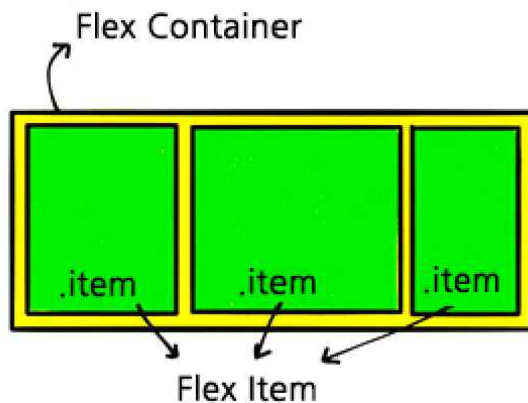


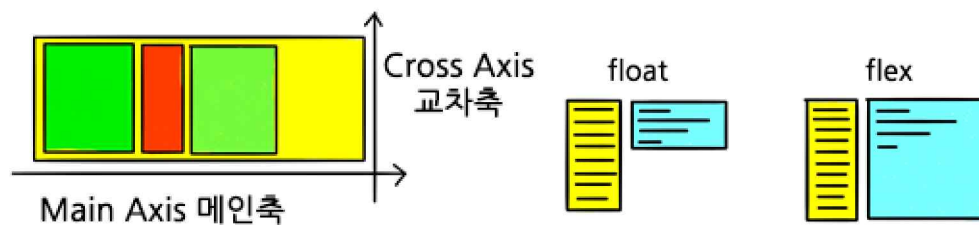
## CSS Flexbox

float 또는 position을 사용하지 않고도 유연한 반응형 레이아웃 구조를 쉽게 설계할 수 있다.

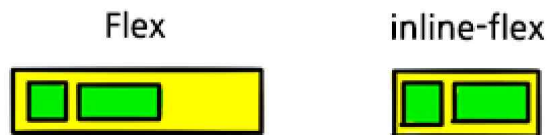
```
<div class="flex-container">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
</div>
```



display: flex;



display: inline-flex;



아이템들이 배치된 방향의 축을 메인축(Main Axis)  
메인축과 수직인 축을 수직축 또는 교차축(Cross Axis)

## Flex 컨테이너에 적용하는 속성들

1. flex-direction
2. flex-wrap
3. flex-flow
4. justify-content
5. align-items
6. align-content

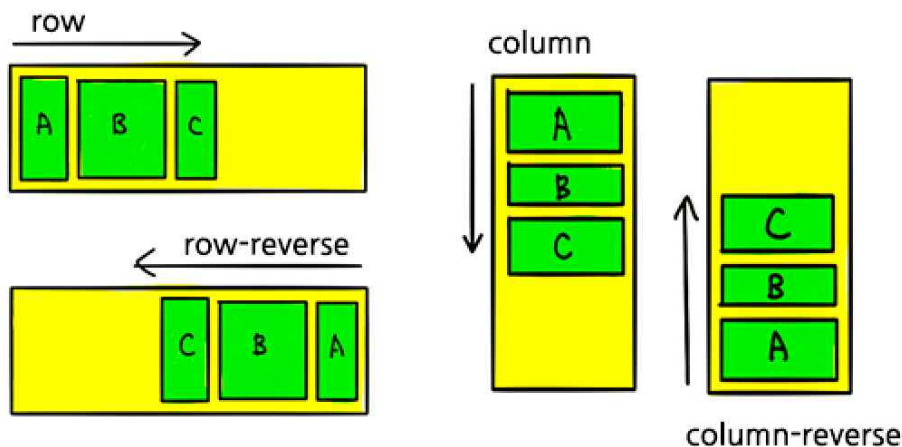
flex-direction : 컨테이너가 플렉스 항목의 방향을 정의

column - 플렉스 항목을 열(세로)방향으로 배치

column-reverse - 플렉스 항목을 역순으로 세로 배치

row(기본값) - 플렉스 항목을 가로로(왼쪽에서 오른쪽으로) 배치.

row-reverse - 플렉스 항목을 역순으로 가로 배치



flex-wrap 줄넘김 처리 설정

컨테이너가 더 이상 아이템들을 한줄에 담을 여유 공간이 없을 때 아이템 줄바꿈을 어떻게 할지 결정하는 속성

nowrap (기본값)

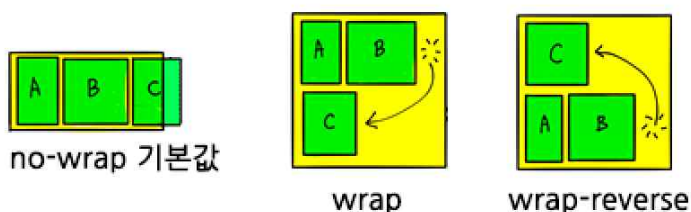
줄바꿈을 하지 않습니다. 넘치면 그냥 빠져 나감.

wrap

줄바꿈을 합니다. float이나 inline-block으로 배치한 요소들과 비슷하게 동작

wrap-reverse

줄바꿈을 하는데, 아이템을 역순으로 배치



flex-flow

flex-direction과 flex-wrap을 한꺼번에 지정할 수 있는 단축 속성이예요.

flex-direction, flex-wrap의 순으로 한 칸 떼고 기입.

justify-content속성은 플렉스 항목을 정렬하는 데 사용(메인축)

flex-start (기본값)

아이템들을 시작점으로 정렬

flex-direction이 row(가로 배치)일 때는 왼쪽, column(세로 배치)일 때는 위

flex-end

아이템들을 끝점으로 정렬

flex-direction이 row(가로배치)일 때는 오른쪽, column(세로배치)일 때는 아래

center

아이템들을 가운데로 정렬합니다.

space-between

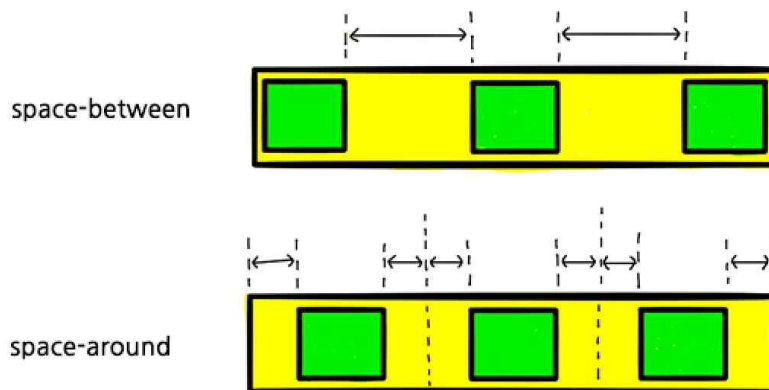
아이템들의 "사이(between)"에 균일한 간격

space-around

아이템들의 "둘레(around)"에 균일한 간격

space-evenly

항목 주위에 균일한 간격



align-items : 수직축 방향 정렬

stretch (기본값)

아이템들이 수직축 방향으로 끝까지 쭉욱 늘어남.

flex-start

아이템들을 시작점으로 정렬

flex-direction이 row(가로 배치)일 때는 위, column(세로 배치)일 때는 왼쪽

flex-end

아이템들을 끝으로 정렬합니다.

flex-direction이 row(가로 배치)일 때는 아래, column(세로 배치)일 때는 오른쪽

center  
아이템들을 가운데로 정렬합니다.

baseline  
아이템들을 텍스트 베이스라인 기준으로 정렬합니다.

align-content - 여러 행 정렬  
flex-wrap: wrap;이 설정된 상태에서, 아이템들의 행이 2줄 이상 되었을 때의 수직축 방향 정렬을 결정하는 속성

```
.flex-container {  
    flex-wrap: wrap;  
    align-content: stretch;  
    /* align-content: flex-start; */  
    /* align-content: flex-end; */  
    /* align-content: center; */  
    /* align-content: space-between; */  
    /* align-content: space-around; */  
    /* align-content: space-evenly; */  
}
```

## Flex 아이템에 적용하는 속성들

1. flex-basis
2. flex-grow
3. flex-shrink
4. flex
5. align-self
6. order

flex-basis는 Flex 아이템의 기본 크기를 설정합니다(flex-direction이 row일 때는 너비, column일 때는 높이).

```
.item {  
    flex-basis: 100px;  
}
```

원래의 width가 100px이 안되는 AAA와 CCC는 100px로 늘어났고, 원래 100px이 넘는 BBB는 그대로 유지



flex-grow – 유연하게 늘리기

아이템들의 flex-basis를 제외한 여백 부분을 flex-grow에 지정된 숫자의 비율로 나누어 가진다.

flex-grow는 아이템이 flex-basis의 값보다 커질 수 있는지를 결정하는 속성  
flex-grow에는 숫자값이 들어가는데, 몇이든 일단 0보다 큰 값이 세팅이 되면 해당 아이템이 유연한(Flexible) 박스로 변하고 원래의 크기보다 커지며 빈 공간을 메우게 됩니다.

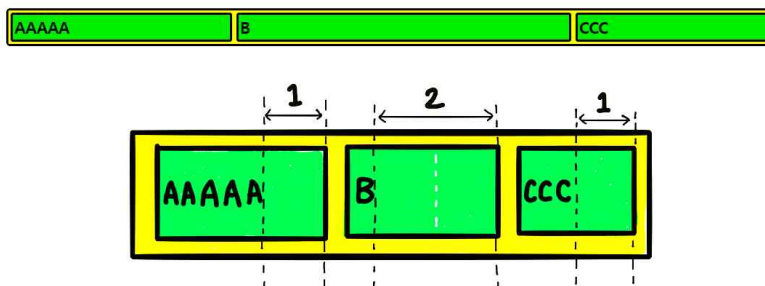
기본값: 0

/\* 1:2:1의 비율로 세팅할 경우 \*/

```
.item:nth-child(1) { flex-grow: 1; }
```

```
.item:nth-child(2) { flex-grow: 2; }
```

```
.item:nth-child(3) { flex-grow: 1; }
```



flex-shrink - 유연하게 줄이기

flex-shrink는 flex-grow와 반대 속성으로, 아이템이 flex-basis의 값보다 작아질 수 있는지를 결정

flex-shrink에는 숫자값이 들어가는데, 몇이든 일단 0보다 큰 값이 세팅이 되면 해당 아이템이 유연한(Flexible) 박스로 변하고 flex-basis보다 작아집니다.

