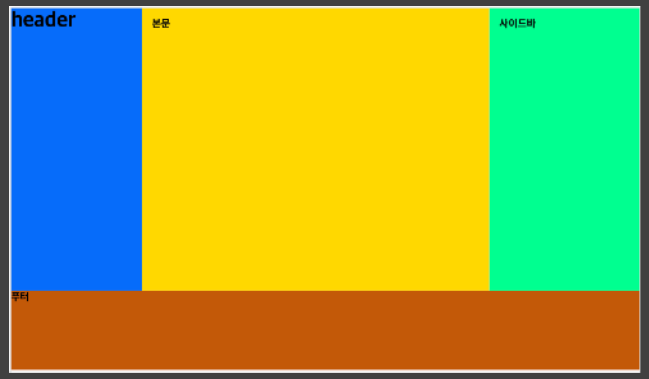


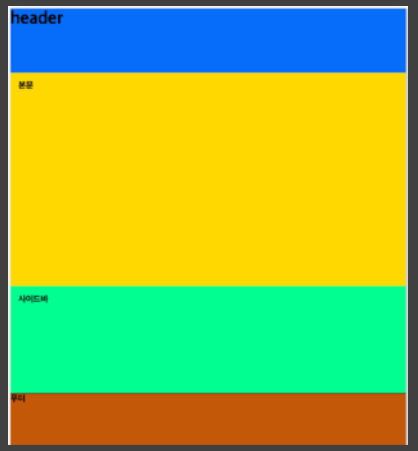


- 1025px 이상일때 기본모습
- 1025px 보다 작아지면 본문/사이드바 50:50
- 768px 보다 작아지면 한줄로 처리

kakao







- 1025px 이상일때 기본모습
- 1025px 보다 작아지면 head 가 left 로 이동 width = 200px
- 768px 보다 작아지면 한줄로 처리

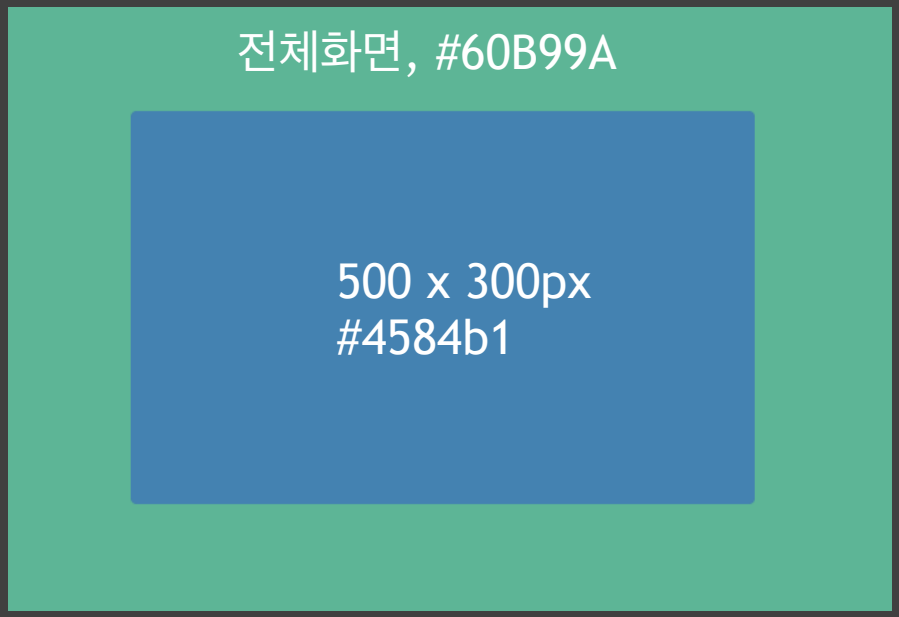
kakao

실습


header
메뉴1
메뉴2
메뉴3
메뉴4
search
본문
width = 600px
height = 400px
background-color = #ffd800
padding = 15px
사이드바
width = 300px
height = 400px
background-color = #00ff90
padding = 15px
푸터
width = 960px, height = 120px
background-color = #c3590a

