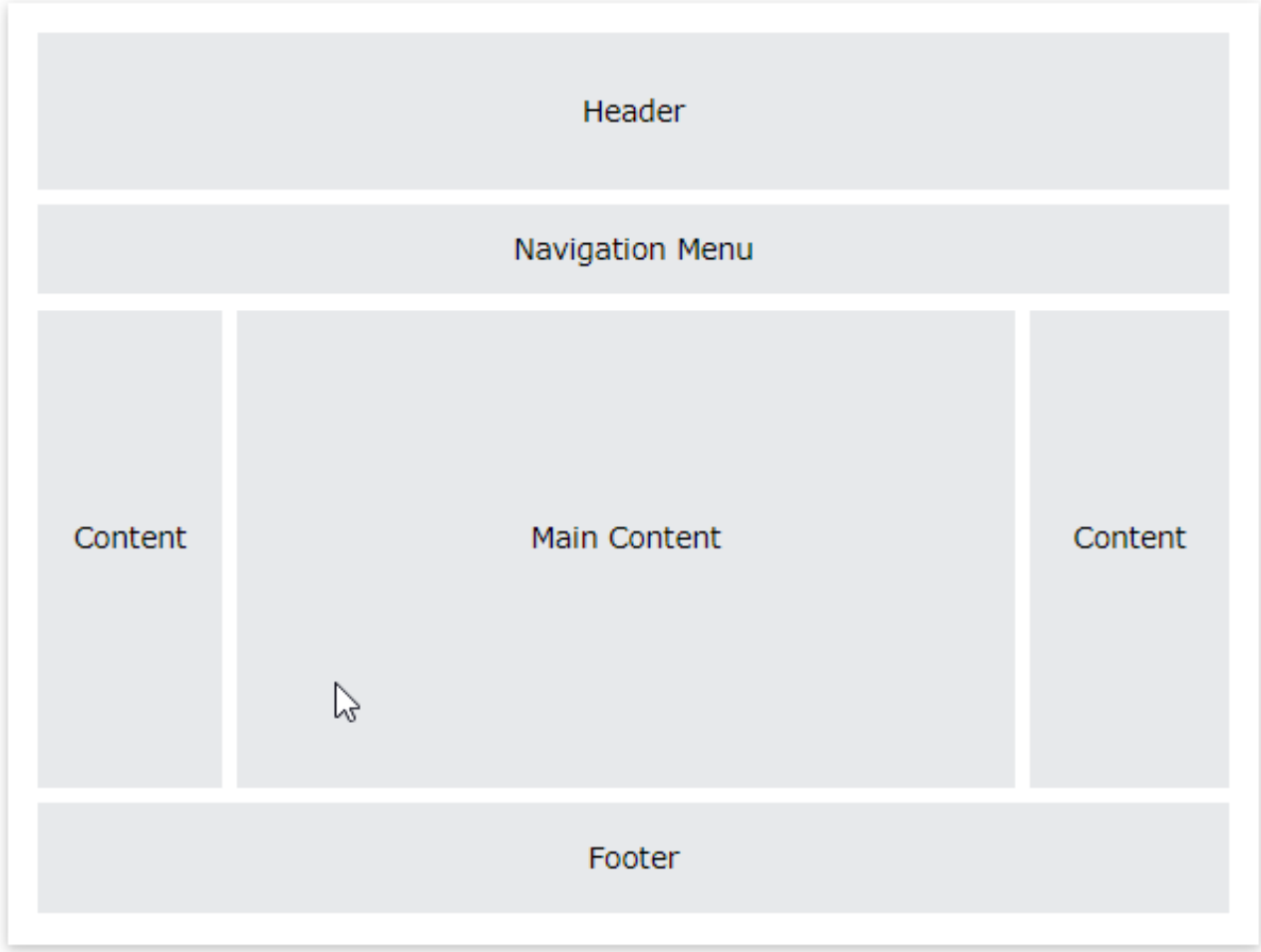
HTML – 06

레이아웃

레이아웃이란?

- 문서를 구성하는 요소들을 어디에 배치할지 결정하는 것
  - 특정 공간에 여러 구성 요소를 효과적으로 배치하는 작업
  - 웹 사이트의 외관을 결정짓는 중요 요소



레이아웃 표시(1/2)

- display 속성

속성 값	설명
block	블록 요소로 표시하며, 새 줄에서 시작해 전체 너비를 차지 함
inline	인라인 요소로 표시하며 높이 및 너비 속성에 영향을 받지 않음
inline-block	인라인 요소 수준의 블록으로 표시 (기본적으로 인라인 요소이지만 높이, 너비 속성 적용 가능)
none	요소 완전 제거 (화면에서 보이지 않음)

레이아웃 표시(2/2)

- 블록(block) 레벨 요소
  - 항상 새 줄에서 시작하고, 화면의 한 줄을 전부 차지
  - This is an inline span

    Hello World

    element inside a paragraph.

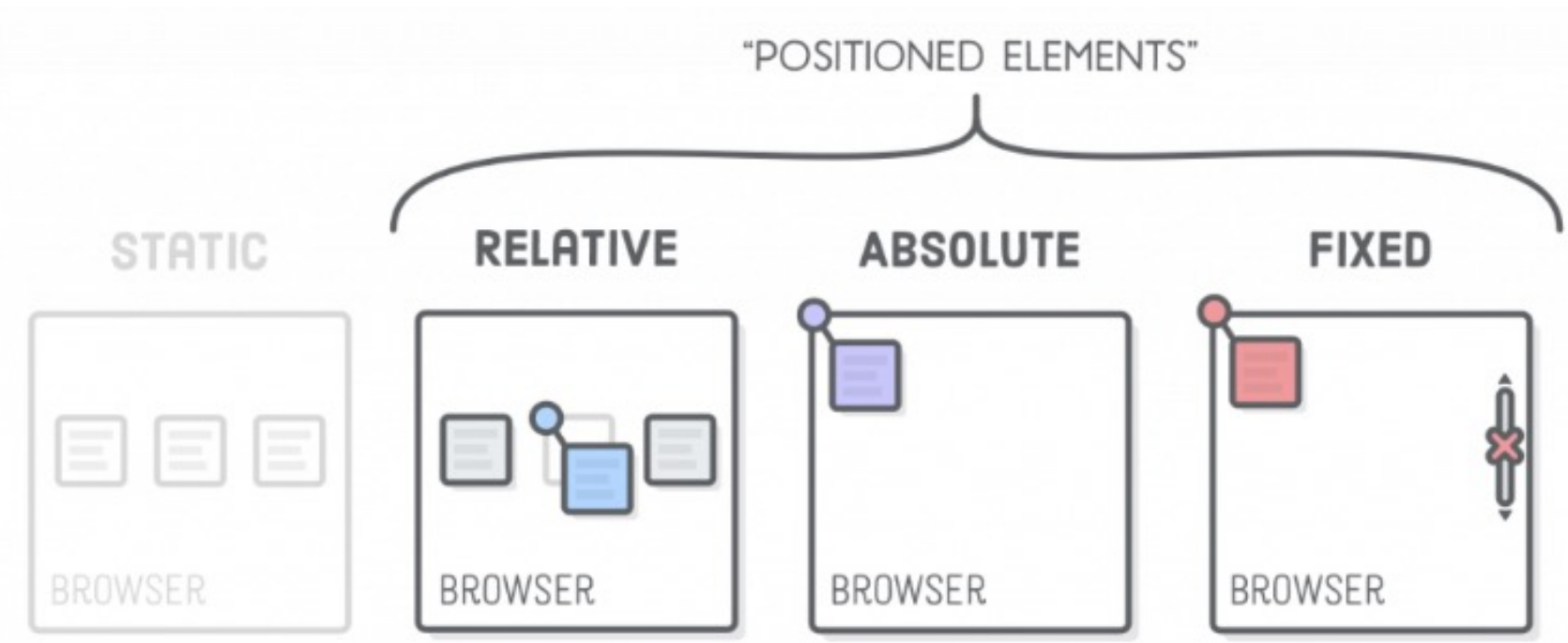
- 인라인(inline) 레벨 요소
  - 한 줄에 차례로 배치되며, 현재 줄에 필요한 너비 만을 차지
  - This is an inline span Hello World element inside a paragraph.