기본값이 1이기 때문에 따로 세팅하지 않았어도 아이템이 flex-basis보다 작아질 수 있다.

```
.flex-container {  
    display: flex;
```

```
}
```

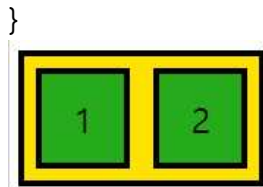
```
.item {  
    flex-basis: 200px;
```

```
}
```



```
.flex-container {  
    display: flex;  
    width: 200px;  
}
```

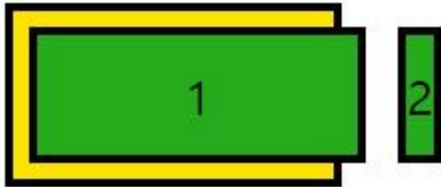
```
.item {  
    flex-basis: 200px;
```



```

}
.flex-container {
  display: flex;
  width: 200px;
}
.item {
  flex-basis: 200px;
}
.item:first-child {
  flex-shrink: 0;
}

```



flex

flex-grow, flex-shrink, flex-basis를 한 번에 쓸 수 있는 축약형 속성

```

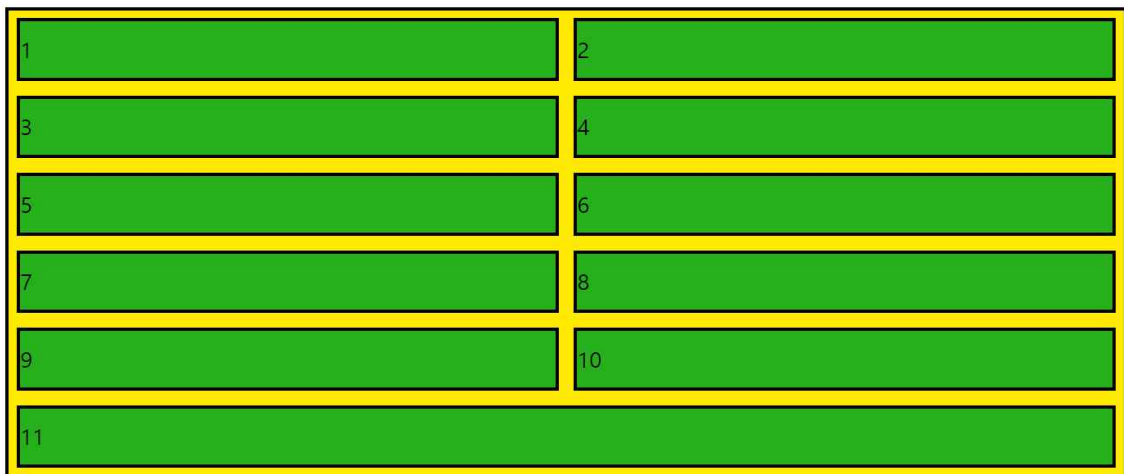
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-wrap: wrap;
  background-color: #ffe300;
  border: 5px solid #000;
}
.flex-container > div {
  background-color: #26ab1c;
  color: #000;
  border: 5px solid #000;
  line-height: 75px;
  font-size: 30px;
  margin: 10px;
}
.item {
  flex: 1 1 40%;
}

```

```

</style>
</head>
<body>
<div class="flex-container">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
  <div class="item">5</div>
  <div class="item">6</div>
  <div class="item">7</div>
  <div class="item">8</div>
  <div class="item">9</div>
  <div class="item">10</div>
  <div class="item">11</div>
</div>
</body>
</html>

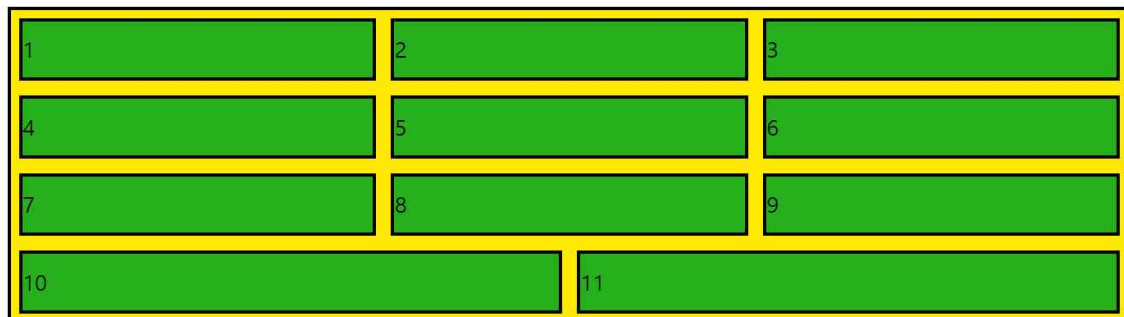
```



```

.item {
  flex: 1 1 30%;
}

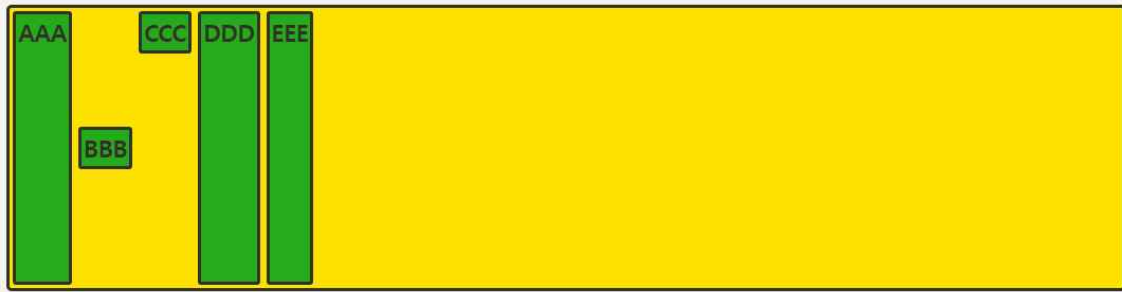
```



align-self

align-items가 전체 아이템의 수직축 방향 정렬이라면, align-self는 해당 아이템의 수직축 방향 정렬.

align-self 값을 BBB는 center, CCC는 flex-start로 설정



order - 배치 순서

각 아이템들의 시각적 나열 순서를 결정하는 속성

숫자값이 들어가며, 작은 숫자일 수록 먼저 배치

기본값은 0

```
.item:nth-child(1) { order: 3; } /* A */
```

```
.item:nth-child(2) { order: 1; } /* B */
```

```
.item:nth-child(3) { order: 2; } /* C */
```





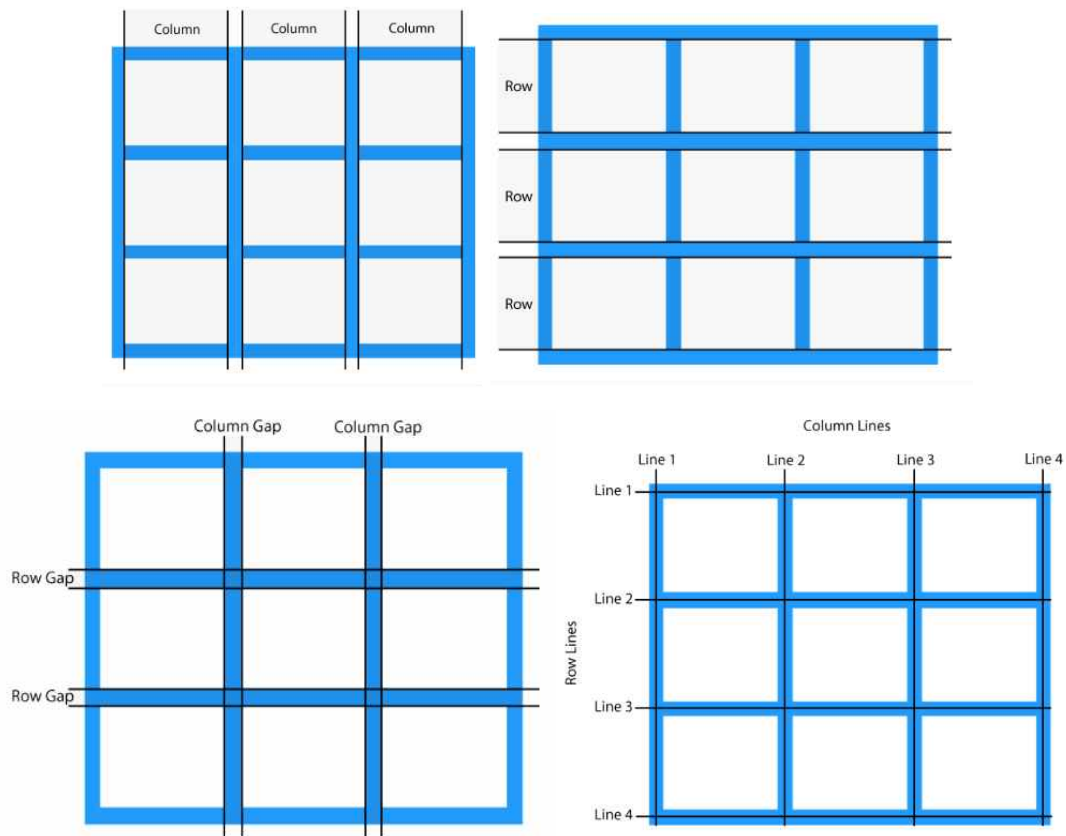
# CSS grid

CSS Grid(그리드)는 2차원(행과 열)의 레이아웃 시스템을 제공합니다.

Flexible Box도 훌륭하지만 비교적 단순한 1차원 레이아웃을 위하여, 좀 더 복잡한 레이아웃을 위해 우리는 CSS Grid를 사용할 수 있습니다.

**CSS 그리드 레이아웃(Grid Layout)**은 페이지를 여러 주요 영역으로 나누거나, 크기와 위치 및 문서 계층 구조의 관점에서 HTML 기본 요소로 작성된 컨트롤 간의 관계를 정의하는 데 아주 탁월합니다.

테이블과 마찬가지로 그리드 레이아웃은 세로 열과 가로 행을 기준으로 요소를 정렬할 수 있습니다. 하지만, 테이블과 달리 CSS 그리드는 다양한 레이아웃을 훨씬 더 쉽게 구현할 수 있습니다. 예를 들어, 그리드 컨테이너 속 자식 요소를, 마치 CSS로 일일이 위치를 지정해 준 것처럼, 실제로 겹치게 층을 지면서 자리를 잡도록 각 요소의 위치를 지정해 줄 수도 있습니다.



## Grid Container Properties

속성	의미
display	그리드 컨테이너(Container)를 정의
grid-template-rows	명시적 행(Track)의 크기를 정의
grid-template-columns	명시적 열(Track)의 크기를 정의
grid-template-areas	영역(Area) 이름을 참조해 템플릿 생성
grid-template	grid-template-xxx의 단축 속성
row-gap(grid-row-gap)	행과 행 사이의 간격(Line)을 정의
column-gap(grid-column-gap)	열과 열 사이의 간격(Line)을 정의
gap(grid-gap)	xxx-gap의 단축 속성
grid-auto-rows	암시적인 행(Track)의 크기를 정의
grid-auto-columns	암시적인 열(Track)의 크기를 정의
grid-auto-flow	자동 배치 알고리즘 방식을 정의
grid	grid-template-xxx과 grid-auto-xxx의 단축 속성
align-content	그리드 콘텐츠(Grid Contents)를 수직(열 축) 정렬
justify-content	그리드 콘텐츠를 수평(행 축) 정렬
place-content	align-content와 justify-content의 단축 속성
align-items	그리드 아이템(Items)들을 수직(열 축) 정렬
justify-items	그리드 아이템들을 수평(행 축) 정렬
place-items	align-items와 justify-items의 단축 속성

## Grid Item Properties

속성	의미
grid-row-start	그리드 아이템(Item)의 행 시작 위치 지정
grid-row-end	그리드 아이템의 행 끝 위치 지정
grid-row	grid-row-xxx의 단축 속성(행 시작/끝 위치)
grid-column-start	그리드 아이템의 열 시작 위치 지정
grid-column-end	그리드 아이템의 열 끝 위치 지정
grid-column	grid-column-xxx의 단축 속성(열 시작/끝 위치)
grid-area	영역(Area) 이름을 설정하거나, grid-row와 grid-column의 단축 속성
align-self	단일 그리드 아이템을 수직(열 축) 정렬
justify-self	단일 그리드 아이템을 수평(행 축) 정렬
place-self	align-self와 justify-self의 단축 속성
order	그리드 아이템의 배치 순서를 지정
z-index	그리드 아이템의 쌓이는 순서를 지정

```
<div class="grid-container">
  <div class="grid-item">1</div>
  <div class="grid-item">2</div>
  <div class="grid-item">3</div>
  <div class="grid-item">4</div>
  <div class="grid-item">5</div>
  <div class="grid-item">6</div>
  <div class="grid-item">7</div>
  <div class="grid-item">8</div>
  <div class="grid-item">9</div>
</div>
```

### 1. 그리드 형태 정의

grid-template-rows  
grid-template-columns

```
.grid-container {
  grid-template-columns: 200px 200px 500px;
  /* grid-template-columns: 1fr 1fr 1fr */
  /* grid-template-columns: repeat(3, 1fr) */
  /* grid-template-columns: 200px 1fr */
  /* grid-template-columns: 100px 200px auto */

  grid-template-rows: 200px 200px 500px;
  /* grid-template-rows: 1fr 1fr 1fr */
  /* grid-template-rows: repeat(3, 1fr) */
  /* grid-template-rows: 200px 1fr */
  /* grid-template-rows: 100px 200px auto */
}
```

grid-template-rows는 행(row)의 배치, grid-template-columns는 열(column)의 배치를 결정합니다.  
grid-template-columns: 200px 200px 500px;-->column을 200px, 200px, 500px

grid-template-columns: 1fr 1fr 1fr;

fr은 fraction(뜻은 여기로[새 창])인데요, 숫자 비율대로 트랙의 크기를 나눕니다.

