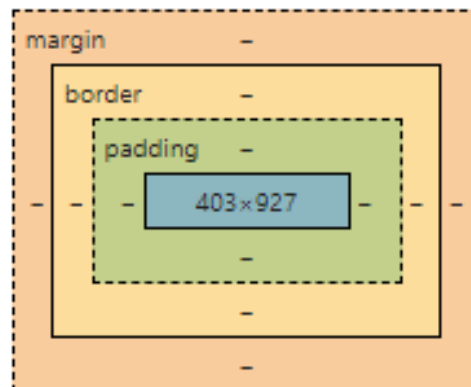


230308

Box Model

- CSS 원칙 1
 - 모든 요소는 네모(박스 모델)이고, 위에서부터 아래로, 왼쪽에서 오른쪽으로 쌓인다 (좌측 상단에 배치)
 - 하나의 박스는 네 부분으로 이루어짐
 - content
 - padding
 - border
 - margin
- Box Model 구성



- Content
 - 글이나 이미지 등
 - 요소의 실제 내용
- Padding
 - 테두리 안쪽의 내부 여백
 - 요소에 적용된 배경색
 - 이미지는 padding까지 적용

```
.margin-padding{
  margin: 10px;
  padding: 30px;
}
```

◦ Border

■ 테두리 영역

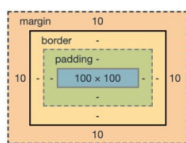
```
.border{
  border-width: 2px;
  border-style: dashed;
  border-color: black;
}
```

◦ Margin

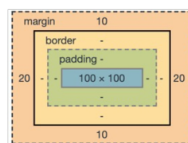
- 테두리 바깥의 외부여백
- 배경색을 지정할 수 없다

```
.margin{
  margin-top: 10px;
  margin-right: 10px;
  margin-bottom: 10px;
  margin-left: 10px;
}
```

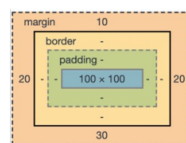
◦ 순서 기억(전체, 십자가, 나누기, 시계방향)



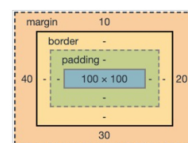
```
.margin-1 {
  margin: 10px;
}
```



```
.margin-2 {
  margin: 10px 20px;
}
```



```
.margin-3 {
  margin: 10px 20px 30px;
}
```



```
.margin-4 {
  margin: 10px 20px 30px 40px;
}
```

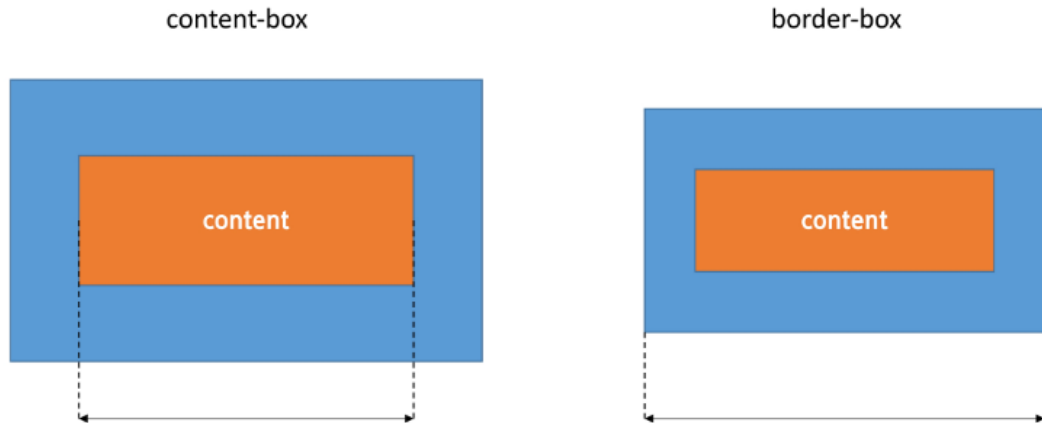
shorthand를 통해서 표현 가능하다.

• Box Sizing

◦ 기본적으로 모든 요소의 box-sizing은 content-box

- Padding을 제외한 순수 contents 영역만을 box로 지정

- 다만, 우리가 일반적으로 영역을 볼 때는 border까지의 너비를 100px 보는 것을 원함
 - 그 경우 box-sizing을 border-box로 설정



개발자 도구

- 크롬 개발자 도구
 - 웹 브라우저 크롬에서 제공하는 개발과 관련된 다양한 기능 제공
 - 주요 기능
 - Elements - DOM 탐색 및 CSS 확인 및 변경
 - Styles - 요소에 적용된 CSS 확인
 - Computed - 스타일이 계산된 최종 결과
 - Event Listeners - 해당 요소에 적용된 이벤트(JS)
 - Sources, Network, Performance, Application, Security, Audits 등