예제1

```
<style>
.menubar {
  text-align: center;
  background-color: yellow;
  border: solid red;
  border-left: none;
  border-right: none;
  padding: .5em;
}
.menubar li {
  display: inline;
  margin: 0 20px;
}
a {
  text-decoration: none;
}
</style>
```

레이아웃 위치(1/6)

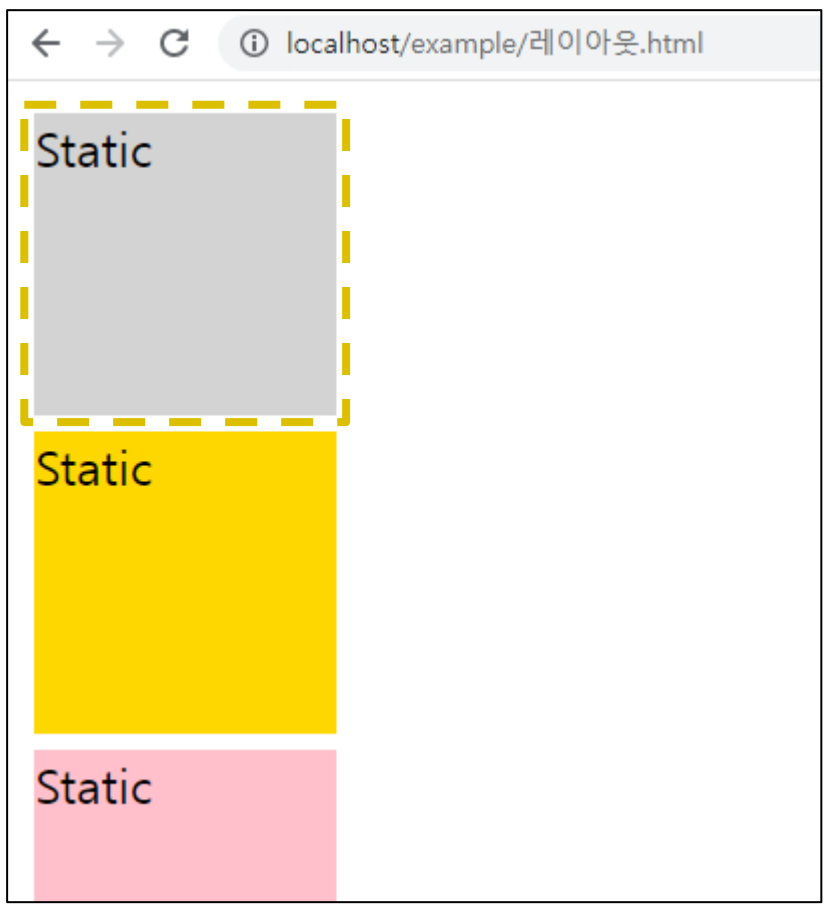
- position 속성 : 요소의 위치를 지정하는 유형 설정
  - static (정적 위치) - 정적으로 배치된다. (기본 값)
  - relative (상대 위치) - 정적인 위치를 기준으로 배치된다.
  - absolute (절대 위치) - 컨테이너를 기준으로 배치된다.
  - fixed (고정 위치) - 항상 같은 위치(컨테이너 원점)에 배치된다.



레이아웃 위치(2/6)

- 정적 위치 설정(static positioning)
  - 특별한 방식으로 배치되지 않으며 정상적인 흐름에 따라 배치된다.
  - top, bottom, left, right 속성의 영향을 받지 않는다.

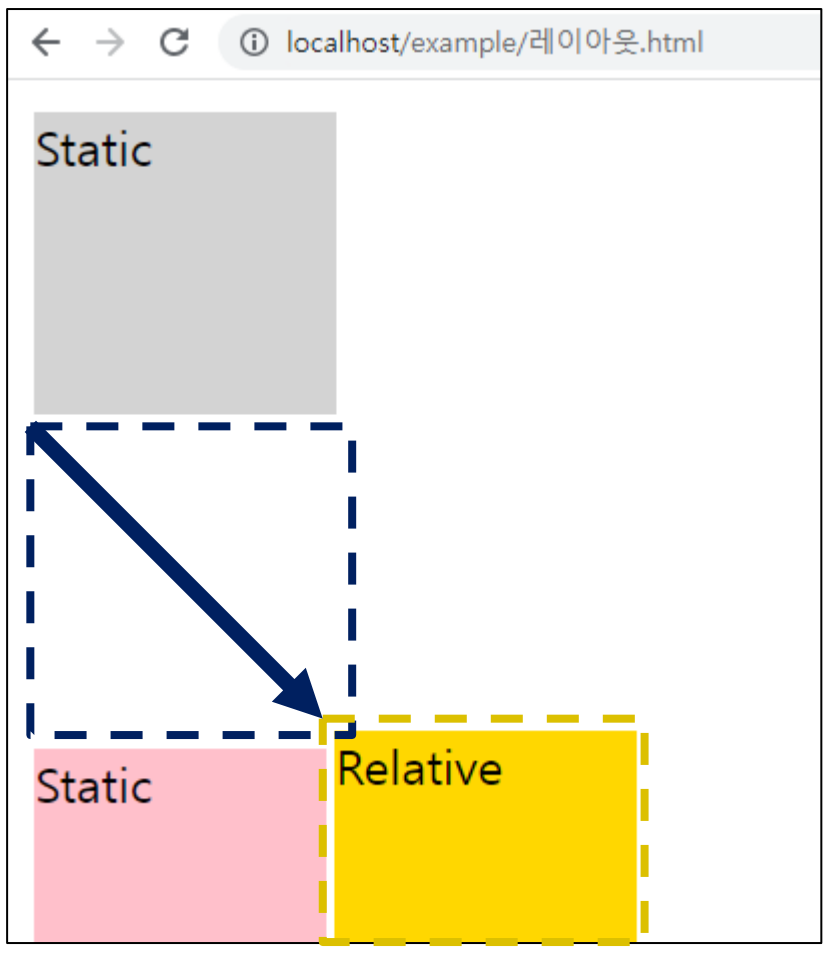
```
<style>
div {
  width: 100px;
  height: 100px;
  left: 100px;
  top: 100px;
  border: 3px solid white;
}
div#static {
  position: static;
  background: lightgray;
}
</style>
```