즉 위의 1fr 1fr 1fr은 균일하게 1:1:1 비율인 3개의 column을 만들겠다는 의미예요.

고정 크기와 가변 크기를 섞어 쓸 수도 있습니다.

첫번째 column은 100px로 고정되고, 나머지 두번째 세번째 column은 2:1의 비율로 유연하게 움직이게 됩니다.

grid-template-columns: 100px 2fr 1fr;

repeat 함수

```
.container {  
    grid-template-columns: repeat(5, 1fr);  
    /* grid-template-columns: 1fr 1fr 1fr 1fr 1fr */  
}
```

repeat는 반복되는 값을 자동으로 처리할 수 있는 함수입니다.

repeat(반복횟수, 반복값)

즉, 위 코드의 repeat(5, 1fr)은 1fr 1fr 1fr 1fr 1fr과 같습니다.

repeat(3, 1fr 4fr 2fr); 이런 식의 패턴도 가능합니다.

minmax 함수

최소값과 최대값을 지정할 수 있는 함수입니다.

minmax(100px, auto)의 의미는 최소한 100px, 최대는 자동으로(auto) 늘어나게 입니다. 즉 아무리 내용의 양이 적더라도 최소한 높이 100px은 확보하고, 내용이 많아 100px이 넘어가면 알아서 늘어나도록 처리해 준 예시입니다.

```
.grid-container {  
    grid-template-columns: repeat(3, 1fr);  
    grid-template-rows: repeat(3, minmax(100px, auto));  
}
```

auto-fill과 auto-fit

auto-fill과 auto-fit은 column의 개수를 미리 정하지 않고 설정된 너비가 허용하는 한 최대한 셀을 채웁니다.

```
.grid-container" {  
    grid-template-columns: repeat(auto-fill, minmax(20%, auto));  
}
```

## 2. 간격 만들기

row-gap, column-gap

gap

그리드 셀 사이의 간격을 설정합니다.

```
.grid-container {  
    row-gap: 10px;  
    /* row의 간격을 10px로 */  
    column-gap: 20px;  
    /* column의 간격을 20px로 */  
}  
  
.grid-container {  
    gap: 10px 20px;  
    /* row-gap: 10px; column-gap: 20px; */  
}
```

## 3. 그리드 형태를 자동으로 정의

grid-auto-columns

grid-auto-rows

grid-template-columns(또는 grid-template-rows)의 통제를 벗어난 위치에 있는 트랙의 크기를 지정하는 속성입니다.

니다. 속성 이름이 헛갈린다면 -template- 자리에 - auto-가 들어간다고 생각하세요.

```
.grid-container {  
  grid-template-rows: repeat(3, minmax(100px, auto));  
}
```

각 셀마다 최소 100px의 높이를 확보하고, 콘텐츠가 높이 100px을 넘어가면 알아서 자동으로 늘어나도록(auto)  
-->row 개수를 미리 알 수 없는 경우에 grid-auto-rows

#### 4. 각 셀의 영역 지정

grid-column-start

grid-column-end

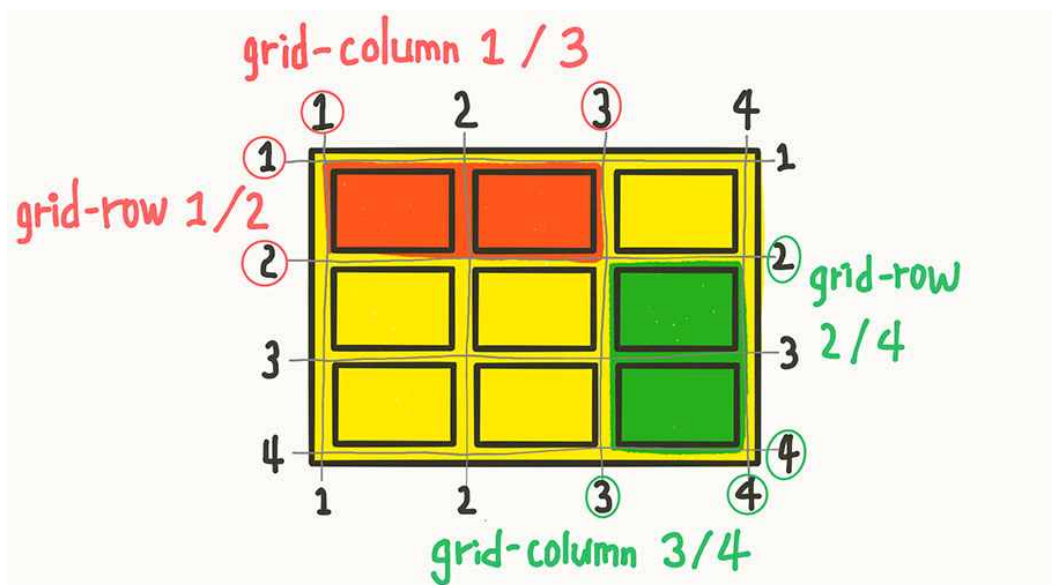
grid-column

grid-row-start

grid-row-end

grid-row

이 속성들은 Grid 아이템에 적용하는 속성으로, 각 셀의 영역을 지정합니다.



1부터 4까지의 Grid 라인 번호가 매겨져 있는데요, 바로 그 번호를 이용해서 column과 row의 범위를 결정하는 겁니다.

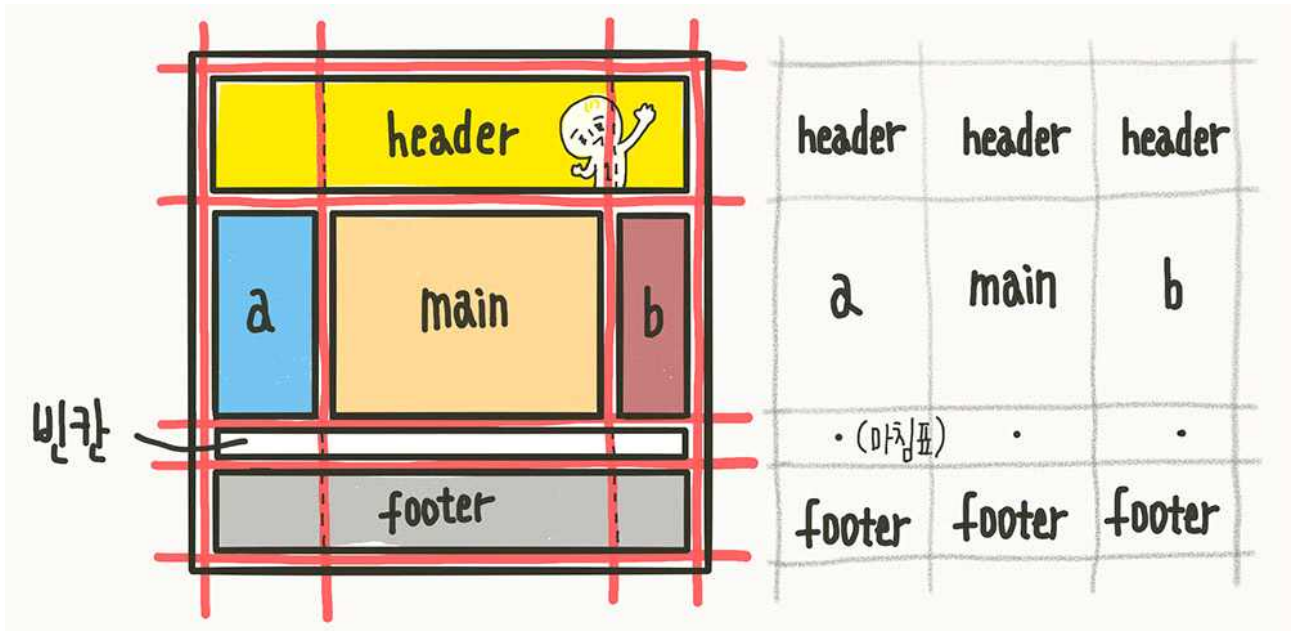
column으로 살펴보면, grid-column-start가 시작 번호, grid-column-end가 끝 번호입니다. grid-column은 start와 end 속성을 한번에 쓰는 축약형입니다.

```
.grid-item{  
  grid-column-start: 1;  
  grid-column-end: 3;  
  grid-row-start: 1;  
  grid-row-end: 2;  
}  
  
.grid-item:nth-child(1) {  
  grid-column: 1 / 3;  
  grid-row: 1 / 2;  
}  
  
.grid-item:nth-child(1) {  
  /* 1번 라인에서 2칸 */  
  grid-column: 1 / span 2;  
  /* 1번 라인에서 3칸 */  
  grid-row: 1 / span 3;  
}
```

## 5. 영역 이름으로 그리드 정의

grid-template-areas

각 영역(Grid Area)에 이름을 붙이고, 그 이름을 이용해서 배치하는 방법입니다.



```
.grid-container {
  grid-template-areas:
    "header header header"
    "  a    main    b  "
    "  .    .    .  "
    "footer footer footer";
}
.header { grid-area: header; }
.sidebar-a { grid-area: a; }
.main-content { grid-area: main; }
.sidebar-b { grid-area: b; }
.footer { grid-area: footer; }
/* 이름 값에 따옴표가 없는 것에 주의하세요 */
```

## 6. 자동 배치

grid-auto-flow

아이템이 자동 배치되는 흐름을 결정하는 속성입니다.

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(25%, auto));
  grid-template-rows: repeat(5, minmax(50px, auto));
  grid-auto-flow: dense;
}
.grid-item:nth-child(2) { grid-column: auto / span 3; }
.grid-item:nth-child(5) { grid-column: auto / span 3; }
.grid-item:nth-child(7) { grid-column: auto / span 2; }
```

## 7. 세로 방향 정렬

### align-items

아이템들을 세로(column축) 방향으로 정렬합니다. 컨테이너에 적용합니다.

```
.grid-container {
  align-items: stretch;
  /* align-items: start; */
  /* align-items: center; */
  /* align-items: end; */
}
```

## 8. 가로 방향 정렬

### justify-items

아이템들을 가로(row축) 방향으로 정렬합니다. 컨테이너에 적용합니다.

```
.grid-container {
  justify-items: stretch;
  /* justify-items: start; */
  /* justify-items: center; */
  /* justify-items: end; */
}
```

### place-items

align-items와 justify-items를 같이 쓸 수 있는 단축 속성이에요.

align-items, justify-items의 순서로 작성하고, 하나의 값만 쓰면 두 속성 모두에 적용됩니다.

```
.grid-container {
  place-items: center start;
}
```

## 9. 아이템 그룹 세로 정렬

### align-content

Grid 아이템들의 높이를 모두 합한 값이 Grid 컨테이너의 높이보다 작을 때 Grid 아이템들을 통째로 정렬합니다.

값	의미	기본값
normal	stretch와 같습니다.	normal
start	시작점(위쪽) 정렬	
center	수직 가운데 정렬	
end	끝점(아래쪽) 정렬	
space-around	각 행 위아래에 여백을 고르게 정렬	
space-between	첫 행은 시작점에, 끝 행은 끝점에 정렬되고 나머지 여백으로 고르게 정렬	
space-evenly	모든 여백을 고르게 정렬	
stretch	열 축을 채우기 위해 그리드 콘텐츠를 늘림	

```
.grid-container {
  align-content: stretch;
  /* align-content: start; */
  /* align-content: center; */
  /* align-content: end; */
  /* align-content: space-between; */
  /* align-content: space-around; */
  /* align-content: space-evenly; */
}
```

## 10. 아이템 그룹 가로 정렬

justify-content

Grid 아이템들의 너비를 모두 합한 값이 Grid 컨테이너의 너비보다 작을 때 Grid 아이템들을 통째로 정렬합니다.