Box Model

- CSS 원칙 2
 - 모든 요소는 네모이고, 좌측 상단에 배치
 - display에 따라 크기와 배치가 달라진다
- 인라인 / 블록 요소
 - 블록 요소
 - 한줄 전체를 차지

- div를 2개 쓰면 밑으로 두줄
 - 너비, 높이, 마진을 지정할 수 있다
- 인라인 요소
 - in line : 줄 안에
 - 한 줄 전체를 차지 하지 않고 content 영역만큼만 차지
 - span같은 요소를 쓰면 그 content에 맞는 요소만큼만 차지
- 대표적으로 활용되는 display
 - display: block
 - 줄바꿈이 일어나는 요소(다른 elem을 밀어낸다!)
 - 화면 크기 전체의 가로 폭을 차지
 - 블록 레벨 요소 안에 인라인 레벨 요소가 들어갈 수 있음
 - 대표적인 블록 레벨 요소
 - div / ul, ol, li / p / hr / form 등
 - display: inline
 - 줄바꿈이 일어나지 않는 행의 일부 요소
 - content를 마크업 하고 있는 만큼만 가로 폭을 차지
 - width, height, margin-top, margin-bottom을 지정할 수 없음
 - 상하 여백은 line-height로 지정
 - 대표적인 인라인 레벨 요소
 - span / a / img / input, label / b, em, i, strong 등
- 속성에 따른 수평 정렬
 - box - 여백을 정렬

```
/*왼쪽*/
margin-right: auto;/*여백 오른쪽을 정렬*/
```

```
/*오른쪽*/
margin-left: auto;/*여백 왼쪽을 정렬*/
```

```
/*중간*/  
margin-right: auto;  
margin-left: auto; /*여백 오른쪽, 왼쪽 둘 다 정렬*/
```

- inline - 텍스트 를 정렬

```
text-align: left; /*텍스트 를 왼쪽에 정렬*/
```

```
text-align: right; /*텍스트 를 오른쪽에 정렬*/
```

```
text-align: center; /*텍스트 를 중간에 정렬*/
```

- 상하 여백은 line-height로 지정

display

- display: inline-block
 - block과 inline 레벨 요소의 특징을 모두 가짐
 - inline처럼 한 줄에 표시 가능하고, block처럼 width, height, margin 속성을 모두 지정할 수 있음
- display: none
 - 해당 요소를 화면에 표시하지 않고, 공간조차 부여되지 않음
 - 이와 비슷한 visibility: hidden은 해당 요소가 공간은 차지하나 화면에 표시만 하지 않는다
- 이 외 다양한 속성은 [여기](#)

CSS Position

- 모를때는 [여기로](#)
- 문서 상에서 요소의 위치를 지정(어떤 기준으로 어디에 배치 시킬지)
 - static
 - 모든 태그의 기본값(기준 위치)

- 일반적인 요소의 배치 순서에 따름(좌측 상단)
- 부모 요소 내에서 배치될 때는 부모 요소의 위치를 기준으로 배치됨

```
div{
  height: 100px;
  width : 100px;
  background-color: #9775fa;
  color: black;
  line-height: 100px;
  text-align: center;
}
```

○ 아래는 좌표를 사용하여 이동 가능

- relative : 상대 위치
 - 자기 자신의 static 위치를 기준으로 이동(normal flow 유지)
 - 레이아웃에서 요소가 차지하는 공간은 static 일 때와 같음(normal position 대비 offset)

```
.relative{
  position: relative;
  top: 100px;
  left: 100px;
}
```

- absolute : 절대 위치
 - 요소를 일반적인 문서 흐름에서 제거 후 레이아웃에 공간을 차지하지 않음(normal flow에서 벗어남)
 - static이 아닌 가장 가까이 있는 부모 / 조상 요소를 기준으로 이동(없는 경우 body)

```
.parent{
  position: relative;
}

.absolute-child{
  position: absolute;
  top: 50px;
  left: 50px;
}
```

- fixed : 고정 위치

- 요소를 일반적인 문서 흐름에서 제거 후 레이아웃에 공간을 차지하지 않음 (normal flow에서 벗어남)
- 부모 요소와 관계없이 viewport(실제로 보고있는 화면)를 기준으로 이동
 - 스크롤 시에도 항상 같은 곳에 위치함

```
.fixed{  
  position: fixed;  
  bottom: 0;  
  right: 0;  
}
```

■ sticky

- 스크롤에 따라 static → fixed로 변경
 - 속성을 적용한 박스는 평소에 문서 안에서 position: static 상태와 같이 일반적인 흐름에 따르지만, 스크롤 위치가 임계점에 이르면 position: fixed와 같이 박스를 화면에 고정할 수 있는 속성