레이아웃 위치(3/6)

- 상대 위치 설정(relative positioning)
  - 정적인 위치를 기준으로 방향 속성을 지정한 곳에 요소를 배치한다.

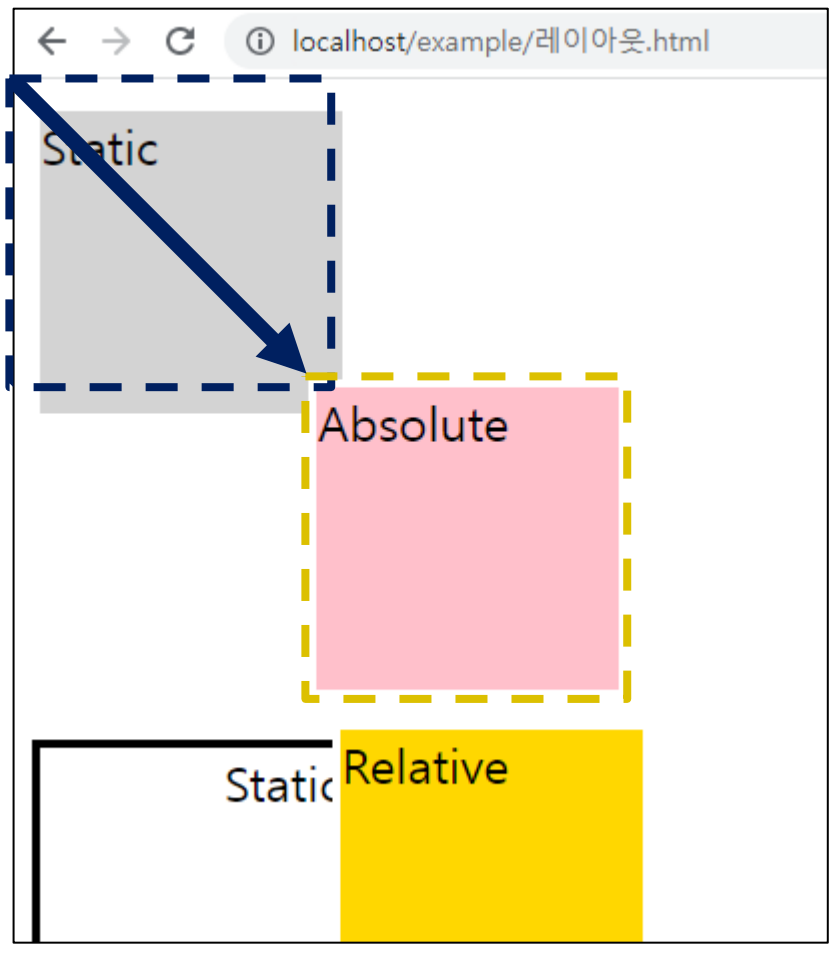
```
<style>
div {
  width: 100px;
  height: 100px;
  left: 100px;
  top: 100px;
  border: 3px solid white;
}
div#relative {
  position: relative;
  background: gold;
}
</style>
```



레이아웃 위치(4/6)

- 절대 위치 설정(absolute positioning)
  - 특정한 기준 위치에서 top, left, bottom, right 위치에 배치한다.

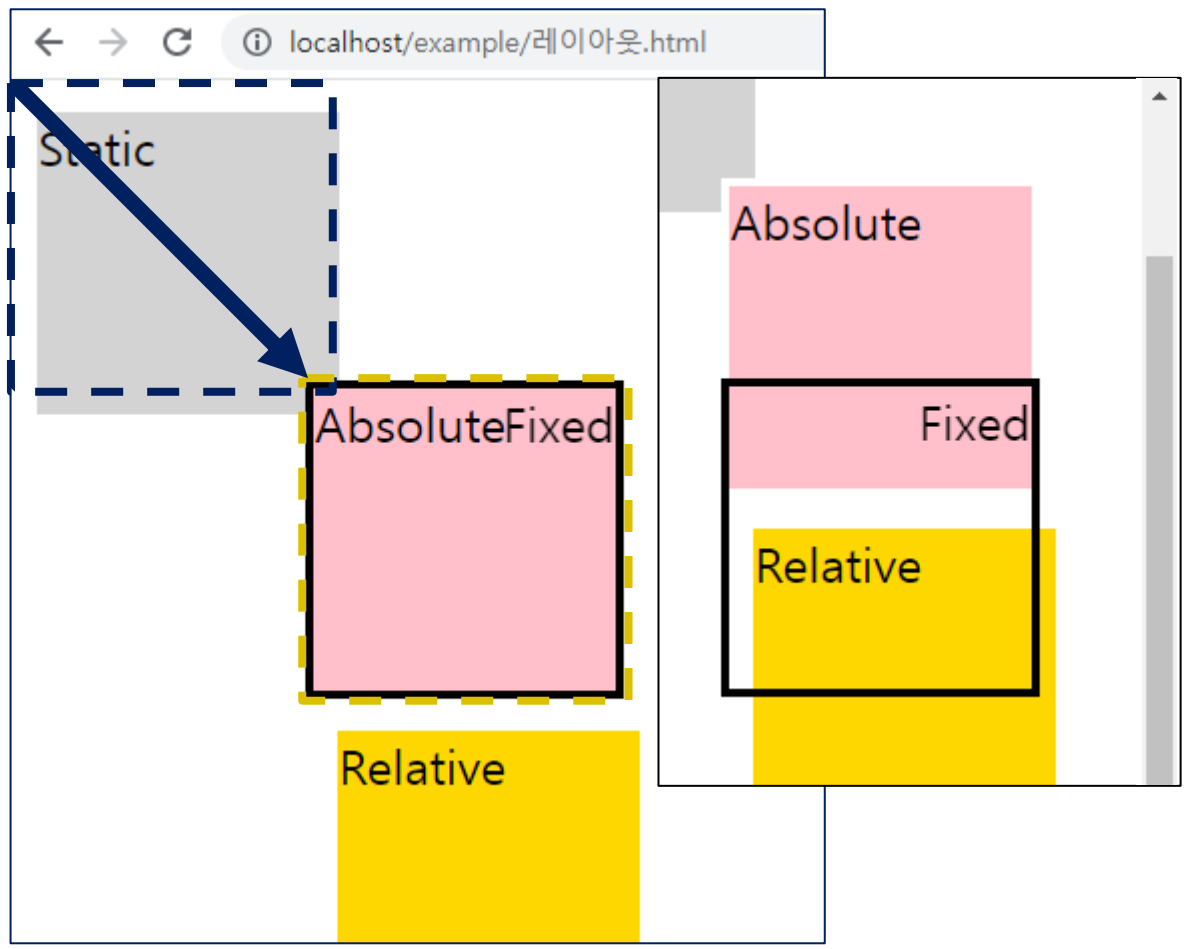
```
<style>
div {
  width: 100px;
  height: 100px;
  left: 100px;
  top: 100px;
  border: 3px solid white;
}
div#absolute {
  position: absolute;
  background-color: pink;
}
</style>
```