```
.grid-container {  
    justify-content: stretch;  
    /* justify-content: start; */  
    /* justify-content: center; */  
    /* justify-content: end; */  
    /* justify-content: space-between; */  
    /* justify-content: space-around; */  
    /* justify-content: space-evenly; */  
}
```

place-content

align-content와 justify-content를 같이 쓸 수 있는 단축 속성이예요.

align-content, justify-content의 순서로 작성하고, 하나의 값만 쓰면 두 속성 모두에 적용됩니다.

```
.grid-container {  
    place-content: space-between center;  
}
```



HTML (Hyper Text Markup Language)

페이지의 제목, 문단, 표, 이미지, 동영상 등 웹의 구조

CSS (Cascading Style Sheets)

실제 화면에 표시되는 방법(색상, 크기, 폰트, 레이아웃 등)을 지정해 콘텐츠를 꾸며주는 시각적인 표현(정적)

JS (JavaScript)

콘텐츠를 바꾸고 움직이는 등 페이지를 동작시키는 동적 처리

#### 웹 표준

웹 표준(Web Standard)이란 '웹에서 사용되는 표준 기술이나 규칙'을 의미,  
W3C의 표준화 제정 단계의 '권고안(REC)'에 해당하는 기술



#### 크로스 브라우징

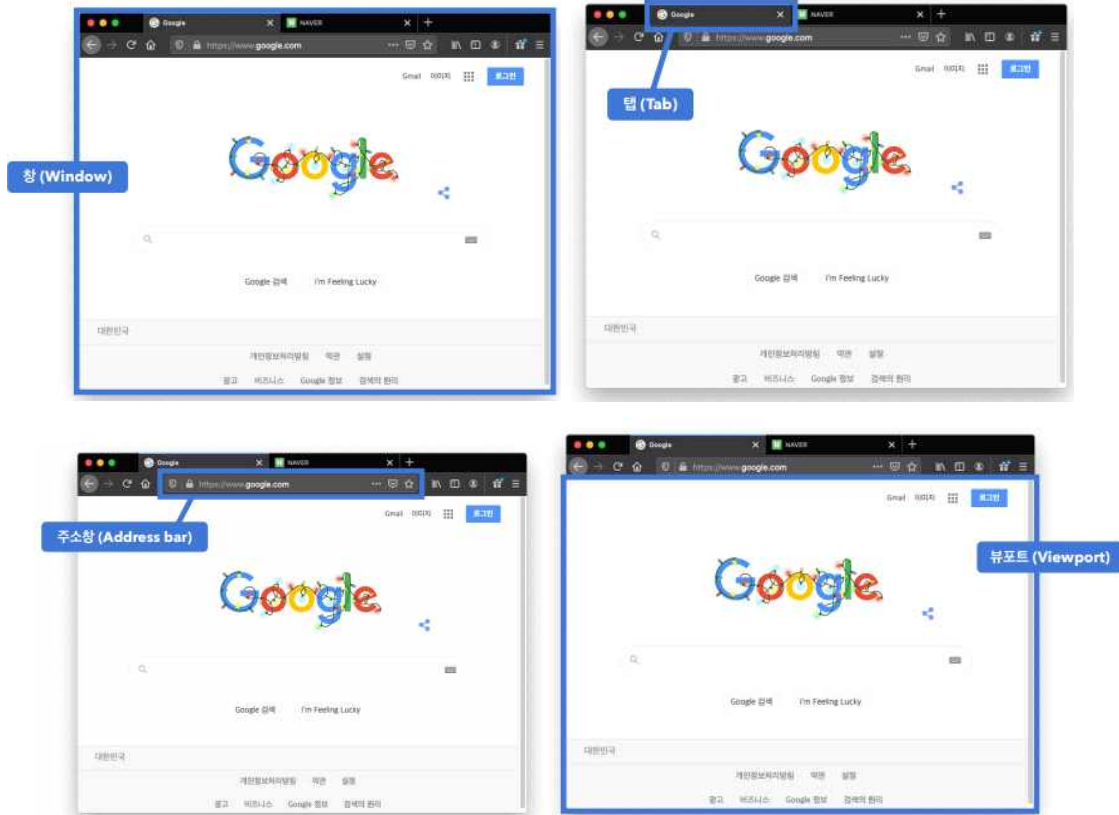
크로스 브라우징(Cross Browsing)이란 조금은 다르게 구동되는 여러 브라우저에서,  
동일한 사용자 경험(같은 화면, 같은 동작 등)을 줄 수 있도록 제작하는 기술, 방법.



#### 웹 접근성

웹 접근성이란 고령자, 장애인 같은 신체적, 환경적 조건에 제한이 있는  
사용자를 포함한,  
모든 사용자들이 동등하게 접근할 수 있는 웹 콘텐츠를 제작하는 방법



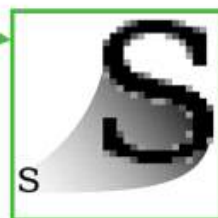


## 웹 이미지

픽셀이 모여 만들어진 정보의 집합,  
래스터(Raster) 이미지라고도 부름.

비트맵(Bitmap)과 벡터(Vector)

점, 선, 면의 위치(좌표), 색상 등  
수학적 정보의 형태(Shape)로  
이루어진 이미지.



**Raster**  
.jpeg .gif .png



**Vector**  
.svg



JPG(Joint Photographic coding Experts Group)는 Full-color와 Gray-scale의 압축을 위해 만들어졌으며, 압축률이 훌륭해 사진이나 예술 분야에서 많이 사용.

## **JPG(JPEG)**



**손실 압축**  
표현 색상도(24비트, 약 1600만 색상)가 뛰어남  
이미지의 품질과 용량을 쉽게 조절 가능  
가장 널리 쓰이는 이미지 포맷

PNG(Portable Network Graphics)는 Gif의 대체 포맷으로 개발됨.

## **PNG**



**비손실 압축**  
8비트(256 색상) / 24비트(약 1600만 색상) 컬러 이미지 처리  
Alpha Channel 지원(투명도)  
W3C 권장 포맷

GIF(Graphics Interchange Format)는 이미지 파일 내에 이미지 및 문자열 같은 정보들을 저장.



비손실 압축  
여러 장의 이미지를 한 개의 파일에 담을 수 있음  
(움짤, 애니메이션)  
8비트 색상만 지원(다양한 색상 표현에는 적합하지 않음)

JPG, PNG, GIF를 모두 대체할 수 있는 구글이 개발한 이미지 포맷.

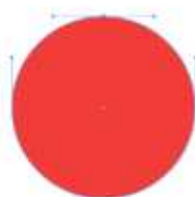


완벽한 손실/비손실 압축 지원  
GIF 같은 애니메이션 지원  
Alpha Channel 지원(손실, 비손실 모두)

IE 지원 불가

IE	Edge	Firefox	Chrome	Safari	Opera	IOS Safari	Opera mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Bravo Browser	4x4 Browser
4-8			4-8		10.1											
9-22			9-22		11.3			2.1-3								
23-31			23-31		23.1-18			4-4.1								
6-10	12-17	2-64	22-31	9-11.5	16-71	32-19.7		4.3-4.4.4	12-12.1				4-12.6			
11	18-86	65-81	32-86	9-11.5	16-71	32-19.7		4.3-4.4.4	12-12.1				4-12.6			
	87	84	87	14	72	14.2	49	81	91	87	81	12.12	11.0	10.4	7.12	2.5
	85-86	86-90	79													

SVG(Scalable Vector Graphics)는 마크업 언어(HTML/XML) 기반의 벡터 그래픽을 표현하는 포맷.



해상도의 영향에서 자유로움  
CSS와 JS로 제어 가능  
파일 및 코드 삽입 가능

## 특수 문자 용어

특수 문자의 영어/한글 이름과 키보드 위치

Backtick(백틱), Grave(그레이브) ~ Tilde(틸드, 물결표시)



! Exclamation mark (엑스클러메이션, 느낌표)



@ At sign(앳, 골뱅이)





# Sharp, Number sign(샵, 넘버, 우물 정)



\$ Dollar sign(달러)



% Percent sign (퍼센트)



^ Caret (캐럿)



& Ampersand (앰퍼샌드)



\* Asterisk (에스터리스크, 별표)



- Hyphen, Dash (하이픈, 대시, 마이너스)

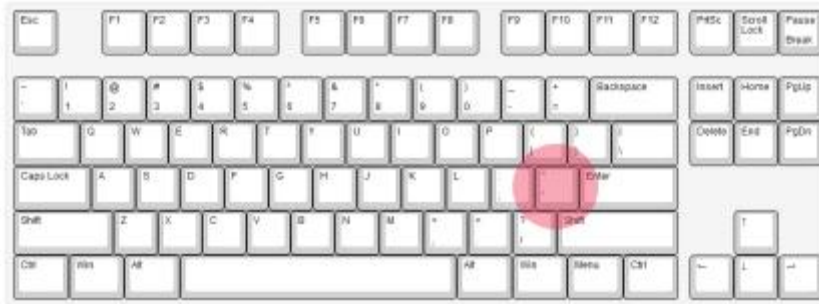
\_ Underscore, Low dash (언더스코어, 로대시, 밑줄 )



= Equals sign (이퀄, 동등)



" Quotation mark (쿼테이션, 큰 따옴표) ' Apostrophe (아포스트로피, 작은 따옴표)



: Colon (콜론) ; Semicolon (세미콜론)



, Comma (кома, 쉼표)



. Period, Dot (피리어드, 닷, 점, 마침표)





? Question mark (퀘스천, 물음표)



/ Slash (슬래시)



| Vertical bar (버티컬)      \ Backslash (백슬래시, 역 슬래시)



( ) Parenthesis (퍼렌서시스, 소괄호, 괄호)



{ } Brace (브레이스, 중괄호)



[ ] Bracket (브래킷, 대괄호)



< > Angle Bracket (앵글 브래킷, 꺾쇠괄호)



## HTML5 기본 구성

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
</body>
</html>
```

<!DOCTYPE html>

```
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

문서의 HTML 버전을 지정

HTML1  
HTML2  
HTML3  
HTML4  
XHTML  
**HTML5 (표준)**

DOCTYPE(DTD, Document Type Definition)은  
마크업 언어에서 문서 형식을 정의하며,  
웹 브라우저가 어떤 HTML 버전의 해석 방식으로  
페이지를 이해하면 되는지를 알려주는 용도.

<!DOCTYPE html>

<html>

```
<head>
</head>
<body>
</body>
```

</html>

문서의 전체 범위

HTML 문서가 어디에서 시작하고, 어디에서 끝나는지 알려주는 역할.

<!DOCTYPE html>

<html>

<head>

</head>

<body>

</body>

</html>

문서의 정보를 나타내는 범위

웹 브라우저가 해석해야 할  
웹 페이지의 제목, 설명, 사용할 파일 위치, 스타일(CSS) 같은,  
웹페이지의 보이지 않는 정보를 작성하는 범위.

```

<!DOCTYPE html>
<html>
  <head>      문자인코딩 방식
    <meta charset="UTF-8" />
    <meta name="author" content="HEROPY" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
  </head> 정보의 종류      정보의 값
  <body></body>
</html>

```

<meta />는 HTML 문서(웹페이지)의 제작자, 내용, 키워드 같은, 여러 정보를 검색엔진이나 브라우저에게 제공.

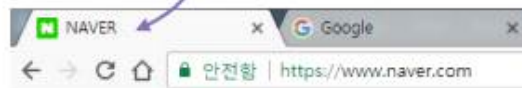
```

<!DOCTYPE html>
<html>
  <head>
    <title>Naver</title>
  </head>
  <body></body>
</html>

```

HTML 문서의 제목(title)을 정의.

웹 브라우저 탭에 표시됨!