실습

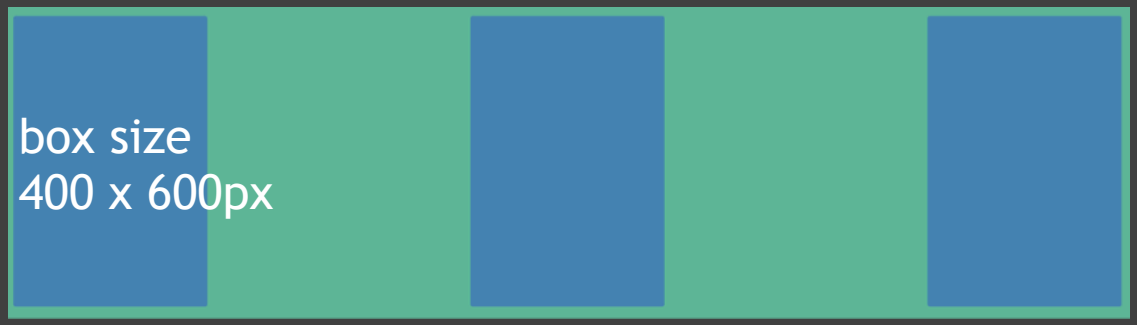
박스 가운데 정렬하기



kakao

실습

박스 3개 나열, 좌측 | 가운데 | 우측 정렬

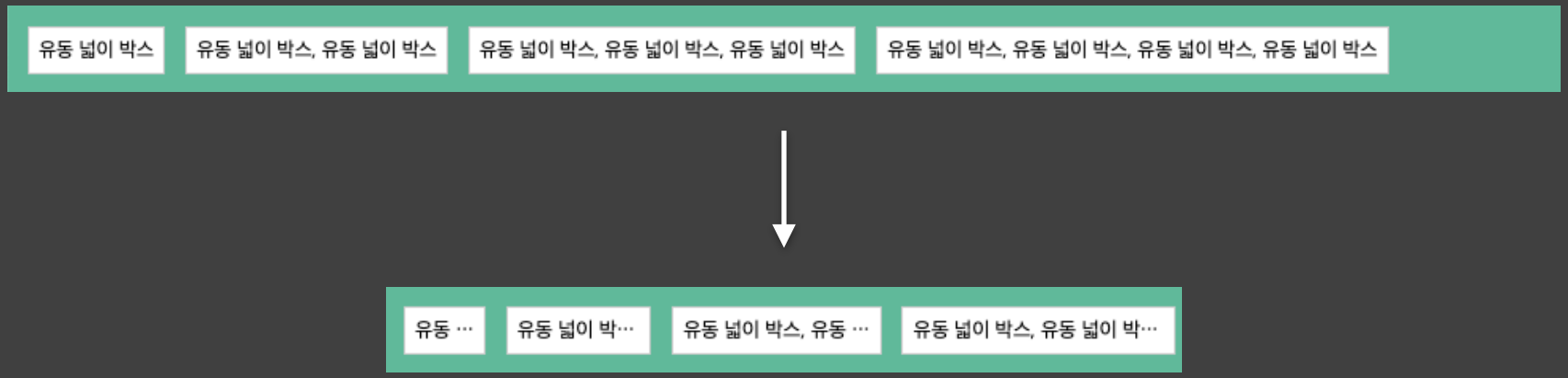


kakao

실습

화면이 줄면 같이 작아지는 박스

flex-container - height : 80

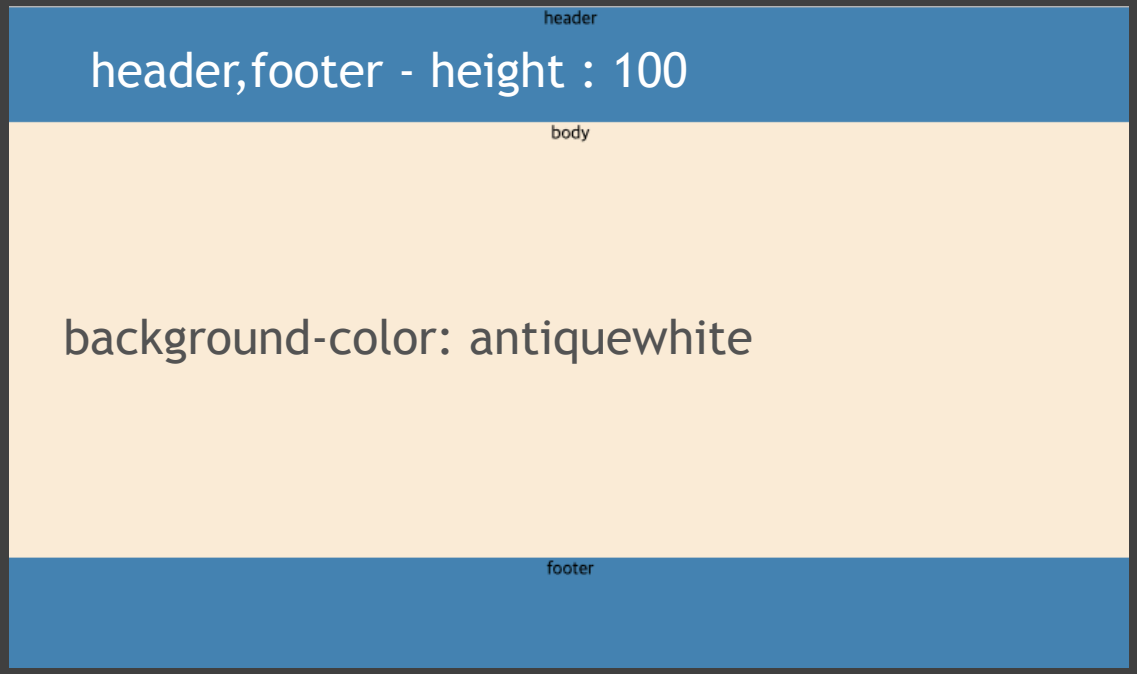


flex-item - height : 40, 말줄임처리

kakao

실습

가운데가 유동적인 Layout



kakao