레이아웃 위치(5/6)

- 고정 위치 설정(fixed positioning)
  - 브라우저 윈도우(컨테이너)를 기준으로 방향 속성 값의 위치에 배치한다.
  - 페이지가 스크롤 되어도 항상 같은 곳에 배치한다.

```
<style>
div {
  width: 100px;
  height: 100px;
  left: 100px;
  top: 100px;
  border: 3px solid white;
}
div#fixed {
  position: fixed;
  border: 3px solid;
  text-align: right;
}
</style>
```

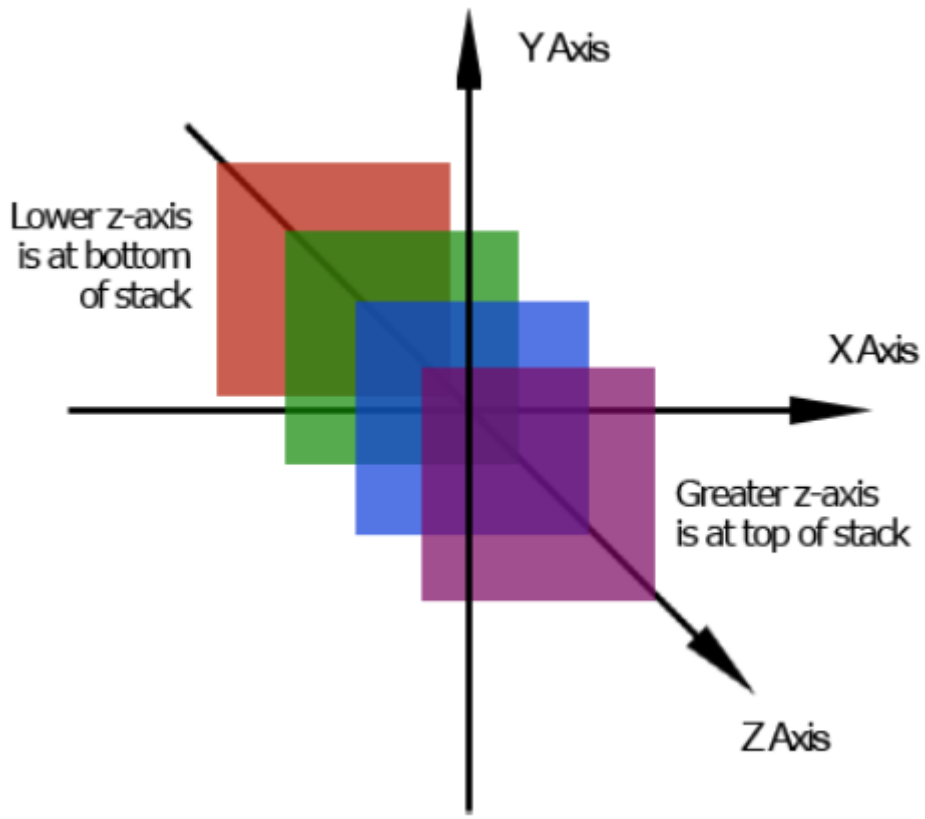


레이아웃 위치(6/6)

- 스택 순서 지정(z-index)
  - 요소가 배치되면서
    다른 요소와 겹치는 경우,
    z-index 속성으로 요소의
    스택 순서를 지정한다.

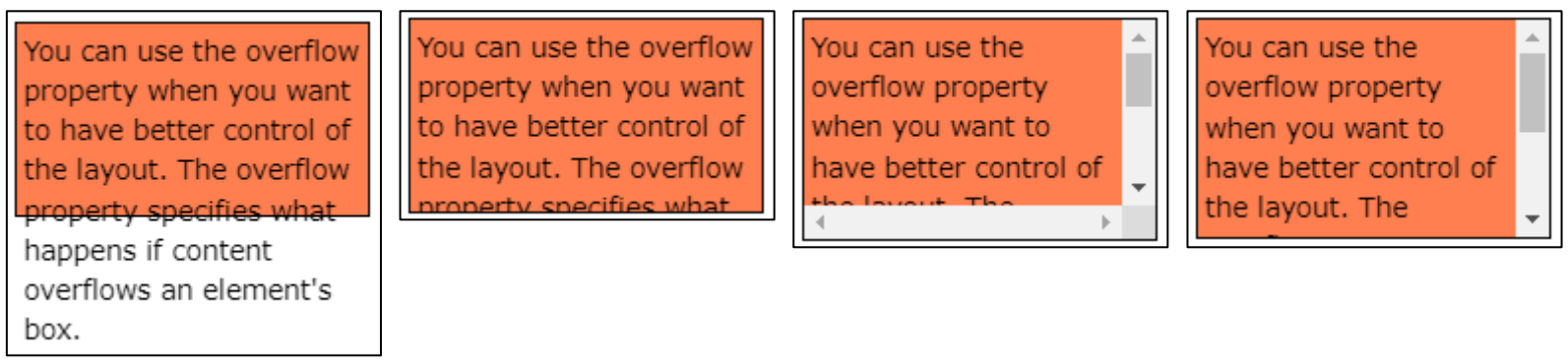
  - 양수와 음수를 사용할 수 있고,
    스택 순서가 큰 요소가
    낮은 요소 앞에 위치한다.

  - z-index가 지정되지 않고
    겹치면, HTML코드에서
    마지막에 배치된 요소가
    맨 위에 표시된다.