```

<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
  </body>
</html>

```

문서의 구조를 나타내는 범위

사용자 화면을 통해 보여지는 로고, 헤더, 푸터, 내비게이션, 메뉴, 버튼, 이미지 같은, 웹페이지의 보여지는 구조를 작성하는 범위.

<html>, <head>,<body> 등과 같이 명령어의 형태가 '<>(각 괄호)'로 되어있는 것을 태그(tag)라 하며, <body> ~ </body>와 같이 태그의 시작(<>)과 종료(</>)를 하나의 요소(element)라고 한다. 또 이러한 요소들로 코딩한 것을 마크업(markup)이라고 한다.



첫줄에 DOCTYPE을 선언하면서 HTML문서의 형식을 브라우저에게 알려 줌으로써 태그/요소의 지원여부나 속성처리 기준을 통하여 브라우저가 HTML 문서의 내용을 올바르게 표시할 수 있도록 해준다. 웹표준 문서의 기본이 되므로 반드시 선언해 주어야 한다.  
<html>요소는 <head>요소와 <body> 요소로 구성되어 있으며 <head>요소 안에는 <meta>, <title>, <link>, <style>, <script>등의 요소들이 올수 있다.

요소명	설명
<meta>	html 문서의 설명, 키워드, 문서의 작성자 등을 지정한다.
<title>	html 문서의 제목을 지정한다.
<link>	외부의 CSS 파일을 html 문서에 연결할 때 사용한다.
<style>	html 문서 내에서 스타일(CSS)을 정의한다.
<script>	자바스크립트를 html 문서 내부에서 정의하거나 외부에서 자바스크립트 파일을 불러올 때 사용한다.

<html lang="ko">나 <meta charset="UTF-8">처럼 태그 내부에 부가적인 설정값을 선언하는 것을 속성(attribute)이라고 한다.  
lang속성은 HTML 문서에서 사용하는 언어를 말하여 lang="ko"는 한국어, lang="en"은 영어를 의미한다.  
<meta>태그의 charset 속성의 값 UTF-8을 '문자 인코딩(charset)'방식이라고 하는데 UTF-8의 경우 거의 대부분의 문자를 처리 할 수 있기 때문에 HTML5의 문자 인코딩 방식으로 주로 사용되고 있다.

## 마크업 기본 문법 규칙

1. 요소는 시작 태그와 종료 태그가 있어야 한다.

<code>&lt;p&gt;요소는 시작 태그와 종료 태그가 있어야 한다.&lt;/p&gt;</code>	✓
<code>&lt;p&gt;요소는 시작 태그와 종료 태그가 있어야 한다.</code>	✗

2. 요소는 제대로 중첩되어야 한다. 마지막에 정의한 태그를 가장 먼저 닫아야 한다.

<code>&lt;p&gt;중첩된 요소가 있을 때는 &lt;strong&gt;바르게&lt;/strong&gt; 표현해야 한다.&lt;/p&gt;</code>	✓
<code>&lt;p&gt;중첩된 요소가 있을 때는 &lt;strong&gt;바르게&lt;/p&gt; 표현해야 한다.&lt;/strong&gt;</code>	✗

3. img 태그에는 alt속성이 있어야 한다.

<code>&lt;img src="images/today.gif" alt="오늘"&gt;</code>	✓
<code>&lt;img src="images/today.gif"&gt;</code>	✗

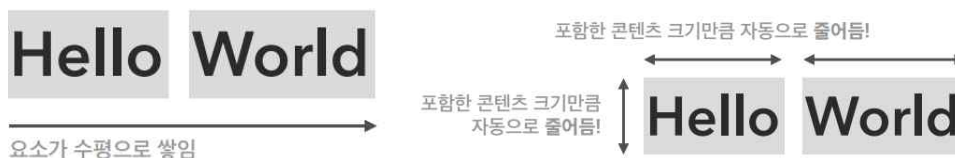
4. 주석처리방법 : 주석은 일반적으로 협업(공동작업)시 다음 작업자가 현재 작업한 내용을 쉽게 알아볼 수 있도록 도와주는 설명문으로 주석 내용은 브라우저 화면에 출력되지 않는다.

<code>&lt;!-- header --&gt;</code>	
<code>&lt;!-- 여기서부터 공지 사항입니다. --&gt;</code>	

## 글자와 상자

요소가 화면에 출력되는 특성, 크게 2가지로 구분.

- 1) 인라인(Inline) 요소 : 글자를 만들기 위한 요소들.



- 2) 블록(Block) 요소 : 상자(레이아웃)를 만들기 위한 요소들.



## 제목과 본문태그

- h1...h6 (Heading) : 제목으로 1부터 6까지의 크기
- p(Paragraph) : 본문 문단 생성
- br(Break) : 줄 바꿈
- hr(Horizontal Rule) : 수평 줄 삽입
- div(Division) : 영역 분할
- pre(Preformatted) : 형식화된 텍스트표시
- 특수문자 : &nbsp;(공백), &lt;(<), &gt;(>), &amp;(&)

## 글자 모양 태그의 종류

- b (Bold) : 굵은 글자
- strong : 볼드체
- i (Italic) : 기울어진 글자
- em (Emphasized) : 기울어진 글자
- small : 작은 글자
- sub (Subscript) : 아래 첨자
- sup (Superscript) : 위 첨자
- ins (Insert) : 밑줄 글자
- del (Delete) : 취소선글자
- mark (Marked) : 하이라이트된 글자

## HTML 인용 및 인용 요소 <blockquote>

<blockquote cite="출처를 남길 인용주소">

For 60 years, WWF has worked to help people and nature thrive. As the world's leading conservation organization, WWF works in nearly 100 countries. At every level, we collaborate with people around the world to develop and deliver innovative solutions that protect communities, wildlife, and the places in which they live.

</blockquote>

## 짧은 인용문의 경우 <q>

<p>WWF's goal is to: <q>Build a future where people live in harmony with nature.</q></p>

## 약어 <abbr>

<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>

## 문서의 작성자/소유자에 대한 연락처 정보 <address>

요소의 텍스트는 일반적으로 기울임 꼴로 렌더링

<address>

Written by John Doe.<br>

Visit us at:<br>

Example.com<br>

Box 564, Disneyland<br>

USA

</address>

## 하이퍼링크 (Hyperlink) 태그

- a (Anchor) : 하이퍼링크를 의미
- href(Hypertext Reference) : 파일의 경로(주소)
- target 속성은 연결된 문서를 열 위치를 지정한다.
  - \_self - 기본값. 클릭한 것과 동일한 창/탭에서 문서를 엽니다.
  - \_blank - 새 창이나 탭에서 문서를 엽니다.
  - \_parent - 상위 프레임에서 문서를 엽니다.
  - \_top - 창의 전체 본문에서 문서를 엽니다.
- URL (Uniform Resource Locator)
  - 절대주소 : 경로의 정확한 주소를 표현
  - 상대주소 : html 파일과의 상대적인 위치를 표현

```
<a href="url">link text</a>
```

ex) index.html 파일을 기준으로 a.html 파일을 불러오려고 하는 경우

- 절대주소
    - href="http://localhost/html/a.html"
    - http:// 또는 / (root 경로) 부터 정확한 파일 위치를 표기
  - 상대주소
    - href="/html/a.html"
    - 현재 파일 위치를 기준으로 파일 위치를 표기
    - / : 루트, ./ : 현재 위치(생략가능), ../ : 현재 위치의 상단 폴더
- ex) index.php가 C:\index\wa에 위치한다면,  
여기서 / 는 C:    ./ 는 a    ../ 는 index라는 것.

## HTML 링크 색상

기본적으로 링크는 모든 브라우저에 다음과 같이 표시됩니다.

방문하지 않은 링크에 밑줄이 그어지고 파란색으로 표시됩니다.

방문한 링크에 밑줄이 그어지고 자주색이 표시됩니다.

활성 링크에 밑줄이 그어지고 빨간색으로 표시됩니다.

```
<style>
a:link {
    color: green;
}
a:visited {
    color: pink;
}

a:hover {
    color: red;
}

a:active {
    color: yellow;
}
</style>
```



HTML 링크

이미지를 링크로 사용

```
<a href="default.asp">  
  
</a>
```

이메일 주소에 연결

```
<a href="mailto:someone@example.com">Send email</a>
```

링크 제목

title 속성은 요소에 대한 추가 정보를 지정합니다. 이 정보는 마우스를 요소 위로 이동할 때 도구 설명 텍스트로 표시되는 경우가 가장 많습니다.

```
<a href="이동할 웹페이지 주소" title="Go to HTML section">Visit our HTML Tutorial</a>
```

페이지 내부에서 이동하는 책갈피 기능을 함

```
<a href = "#point 1">1번 단락</a><br />
```

```
<a href = "#point 2">2번 단락</a><br />
```

```
<p><a id = "point 1">1번 단락</a></p>
```

```
<p><a id = "point 2">2번 단락</a></p>
```

이미지 (img) 태그

- img (Image) : 이미지태그
- src (Source) : 이미지의 경로
- alt (Alternate Text) : 이미지가 없을 때 나오는 글자
- width : 이미지의 가로 크기
- height : 이미지의 세로 크기

리스트 태그의 종류

- ol : 순서가 있는 리스트
- ul : 순서가 없는 리스트
- dl : 정의 리스트