레이아웃 overflow

- overflow 속성 : 영역에 맞지 않는 콘텐츠를 제어
  - visible - 기본 값이지만 과적된 내용이 제어되지 않는 상태
  - hidden - 과적된 부분은 보이지 않는 상태
  - scroll - 스크롤 바가 추가되어 내용을 볼 수 있는 상태
  - auto - 필요 시 스크롤 바 생성



- 높이가 지정된 블록 요소에서만 작동
- overflow, overflow-x, overflow-y

float과 clear(1/3)

- float : 블록 레벨의 요소를 정렬할 때 많이 사용
  - 블록 레벨의 요소는 한 줄을 모두 차지하므로 다른 요소와 나란히 위치할 수 없다.

    (display: block => width 100%, 수직 정렬만 가능하다.)
  - float 속성 부여 시 요소의 너비 값이 최소 크기로 변경되며 빈 공간이 발생하는데 그 공간으로 다음 요소가 자리를 차지한다.

    (수평 정렬)

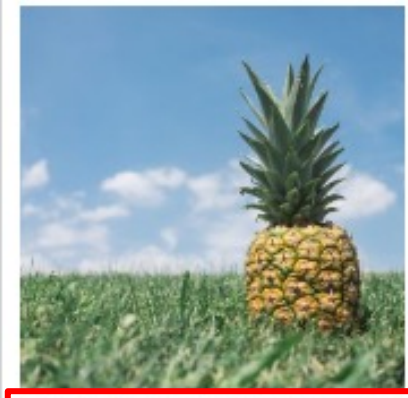
```
<style>
img {
  float: left;
}
</style>
</head>
<body>
<img src="pineapple.jpg" alt="Pineapple"
style="width:170px;height:170px;margin-right:15px;">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Phasellus imperdiet, nulla et dictum interdum, nisi lorem
egestas odio, vitae scelerisque enim ligula venenatis dolor.
Maecenas nisl est, ultrices nec congue eget, auctor vitae
massa. </p>
</body>
</html>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa.

float과 clear(2/3)

- clear : float 속성 중단 시 사용
  - float 속성이 부여된 요소의 옆 요소는 비어있는 영역을 채우듯 표현된다. 해당 속성을 적용하고 싶지 않은 경우 사용한다.

```
<style>
img {
  float: left;
}
#p {
  clear: left;
}
</style>
</head>
<body>
<img src="pineapple.jpg" alt="Pineapple"
style="width:170px;height:170px;margin-right:15px;">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
</p>
<p id="p">Phasellus imperdiet, nulla et dictum interdum, nisi
lorem egestas odio, vitae scelerisque enim ligula venenatis
dolor. Maecenas nisl est, ultrices nec congue eget, auctor
vitae massa.</p>
</body>
</html>
```

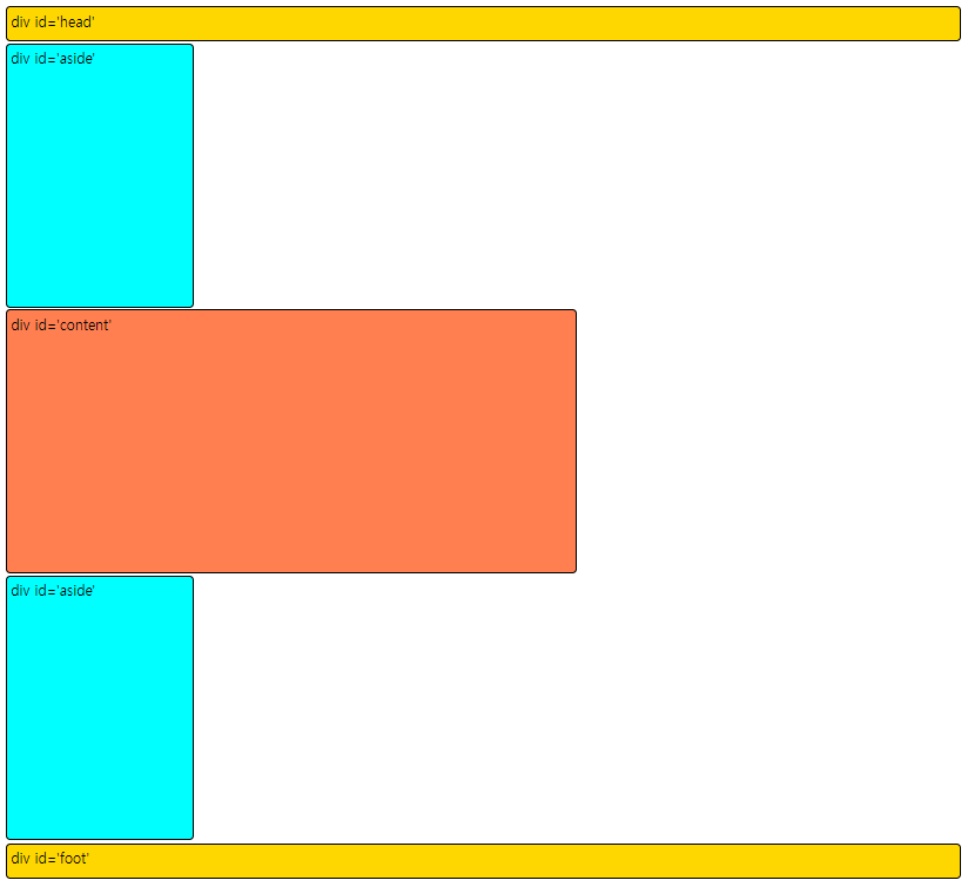


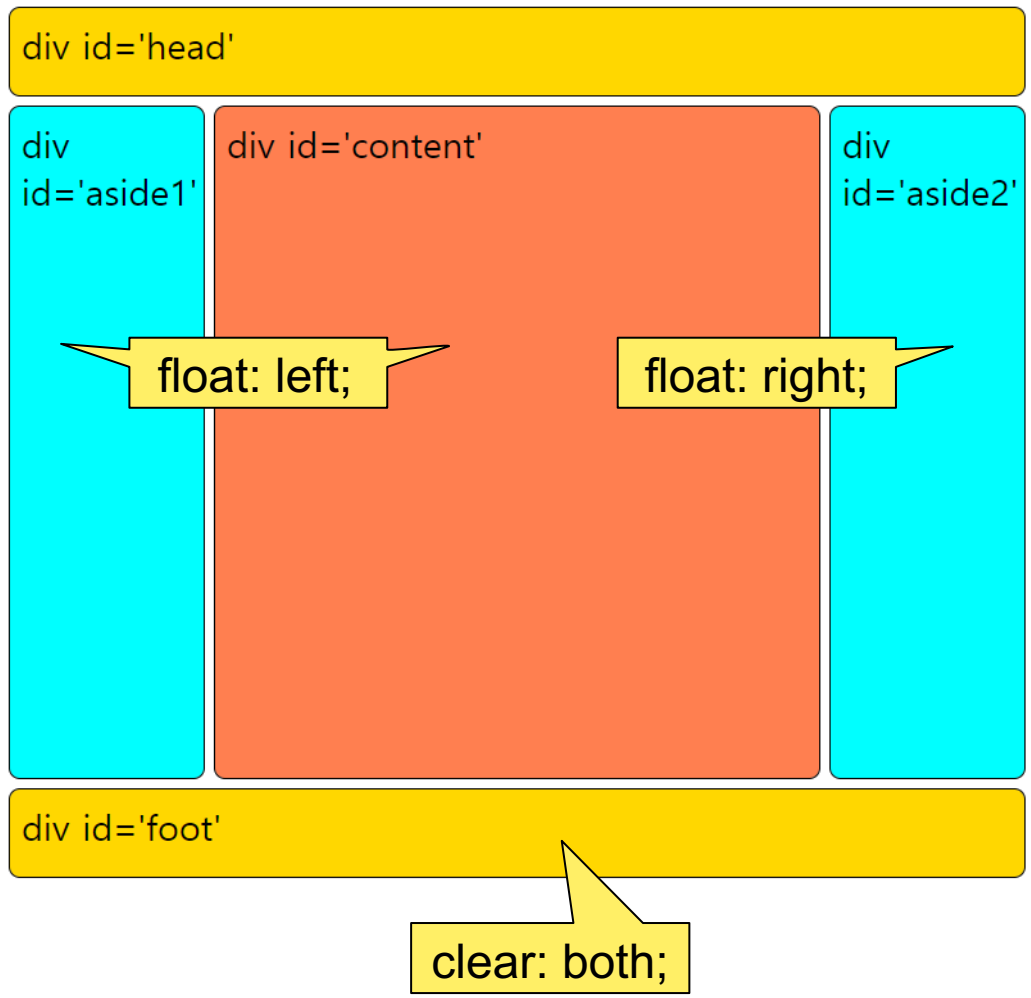
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa.

float과 clear(3/3)

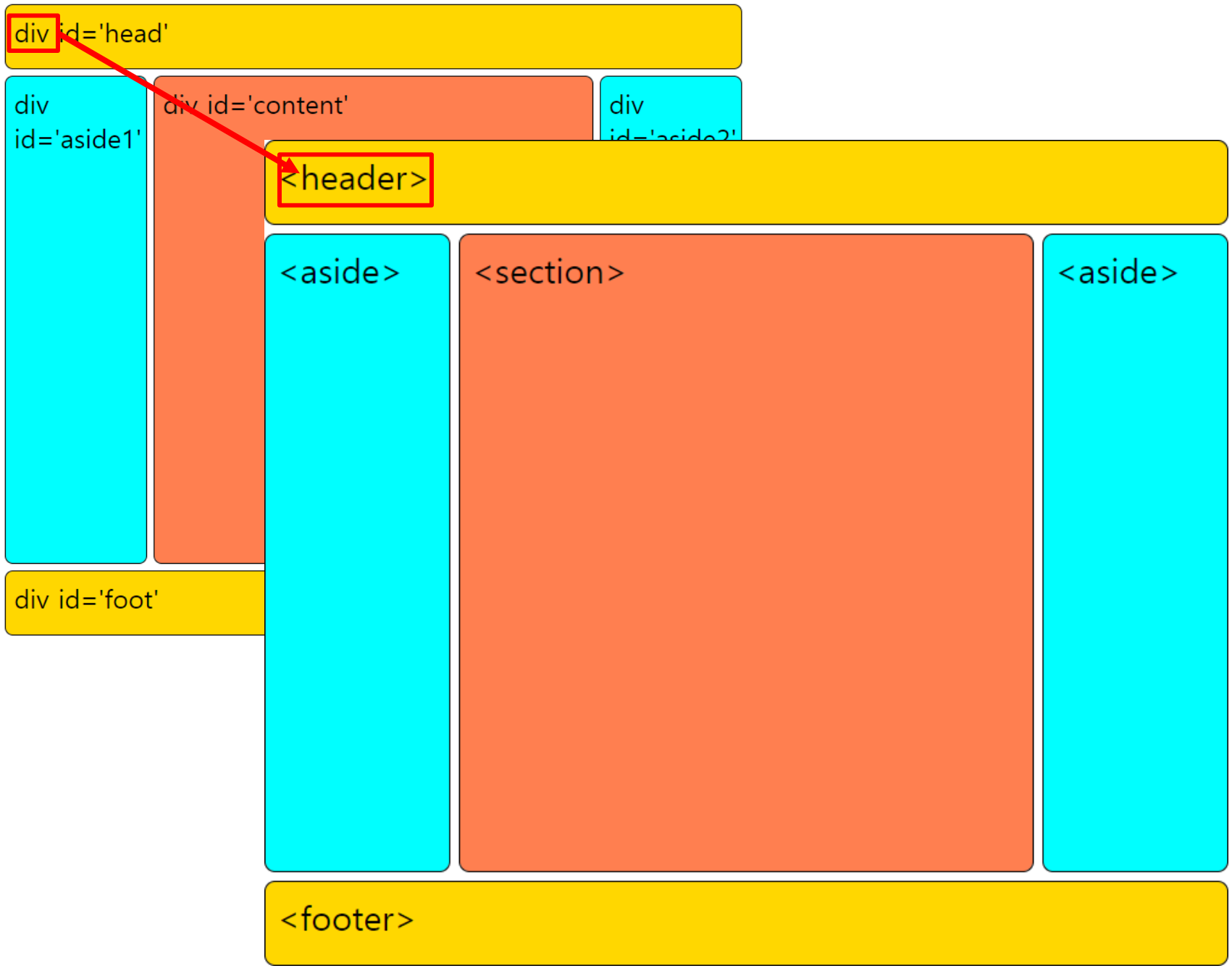
- div요소와 float 속성을 활용한 layout





웹사이트 레이아웃(1/3)

- 웹사이트 구조의 요소 변화 - 시맨틱 요소



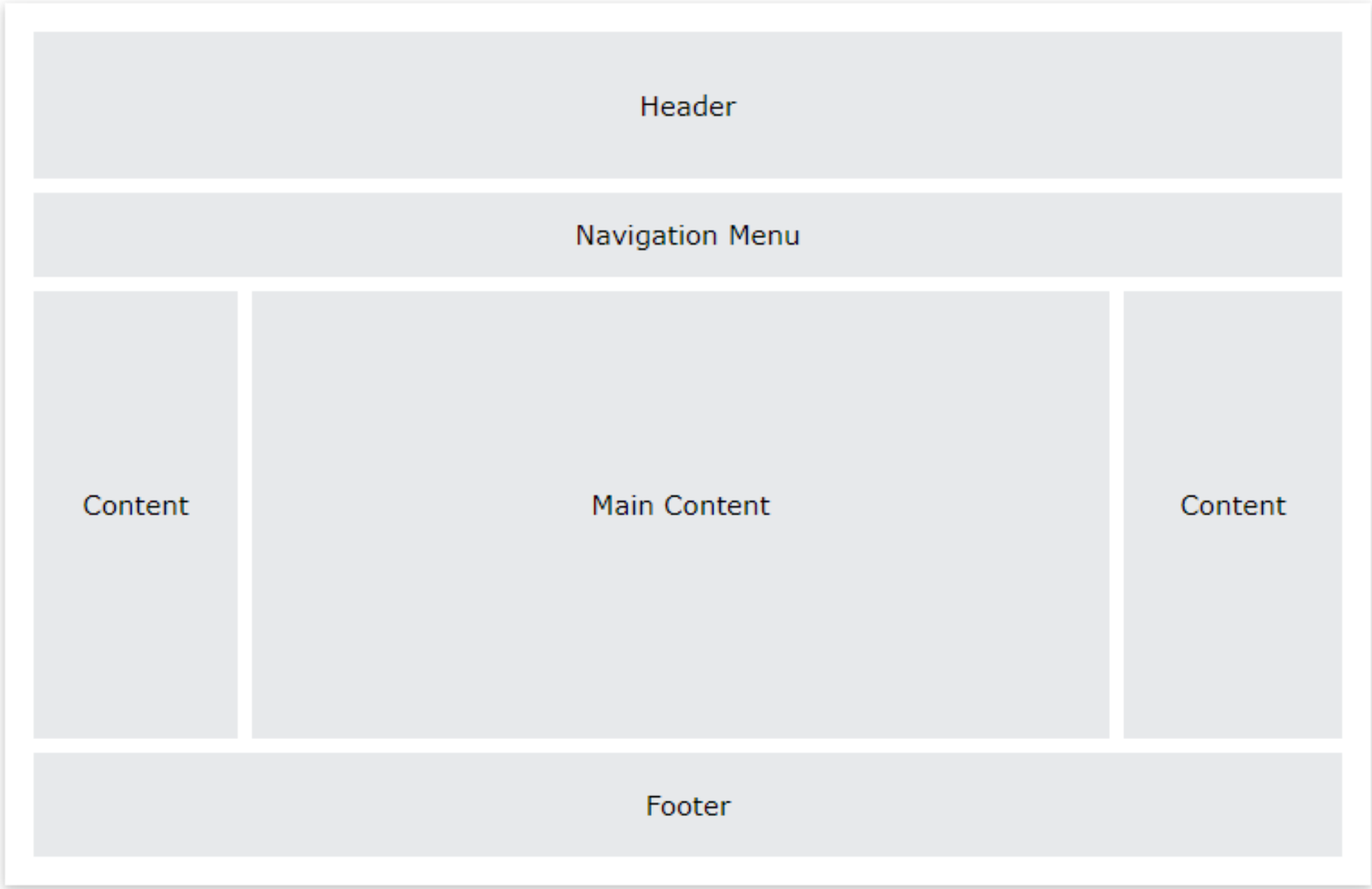
웹사이트 레이아웃(2/3)

- Semantic Elements : 웹 페이지의 구조를 쉽게 이해할 수 있도록 정의된 태그. 단순 컨테이너 역할
  - 개발자 간 혼란을 없애고, 검색 엔진 등에서 소스를 쉽게 판독

태그	설명
<header>	문서의 머리말로 홈페이지 상단에 로고나 메인 메뉴를 포함 headline tag 하나를 가지고 있어야 함