```
<ul>
```

```
<li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Milk</li>
```

```
</ul>
```

```
<dl>-->설명 목록 정의
```

```
<dt>Coffee</dt>-->설명 목록에서 용어를 정의
```

```
<dd>- black hot drink</dd>-->설명 목록에서 용어를 설명
```

```
<dt>Milk</dt>
```

```
<dd>- white cold drink</dd>
```

```
</dl>
```

블록(block) 요소와 인라인(inline) 요소

요소 유형	특징
블록(block) 요소	<ol style="list-style-type: none"> <li>1. 블록 요소는 줄바꿈이 일어난다.</li> <li>2. 블록 요소는 텍스트(문자)와 인라인 요소를 자식 요소로 포함할 수 있다.</li> <li>3. 블록 요소 중에는 블록 레벨 요소를 자식요소로 포함 할 수 있는 요소와 포함할 수 없는 요소가 있다.</li> <li>4. 블록 요소는 가로, 세로 너비 지정할 수 있다.</li> </ol>
인라인(inline) 요소	<ol style="list-style-type: none"> <li>1. 인라인 요소는 줄바꿈이 일어나지 않는다.</li> <li>2. 인라인 요소는 텍스트(문자)와 인라인 요소를 자식 요소로 포함할 수 있다.</li> <li>3. 인라인 요소는 블록 레벨 요소를 자식 요소로 포함할 수 없다.</li> <li>4. 인라인 요소는 가로, 세로 너비 지정할 수 없다.</li> </ol>

블록(block) 요소

요소유형	태그명	태그의 의미 및 특징
블록(block) 요소	<code>&lt;h1&gt;&lt;/h1&gt;</code> <code>&lt;h2&gt;&lt;/h2&gt;</code> <code>&lt;h3&gt;&lt;/h3&gt;</code> <code>&lt;h4&gt;&lt;/h4&gt;</code> <code>&lt;h5&gt;&lt;/h5&gt;</code> <code>&lt;h6&gt;&lt;/h6&gt;</code>	1. HTML 문서에서 제목(heading)을 정의 할 때 사용하는 태그이다. 2. <code>&lt;h1&gt;</code> ~ <code>&lt;h6&gt;</code> 태그는 텍스트와 인라인 요소를 포함할 수 있지만 블록 레벨 요소는 포함할 수 없다.
	<code>&lt;p&gt;&lt;/p&gt;</code>	1.HTML 문서에서 단락(paragraph)을 정의 할 때 사용하는 태그이다. 2. 텍스트와 인라인 요소를 포함할 수 있지만 블록 레벨 요소는 포함할 수 없다. 3. 텍스트의 행을 바꾸려면 <code>&lt;br&gt;</code> 태그를 사용한다.
	<code>&lt;ol&gt;&lt;/ol&gt;</code>	1. Order List의 약자로 순서 목록을 정의할 때 사용하는 태그이다. 2. 자식 요소로 반드시 <code>&lt;li&gt;</code> 태그를 정의해 주어야 하며 <code>&lt;li&gt;</code> 태그 이외의 다른 태그는 자식 요소로 올 수 없다.
	<code>&lt;ul&gt;&lt;/ul&gt;</code>	1. Unorder List의 약자로 비순서 목록을 정의할 때 사용하는 태그이다. 2. 자식 요소로 반드시 <code>&lt;li&gt;</code> 태그를 정의해 주어야 하며 <code>&lt;li&gt;</code> 태그 이외의 다른 태그는 자식 요소로 올 수 없다.
	<code>&lt;li&gt;&lt;/li&gt;</code>	1. List Items로 항목을 정의할 때 사용하는 태그로 <code>&lt;ol&gt;</code> , <code>&lt;ul&gt;</code> 태그의 자식요소이다. 2. 텍스트, 인라인 요소, 블록 레벨 요소를 모두 포함할 수 있다.

블록(block) 요소

요소유형	태그명	태그의 의미 및 특징
블록(block) 요소	<code>&lt;dl&gt; &lt;/dl&gt;</code>	<ol style="list-style-type: none"> <li>1. Description List의 약자로 용어에 대한 설명 목록을 정의할 때 사용하는 태그이다.</li> <li>2. 자식 요소로 반드시 <code>&lt;dt&gt;</code>태그와 <code>&lt;dd&gt;</code> 태그를 정의해 주어야 하며 <code>&lt;dl&gt;-&gt;&lt;dt&gt;-&gt;&lt;dd&gt;</code>순으로 마크업해야 한다. <code>&lt;dl&gt;</code>태그는 <code>&lt;dt&gt;</code>,<code>&lt;dd&gt;</code> 이외의 태그는 자식요소로 올 수 없지만 <code>&lt;dt&gt;</code>,<code>&lt;dd&gt;</code>을 한 쌍으로 묶는 <code>&lt;div&gt;</code> 태그는 예외적으로 올 수 있다.</li> </ol>
	<code>&lt;dt&gt; &lt;/dt&gt;</code>	<ol style="list-style-type: none"> <li>1. Description Term의 약자로 설명 목록의 용어를 정의한다.</li> <li>2. 텍스트, 인라인 요소, 블록 레벨 요소를 모두 포함할 수 있다.</li> </ol>
	<code>&lt;dd&gt; &lt;/dd&gt;</code>	<ol style="list-style-type: none"> <li>1. Description Details의 약자로 용어에 대한 세부 설명을 정의한다.</li> <li>2. 텍스트, 인라인 요소, 블록 레벨 요소를 모두 포함할 수 있다.</li> </ol>
	<code>&lt;div&gt; &lt;/div&gt;</code>	<ol style="list-style-type: none"> <li>1. 요소들을 그룹으로 정의하는 태그이다.</li> <li>2. 텍스트, 인라인 요소, 블록 레벨 요소를 모두 포함할 수 있다.</li> </ol>
	<code>&lt;header&gt; &lt;/header&gt;</code>	<ol style="list-style-type: none"> <li>1. HTML 문서의 헤더 영역을 의미하는 태그로 제목이나 내비게이션, 검색 등의 내용들을 포함할 수 있다.</li> <li>2. 텍스트, 인라인 요소, 블록 레벨 요소들 포함할 수 있지만 <code>&lt;header&gt;</code>, <code>&lt;footer&gt;</code>태그는 포함할 수 없다.</li> </ol>
	<code>&lt;nav&gt; &lt;/nav&gt;</code>	<ol style="list-style-type: none"> <li>1. HTML 문서의 메인 메뉴나 목차 등을 정의해 주는 태그이다.</li> <li>2. 텍스트, 인라인 요소, 블록 레벨 요소를 포함할 수 있다.</li> </ol>
	<code>&lt;section&gt; &lt;/section&gt;</code>	<ol style="list-style-type: none"> <li>1. HTML 문서에서 맥락이 같은 요소들을 주제별로 그룹화해주는 태그이며 섹션 주제에 대한 제목요소(<code>&lt;h2&gt;~&lt;h6&gt;</code>)를 포함하는 것이 좋다.</li> <li>2. 텍스트, 인라인 요소, 블록 레벨 요소를 포함할 수 있다.</li> </ol>

## 블록(block) 요소

요소유형	태그명	태그의 의미 및 특징
블록(block) 요소	<code>&lt;article&gt;</code> <code>&lt;/article&gt;</code>	<ol style="list-style-type: none"> <li>1. HTML 문서 내에서 독립적으로 배포 또는 재사용이 가능한 게시물, 뉴스 기사, 블로그 포스팅 등을 의미하는 태그이며 제목요소(&lt;h2&gt;~&lt;h6&gt;)를 포함하는 것이 좋다.</li> <li>2. 텍스트, 인라인 요소, 블록 레벨 요소를 포함할 수 있다.</li> </ol>
	<code>&lt;section&gt;</code> <code>&lt;article&gt;</code> <code>&lt;div&gt;</code>	<ol style="list-style-type: none"> <li>1. 내용이 독립적이고 스스로 설 수 있는 내용이라면 article을 사용.</li> <li>2. 내용이 서로 관계가 있다면 section</li> <li>3. 의미적으로 관계가 없다면 div를 사용(div는 오직 내용을 묶는 역할)</li> </ol> <p>사용 예1.</p> <pre> &lt;article&gt;   사람   &lt;section&gt;똥똥한 사람&lt;/section&gt;   &lt;section&gt;마른 사람&lt;/section&gt; &lt;/article&gt; </pre> <p>사용 예2.</p> <pre> &lt;section&gt;   &lt;article&gt;블로그 글1&lt;/article&gt;   &lt;article&gt;블로그 글2&lt;/article&gt;   &lt;article&gt;블로그 글3&lt;/article&gt; &lt;/section&gt; </pre>
	<code>&lt;aside&gt;</code> <code>&lt;/aside&gt;</code>	<ol style="list-style-type: none"> <li>1. 메인 콘텐츠와 직접적으로 관련이 없는 영역을 의미하는 태그이며 HTML 문서의 오른쪽이나 왼쪽의 사이드 메뉴나 광고 등의 영역으로 사용된다.</li> <li>2. 텍스트, 인라인 요소, 블록 레벨 요소를 포함할 수 있다.</li> </ol>
	<code>&lt;footer&gt;</code> <code>&lt;/footer&gt;</code>	<ol style="list-style-type: none"> <li>1. HTML 문서의 푸터 영역을 의미하는 태그로 섹션 작성자나 저작권에 대한 정보, 관련된 문서의 링크를 포함할 수 있다.</li> <li>2. 텍스트, 인라인 요소, 블록 레벨 요소들 포함할 수 있지만 &lt;header&gt;, &lt;footer&gt;태그는 포함할 수 없다.</li> </ol>

블록(block) 요소

요소유형	태그명	태그의 의미 및 특징
블록(block) 요소	<code>&lt;table&gt;</code> <code>&lt;/table&gt;</code>	<ol style="list-style-type: none"> <li>1. 표를 정의할 때 사용하는 태그이며, 행(Row)과 열(Column)의 2차원 정보로 구성되어 있다.</li> <li>2. 자식 요소로 반드시 <code>&lt;tr&gt;</code>태그를 정의해 주어야 하며 <code>&lt;table&gt;-&gt;&lt;tr&gt;-&gt;&lt;td&gt;</code>순으로 마크업해야 한다.</li> <li>3. <code>&lt;tr&gt;</code>태그는 <code>&lt;table&gt;</code>태그의 자식 요소로 표에서 행(Row)을 정의할 때 사용하는 태그이다.</li> <li>4. <code>&lt;td&gt;</code>태그는 <code>&lt;tr&gt;</code>태그의 자식 요소로 표에서 열(Column)을 정의할 때 사용하는 태그이다.</li> <li>5. <code>&lt;th&gt;</code>태그는 제목 역할을 하는 셀을 정의할 때 사용하는 태그이다.</li> <li>6. 표의 내용을 기중으로 제목 영역<code>&lt;thead&gt;</code>, 본문 영역<code>&lt;tbody&gt;</code>, 열의 요약 영역 <code>&lt;tfoot&gt;</code>으로 구분하여 정의할 수 있다.</li> <li>7. 표는 <code>&lt;tr&gt;</code>태그처럼 <code>&lt;td&gt;</code>나 <code>&lt;th&gt;</code> 태그들을 가로로 묶는 행(Row) 그룹은 있지만 세로로 묶는 열(Column) 그룹은 없다</li> </ol> <p>※ border-collapse: collapse  <code>&lt;td&gt;</code>요소들의 위, 아래, 좌, 우 테두리선이 서로 겹쳤을 때 하나의 테두리선으로 표현해 준다.</p>