<footer>	문서의 꼬리말(바닥글)로서 한 페이지에 한 번만 작성 제작자의 정보, 연락처 정보, 저작권 정보 등을 나타냄
<section>	제목이 있는 주제별 콘텐츠 영역 (headline tag 하나를 가지고 있어야 함)
<article>	웹 사이트와 독립적인 콘텐츠 정의 시 사용 | 블로그의 포스트, 신문기사 등
<nav>	내비게이션 링크 | 메뉴, 목차, 인덱스 등을 표현
<aside>	좌우측 사이드 바 | 페이지 콘텐츠 외 콘텐츠 (배너, 퀵 메뉴 등)를 정의
<figure>	그림, 사진, 도표 등의 멀티미디어 콘텐츠 표시
<figcaption>	<figure>요소에 대한 캡션을 정의

웹사이트 레이아웃(3/3)

- 시맨틱 요소로 구성된 웹 사이트



연습2

header

My Blog Page

시멘틱 태그로 만들어진 페이지 입니다.

Links

- w3schools
- Naver
- Daum

스폰지밥

Semantic Tags

시멘틱 요소(Semantic elements)들은 브라우저에게 요소의 의미나 목적을 명확하게 알려주는 요소이다. 레이아웃을 만들때 쓰는 것. 의미가 있는 요소라면 div보다는 시멘틱 태그를 사용해준다.

div와 span

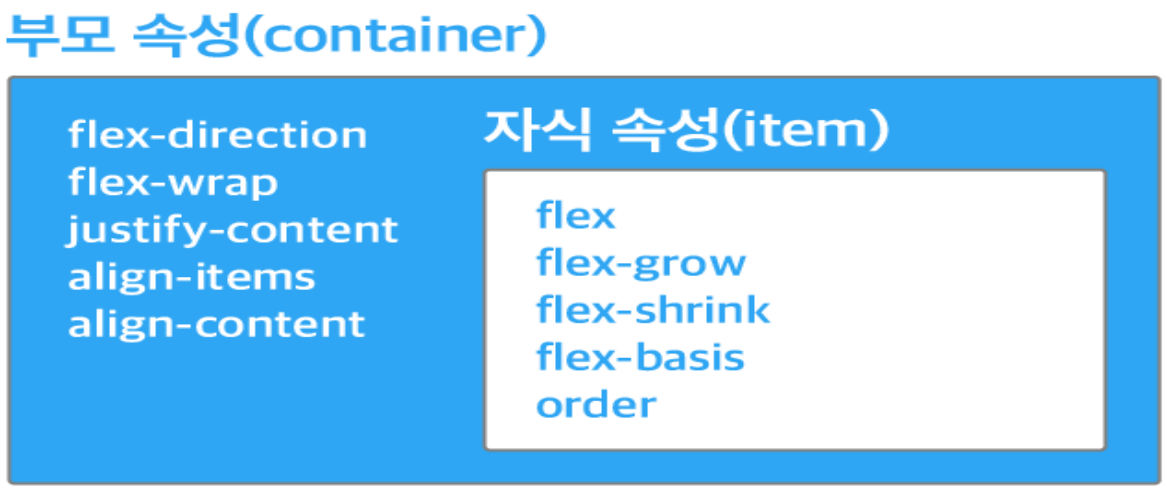
div은 "divide"의 약자로서 페이지를 논리적인 섹션으로 분리하는데 사용되는 태그이다. span요소는 인라인 요소로서 텍스트를 위한 컨테이너로 사용할 수 있다.

footer

Copyright (c)2019 Hong

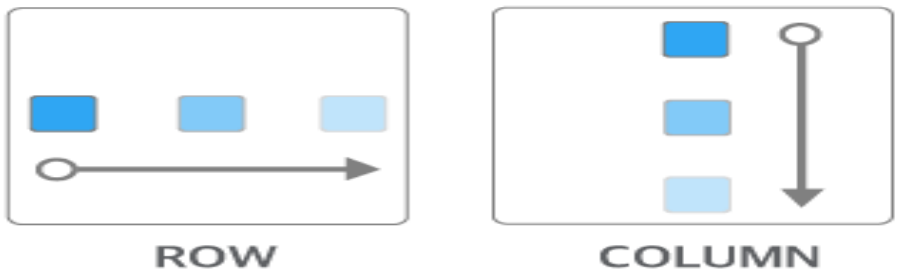
flex1

- flexbox는 뷰포트나 요소의 크기가 불명확하거나 동적으로 변할 때에도 효율적으로 요소를 배치, 정렬, 분산할 수 있는 방법을 제공하는 CSS3의 새로운 레이아웃 방식이다.
- flexbox의 장점을 한 마디로 표현하면 복잡한 계산 없이 요소의 크기와 순서를 유연하게 배치할 수 있다
- flexbox는 복수의 자식 요소인 flex item과 그 상위 부모 요소인 flex container로 구성된다.
- `.flex_container{ display: flex;}`

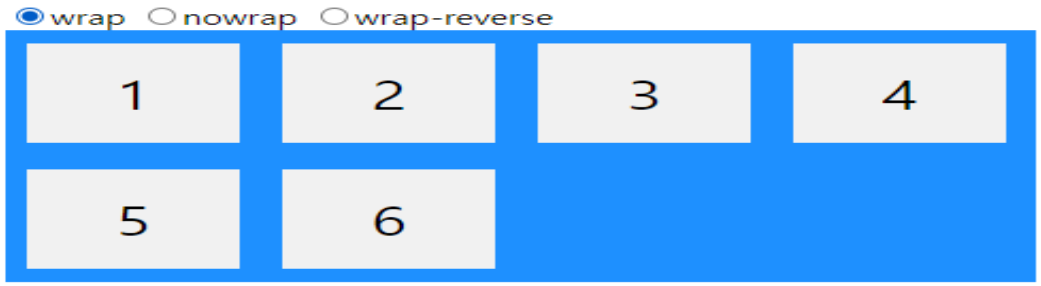


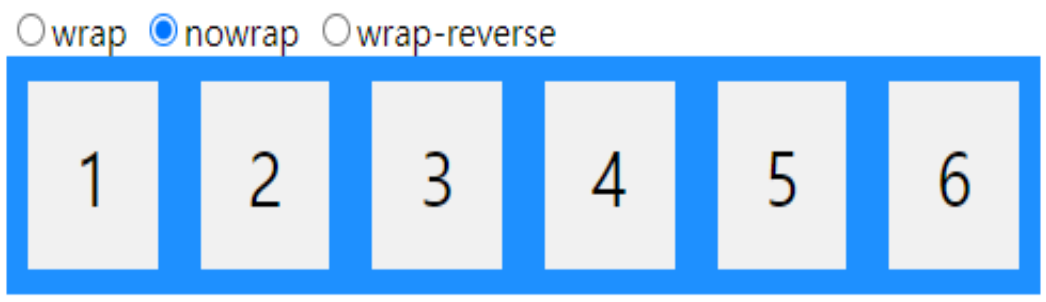
flex2

- flex-direction:



- flex-wrap:





flex3

- flex-direction 속성과 flex-wrap 속성을 flex-flow 속성으로 단축
- flex-flow:<flex-direction>||<flex-wrap>;
- **flex-flow:** column wrap;

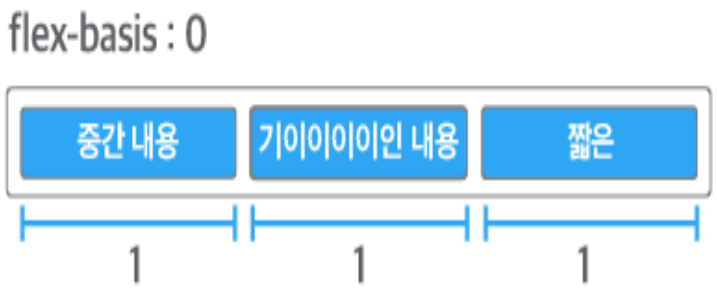
- flex-grow : flex item의 확장에 관련된 속성

- flex-shrink : flex item의 축소에 관련된 속성

flex4

- flex-basis 속성은 flex item의 기본 크기를 결정하는 속성

- flex-basis: 0 같은 크기로
- flex-basis: auto content 의 크기에 따라 달라짐



flex5

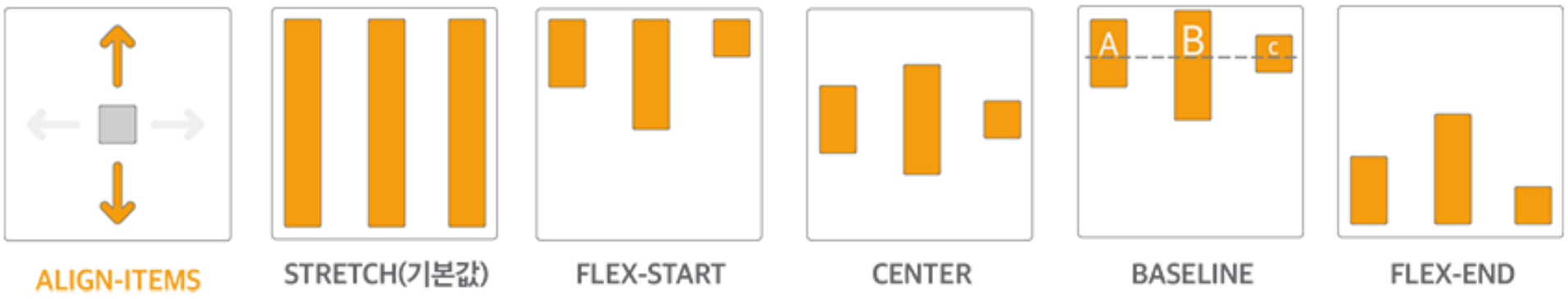
- flex： flex-grow 속성과 flex-shrink 속성, flex-basis 속성을 축약

- flex 속성의 기본값은 0 1 auto
- flex: 1 속성은 flex: 1 1 0
- flex: auto 속성은 flex: 1 1 auto

- lex: initial로 지정하면 flex: 0 1 auto
- flex: auto 로 지정하면 flex: 1 1 auto
- flex: none으로 지정하면 flex: 0 0 auto
- flex: 2는 flex: 2 1 0

flex5

- align-items: flex item을 수직으로 정렬



- justify-content: flex item을 수평으로 정렬