	<code>&lt;form&gt;</code> <code>&lt;/form&gt;</code>	<ol style="list-style-type: none"> <li>1. 온라인 서식(검색, 로그인, 회원가입 등)에서 입력한 값들을 처리하는 프로그램으로 전송할 때 사용하는 태그이다.</li> <li>2. 텍스트, 인라인 요소, 블록 레벨 요소를 포함할 수 있다. 일반적으로 사용자 입력에 관련된 태그들을 포함한다.</li> <li>3. form 태그에는 action 속성과 method 속성이 있다. action 속성은 서식에서 작성한 값을 처리하는 프로그램의 주소(URL)를 지정하며, method속성은 서식의 값들을 서버 프로그램에 어떤 방식으로 전달할 것인지를 정의한다. 전달 방식에는 get방식과 post방식이 있다</li> </ol>
	<code>&lt;fieldset&gt;</code> <code>&lt;/fieldset&gt;</code>	<ol style="list-style-type: none"> <li>1. 양식 요소들을 그룹화할 때 사용하는 태그이다.</li> <li>2. 텍스트, 인라인 요소, 블록 레벨 요소를 포함할 수 있다.</li> <li>3. 그룹화 된 서식의 제목을 정의 할 수 있는 <code>&lt;legend&gt;</code> 태그를 자식요소로 포함한다.</li> </ol>

## 인라인(inline) 요소

요소유형	태그명	태그의 의미 및 특징
인라인(inline) 요소	<code>&lt;span&gt;</code> <code>&lt;/span&gt;</code>	1. 인라인 요소들을 그룹으로 정의하는 태그이다. 2. 텍스트, 인라인 요소를 포함할 수 있다.
	<code>&lt;a&gt;</code> <code>&lt;/a&gt;</code>	1. 다른 HTML 문서의 이동(외부링크)이나 동일한 HTML 문서 내에서 이동(내부링크), 이메일 주소에 링크를 지정할 수 있다. 2. href 속성은 링크의 목적지를 지정할 수 있다. 3. HTML5에서는 블록 레벨 요소도 포함할 수 있게 되었지만 <code>&lt;input&gt;</code> , <code>&lt;button&gt;</code> , 다른 link( <code>&lt;a&gt;</code> )등의 요소들은 포함할 수 없다.
	<code>&lt;img&gt;</code>	1. HTML 문서에 이미지를 삽입할 때 사용하는 태그이다. 2. <code>&lt;img&gt;</code> 태그의 필수 속성인 src 속성은 불러올 이미지의 경로를 지정한다.
	<code>&lt;em&gt;</code> <code>&lt;/em&gt;</code>	강조하고 싶은 텍스트를 정의하기 위한 태그이며 기울임체로 표시된다.
	<code>&lt;strong&gt;</code> <code>&lt;/strong&gt;</code>	중요한 텍스트를 정의하기 위한 태그이며 굵은체로 표시된다.
	<code>&lt;mark&gt;</code> <code>&lt;/mark&gt;</code>	주의깊게 볼 텍스트 부분을 강조하기 위한 태그이며 노란색으로 표시된다.
	<code>&lt;b&gt;</code> <code>&lt;/b&gt;</code>	의미를 가지고 있지 않으며 단순히 텍스트를 굵은체로 표시한다
	<code>&lt;small&gt;</code> <code>&lt;/small&gt;</code>	주의 사항, 법적 제한, 저작권 등을 정의하기 위한 태그이며 작은 텍스트로 표시된다.
	<code>&lt;sub&gt;</code> <code>&lt;/sub&gt;</code>	아래첨자 텍스트를 정의하기 위한 태그이다.
	<code>&lt;sup&gt;</code> <code>&lt;/sup&gt;</code>	위첨자 텍스트를 정의하기 위한 태그이다.

## 인라인(inline) 요소

요소유형	태그명	태그의 의미 및 특징
인라인(inline) 요소	<code>&lt;address&gt;</code> <code>&lt;/address&gt;</code>	1. HTML 문서의 소유자나 조직에 대한 연락처 정보를 정의할 때 사용하는 태그이다. 2. 텍스트와 인라인 요소를 포함할 수 있고 <code>&lt;h1&gt;~&lt;h6&gt;</code> <code>&lt;section&gt;</code> <code>&lt;article&gt;</code> <code>&lt;aside&gt;</code> <code>&lt;nav&gt;</code> <code>&lt;header&gt;</code> <code>&lt;footer&gt;</code> 와 같은 일부 블록 레벨 요소는 포함할 수 없다. 3. <code>&lt;address&gt;</code> 태그는 보통 <code>&lt;footer&gt;</code> 태그 내에서 많이 정의한다.
	<code>&lt;input&gt;</code> <code>&lt;/input&gt;</code>	1. 사용자로부터 데이터 값을 입력받기 위한 태그이다. 2. type 속성값에 따라 다양한 폼 컨트롤을 생성할 수 있다. 3. 공통 속성으로 name, value 속성이 있다.
	<code>&lt;textarea&gt;</code> <code>&lt;/textarea&gt;</code>	1. 여러 줄 텍스트 입력 상자 태그이다. 2. rows 속성과 cols 속성으로 텍스트 입력 상자의 행과 열 수를 지정할 수 있다.
	<code>&lt;select&gt;</code> <code>&lt;/select&gt;</code>	1. 선택 목록 상자 태그이다. 2. <code>&lt;option&gt;</code> 태그 외에 다른 태그는 자식요소로 올 수 없다. 3. <code>&lt;option&gt;</code> 태그에 selected 속성이 정의 되면 웹 페이지 실행 시 해당 <code>&lt;option&gt;</code> 태그가 기본 목록으로 표시된다.
	<code>&lt;label&gt;</code> <code>&lt;/label&gt;</code>	1. 사용자 입력 태그 <code>&lt;input&gt;</code> , <code>&lt;textarea&gt;</code> , <code>&lt;select&gt;</code> 들에 대한 제목을 정의하는 태그이다. 2. <code>&lt;label&gt;</code> 태그를 사용자 입력 태그의 제목으로 연결하기 위해서는 <code>&lt;label&gt;</code> 태그의 for 속성 값과 사용자 입력 태그의 id 속성 값을 동일하게 정의해야 한다. <code>&lt;label for="id"&gt;&lt;/label&gt; &lt;input type="text" id="id"&gt;</code>



## Input Type 속성

type 속성 값	설명
text	<ul style="list-style-type: none"> <li>한 줄 글 입력 상자</li> <li>maxlength 속성으로 최대 글자 수를 제한 할 수 있다.</li> </ul>
password	<ul style="list-style-type: none"> <li>비밀번호 입력 상자</li> <li>입력된 내용은 '*'로 표시된다.</li> </ul>
search	<ul style="list-style-type: none"> <li>검색 입력 상자</li> </ul>
email	<ul style="list-style-type: none"> <li>이메일 입력 상자</li> </ul>
tel	<ul style="list-style-type: none"> <li>전화번호 입력 상자</li> </ul>
color	<ul style="list-style-type: none"> <li>컬러 입력 상자</li> </ul>
number	<ul style="list-style-type: none"> <li>숫자 입력 상자</li> <li>숫자의 최소값, 최대값을 min, max 속성으로 지정할 수 있다.</li> </ul>
range	<ul style="list-style-type: none"> <li>슬라이드바</li> <li>최소값, 최대값을 min, max속성으로 지정할 수 있다.</li> </ul>
radio	<ul style="list-style-type: none"> <li>라디오 버튼</li> <li>여러개 radio 중 하나만 선택 가능하다.(요소의 name 속성 값이 같아야 한다)</li> <li>checked 속성을 정의하면 웹 페이지 실행시 해당 radio가 기본으로 표시된다.</li> </ul>
checkbox	<ul style="list-style-type: none"> <li>체크박스</li> <li>다중 선택이 가능하다</li> <li>checked 속성을 정의하면 웹 페이지 실행시 해당 checkbox가 기본으로 표시된다.</li> </ul>
submit	<ul style="list-style-type: none"> <li>전송 버튼</li> <li>value 속성으로 버튼에 표시되는 텍스트를 지정한다.</li> </ul>
reset	<ul style="list-style-type: none"> <li>초기화 버튼</li> <li>value 속성으로 버튼에 표시되는 텍스트를 지정한다.</li> </ul>
button	<ul style="list-style-type: none"> <li>범용 버튼</li> <li>value 속성으로 버튼에 표시되는 텍스트를 지정한다.</li> </ul>
image	<ul style="list-style-type: none"> <li>이미지 버튼</li> <li>alt 속성을 반드시 정의해야 한다.</li> </ul>
file	<ul style="list-style-type: none"> <li>첨부 파일</li> </ul>
hidden	<ul style="list-style-type: none"> <li>사용자에게 노출할 필요가 없는 데이터를 지정한다.</li> <li>화면에 표시되지 않는다.</li> </ul>
required	<ul style="list-style-type: none"> <li>required 속성이 적용된 입력 요소에 내용이 비어 있을 경우 경고 메시지를 표시해 주는 속성이다.</li> </ul>
placeholder	<ul style="list-style-type: none"> <li>입력 내용에 대한 힌트를 제공한다. 내용을 입력하지 않고 비어 있는 상태로 포커스를 이동하면 초기 힌트 값으로 표시해 주는 속성이다.</li> </ul>

## 미디어 요소

태그명	태그의 의미 및 특징
<audio> </audio>	1. HTML 문서에 음악과 같은 오디오를 삽입할 때 사용하는 태그이다. 2. mp3, wav, ogg 3가지 파일 형식이 있다.
속성명	설명
src	• 오디오 파일의 주소이다.
autoplay	• HTML 문서가 로드될 때 오디오를 자동으로 실행한다.
muted	• 오디오 음소거를 한다.
loop	• 오디오를 반복 재생한다.
controls	• 오디오 컨트롤을 표시한다.

태그명	태그의 의미 및 특징
<video> </video>	1. HTML 문서에 동영상을 삽입할 때 사용하는 태그이다. 2. mp4, webm, ogg 3가지 파일 형식이 있다.
속성명	설명
src	• 영상 파일의 주소이다.
autoplay	• HTML 문서가 로드될 때 영상을 자동으로 실행한다.
muted	• 영상의 음소거를 한다.
loop	• 영상을 반복 재생한다.
controls	• 영상 컨트롤을 표시한다.
width	• 영상의 가로 크기를 지정한다.
height	• 영상의 세로 크기를 지정한다.

※ 하나의 이미지에 여러 개의 링크가 적용 되어야 하는 경우 <map> 태그를 사용하며 링크의 영역을 사각형, 원, 다각형으로 나타낼수 있다.

```

<map name="맵이름">
  <area shape="rect" coords="맵좌표" href="url" alt="대체텍스트">
  <area shape="circle" coords="맵좌표" href="url" alt="대체텍스트">
  <area shape="poly" coords="맵좌표" href="url" alt="대체텍스트">
</map>
```

shape의 맵좌표 구하는 방법

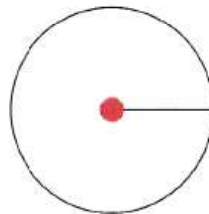
■ rect(사각형)

좌측상단 꼭짓점 x, y 좌표값과 우측하단 x, y 좌표값



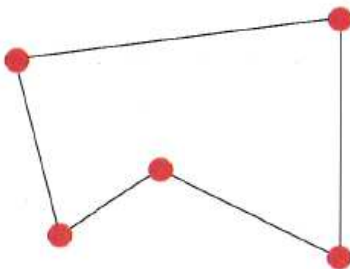
■ circle(원)

원의 중심점 x, y 좌표값과 원의 반지름값



■ poly(다각형)

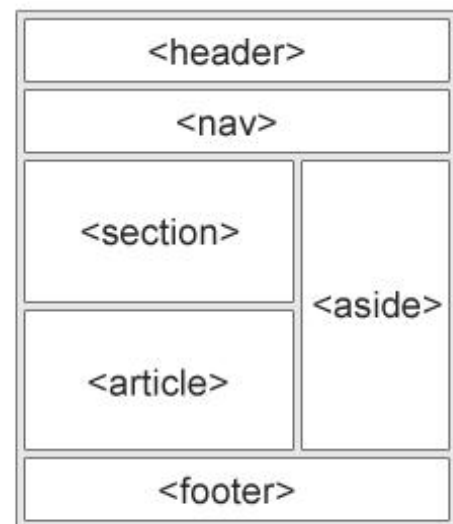
각 꼭짓점의 x, y 좌표값



Typical HTML4	Typical HTML5
<div id="header">	<header>
<div id="menu">	<nav>
<div id="content">	<section>
<div class="article">	<article>
<div id="footer">	<footer>

HTML5 offers new semantic elements to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



## CSS

### 1. css 기본문법

CSS(Cascading Style Sheets)를 정의하는 방법으로 내부 스타일 시트, 외부 스타일 시트, 인라인 스타일시트 등이 있다.

css에서의 주석은 /\* \*/로 표현한다.

#### 내부 스타일시트

css를 HTML 문서 안에 <style> 태그로 기재하여 HTML 문서 안에 css가 함께 저장되도록 하는 형태이다.

#### 외부 스타일시트

스타일 속성들을 따로 저장하여 HTML 문서에 파일명으로 연결할 수 있다. 이때 스타일은 \*.css 확장자를 가진 별도의 파일로 저장하고 HTML에서는 <link> 태그를 이용하여 선언한다. css 파일의 상단에는 @charset "utf-8"; 등 HTML 문서에서 지정한 것과 동일한 문자셋을 지정한다.

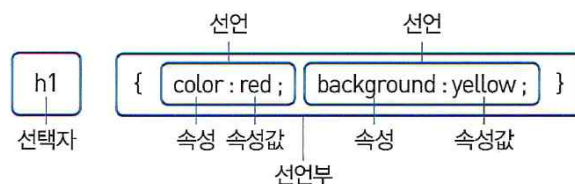
내부 스타일시트는 그 스타일이 기술된 페이지에만 적용되는 반면, 외부 스타일시트는 그 스타일을 선언한 모든 HTML페이지에 적용된다.

#### 인라인 스타일

인라인 스타일은 태그에 직접 CSS를 정의해주는 기법으로, 해당 요소에 style=" " 형식으로 기술한다. 인라인 스타일은 내부스타일이나 외부스타일에서 기술된 속성보다 우선으로 적용되므로 우선순위가 절대적으로 높아야 할 경우에 사용할 수 있으나, 스타일을 공통 CSS로 수정할 수 없고 일일이 HTML을 열어서 편집해야 하므로 일반적으로 많이 사용하지는 않는다.

### 2. 선택자(selector)

선택자란 CSS로 UI의 어느 부분을 디자인할지, 즉 표현할 대상이 되는 부분을 말한다.



{ } 안에 여러 속성을 지정할 수 있고, 각 속성 설정 간에는 ';'으로 구분해 주며 가로로 붙여 써도 되고 한 줄씩 따로 기술해도 무방하다.

- (1) type 선택자
  - html 문서의 태그 이름을 선택자로 사용할 수 있다.
- (2) id 선택자
  - HTML 요소에 id로 이름을 붙일 때에는 유일한 이름을 부여해야 하며, 다른 요소에 같은 id명을 또 주어서는 안된다. CSS에서는 id 선택자 앞에 '#'을 붙여야 한다.
  - id명이나 class명은 숫자로 시작할 수 없으며, 주로 영문자로 시작한다.
- (3) class 선택자
  - HTML 문서의 여러 요소 중 같은 이름을 갖는 요소들만 모아 따로 속성을 부여할 수 있다.
  - HTML 요소에 같은 이름을 부여할 때에는 class로 이름을 붙인다. 즉 여러개의 요소에 같은 class 명을 부여할 수 있다. CSS에서는 class 선택자 앞에 '.'을 붙여야 한다.
- (4) 전체 선택자
  - 전체 선택자는 페이지의 모든 요소를 가리키는 선택자로서 '\*'로 표시한다.
- (5) 하위 선택자
  - 하위 선택자는 요소 내부에 있는 모든 해당 요소를 가리키며, 선택자 사이를 공백으로 분리한다.
- (6) 자식 선택자
  - 자식 선택자는 요소 내부에 있는 해당 요소를 가리키지만, 하위 요소의 하위 요소는 가리키지 않으며 선택자 사이를 '>'으로 분리한다.
- (7) 인접 선택자
  - 인접 선택자는 현재 요소의 바로 뒤에 나오는 요소만을 가리키는 선택자로, 선택자 사이를 '+'로 분리한다.
- (8) 형제 선택자
  - 형제 선택자는 현재 요소와 같은 계층에 있는 요소만을 선택할 수 있으며 '~'로 구분한다.
- (9) 그룹 선택자
  - 그룹 선택자는 여러 선택자들을 ','로 구분하여 함께 묶어 속성을 부여하는 것이다.
- (10) 속성 선택자
  - HTML 요소의 속성 유무 또는 속성값을 중괄호 [ ] 안에 넣어 선택자로 사용할 수 있다.

속성값	설명
h1[class]	class명을 가진 h1요소
img[alt]	alt속성을 가진 img요소
p[class="abc"]	class명이 유일하게 'abc'인 p요소
p[class~="abc"]	class명이 유일하게 'abc'이거나 여러 개의 class명 중 하나가 'abc'인 p요소
p[class="abc"]	class명이 'abc'이거나 'abc'로 시작하는 p요소
p[class^="abc"]	class명이 'abc'로 시작하는 p요소
p[class\$="abc"]	class명이 'abc'로 끝나는 p요소
p[class*="abc"]	class명이 'abc'가 포함되어 있는 p요소
a[href^="mailto"]	href 속성값이 'mailto'로 시작하는 a요소

(11) 가상클래스 선택자

- 가상클래스 선택자란 링크가 걸린 문자에 스타일을 부여하는 것이다.

속성 값	설명
a:link	링크가 걸린 문자에 속성을 부여하는 선택자
a:visited	링크를 클릭하여 해당 페이지에 갔다가 돌아온 경우의 속성을 부여하는 선택자
a:hover	링크가 걸린 문자에 마우스가 닿았을 경우의 속성을 부여하는 선택자
a:active	링크 걸린 글자가 활성화되었을 경우의 속성을 부여하는 선택자
a:focus	링크 걸린 글자에 포커스가 생길 경우의 속성을 부여하는 선택자

(12) 가상요소 선택자

- 가상요소 선택자는 요소에 id명이나 class명을 부여하지 않고도 위치를 찾아서 선택할 수 있는 선택자이다.

속성 값	설명
:first-letter	요소의 첫글자
:first-line	요소의 첫 줄
:first-child	같은 요소 중 첫 번째 요소
:last-child	같은 요소 중 마지막 요소
:nth-child(n)	같은 요소중 n번째 요소
:before	요소 안 맨 앞에 배치될 요소(마크업에는 없는 가상 요소)
:after	요소 안 맨 뒤에 배치될 요소(마크업에는 없는 가상 요소)

요소에 직접 style=" " 형식으로 CSS를 기술하는 인라인 스타일(inline style)은 위의 모든 선택자보다 우선으로 실행된다. 지금까지 나온 방법보다 최우선으로 CSS를 적용해야 할 경우에는 선택자에 속성을 부여하는 마지막에 "!important"를 붙이면 된다.

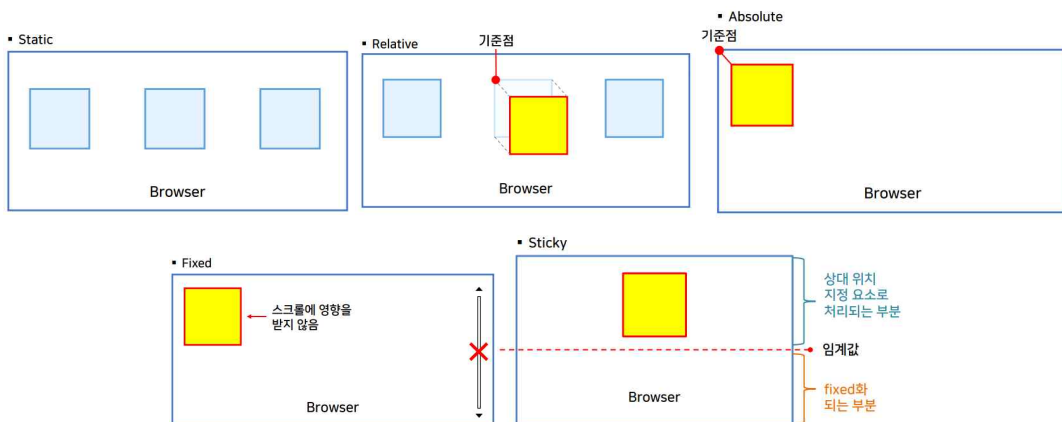
```
P { color: red !important; }
```

Type 선택자 < class 선택자 < id 선택자 < inline style < !important

## 레이아웃

position : 레이아웃을 배치하거나 요소 위치를 지정하는 중요한 속성

속성 값	속성 설명
static	요소를 이동하거나 겹칠 수 없는 원래 그대로의 상태
relative	left, top, right, bottom 속성으로 위치 지정할 수 있음 position: absolute로 지정된 요소의 부모 요소 역할을 할수도 있음
absolute	원래 위치에서 따로 떼어내 독립적으로 새로운 위치를 설정함. left, top, right, bottom 속성으로 위치 지정할 수 있음 부모 박스를 기준으로 위치를 지정함
fixed	요소의 위치를 screen 기준으로 지정함 left, top, right, bottom 속성으로 위치 지정할 수 있음
sticky	position의 계산값이 sticky (끈적끈적한)인 요소 평소에는 상대 위치 지정 요소로 처리하지만, 자신의 플로우 루트에서 지정한 임계값을 넘으면 마치 fixed처럼 화면에 고정되는 요소



z-index : 겹쳐 있는 요소들의 계층을 변경할 때 사용하는 속성

float : block 요소는 원래 세로로만 나열되지만 float을 이용하여 가로로 배치할 수 있다.

속성값	설명
left	요소가 왼쪽으로 이동되도록 지정
right	요소가 오른쪽으로 이동되도록 지정
none	유동성이 없도록 지정, 기본값
inherit	부모 요소의 속성을 이어 받음

clear : float된 박스들의 바로 다음 박스에게 주변을 흐르지 않고 원래대로 아래에 배치되도록 float을 해제시켜주는 속성

그 외 float 해제하는 방법 : float된 요소를 감싸는 박스에 overflow: hidden;



## 문자 관련 스타일

font-family : 문자의 글꼴을 지정하는 속성

여러 단어이거나 한글 글꼴이 경우는 따옴표로 감싸고 여러 개의 글꼴을 쉼표로 구분한다.

font-family: "돋움", Dotum, Arial, sans-serif;

font-size : 문자의 글자 크기를 지정하는 속성

px : 해상도에 따라 상대적으로 기본 단위

% : 부모 요소의 글자 크기를 100% 기준으로 계산한 % 단위

em : 부모 요소의 글자 크기를 100% 기준으로 계산한 100분의 1 단위

rem : em과 비슷하지만 부모 요소가 아닌 최상위(html) 요소의 크기를 100% 기준으로 계산

vw : 뷰포트 너비값의 100분의 1단위

vh : 뷰포트 높이값의 100분의 1단위

font-weight : 문자를 굵게 하거나 굵은 문자를 보통으로 재설정하는 속성

font-style : 문자를 기울어지게 하거나 기울어지지 않게 재설정하는 속성

font-variant : 문자를 작은 대문자로 설정하거나 원래 문자로 재설정하는 속성

line-height : 줄간격을 px, %, em등의 단위로 지정

CSS font - Shorthand property

font: italic small-caps bold 12px/30px Georgia, serif;

font-style

font-variant

font-weight

font-size/line-height

font-family

글자 크기와 글꼴은 생략할 수 없다.

## 웹폰트

@font-face{

font-family: "Open Sans";

src: url(webfonts/OpenSans-Regular.eot);

src: url(webfonts/OpenSans-Regular.woff) format("woff"),

url(webfonts/OpenSans-Regular.ttf) format("ttf");

}

color : 글자의 색상을 지정하는 속성

letter-spacing : 글자 간의 간격

word-spacing : 단어 간의 간격

text-decoration : 글자에 밑줄, 윗줄, 가운데줄을 치거나 쳤던 줄을 없애준다.

text-transform : 대소문자 변경을 실행한다.

text-shadow : 글자에 그림자를 주는 속성

text-align : 블록 요소에만 적용되는 속성으로서 문단을 블록의 왼쪽(left), 가운데(center), 오른쪽(right), 양쪽(justify) 등으로 정렬시킨다.

vertical-align : 이미지나 폼 요소를 위(top), 가운데(middle), 아래(bottom)로 정렬시켜 주는 속성으로 img, input, select, label, 테이블의 th, td등에 사용한다.

text-indent : 문단의 첫머리를 들여쓰기 해주는 속성으로서, 블록 요소에만 적용된다.

text-shadow: 2px 3px 5px rgba(0,0,0,0.4);

속성 값	속성 설명
2px	그림자가 원본에서 떨어지는 가로 거리
3px	그림자가 원본에서 떨어지는 세로 거리
5px	그림자가 흐릿하게 퍼지는 정도
rgba(0,0,0,0.4)	그림자의 색상(red, green, blue, alpha)

white-space : 마크업한 문자들 사이의 공백을 처리하여 내보내는 속성

속성 값	속성 설명
normal	연속 공백들과 줄바꿈이 하나의 공백으로 처리->길면 줄바꿈 일어남
nowrap	연속 공백들과 줄바꿈이 하나의 공백으로 처리->길어도 줄바꿈 없이 박스 밖으로 나감
pre	연속 공백들과 줄바꿈이 소스 그대로 표현됨
pre-wrap	연속 공백들과 줄바꿈이 소스 그대로 표현되나, 길면 줄바꿈 일어남
pre-line	연속 공백들이 하나의 공백으로 처리, 줄바꿈은 소스 그대로 표현되나, 길면 줄바꿈 일어남

text-overflow : 긴 텍스트에 말줄임 기호를 사용하고자 할 때 사용

display : 요소의 성격을 바꿔주는 속성 블록, 인라인 요소 변경

visibility : 요소를 숨기거나 보이도록 해주는 속성

overflow : 요소 안의 내용이 요소의 크기보다 많을 경우 넘치는 부분을 처리하는 속성

opacity : 요소에 투명도를 부여할 수 있는 속성으로 속성값은 0(투명)~1(불투명)로 표현

## 배경(background)

background-color : 요소의 배경 색상을 지정하는 속성

background-image : 요소의 배경에 들어갈 이미지를 지정하는 속성

background-repeat : 배경 이미지를 어떻게 반복시킬지를 지정하는 속성

background-position : 배경 이미지를 원하는 위치로 옮겨주는 속성

background-attachment : 배경 이미지를 요소 내에 고정시킬지 화면에 고정시킬지에 대한 속성

background-size : 배경이미지의 크기를 변경할 수 있는 속성

background-origin : 배경 이미지의 시작점을 정하는 속성

속성 값	속성 설명
border-box	배경 이미지가 테두리의 좌측 상단 모퉁이에서 시작함
padding-box	배경 이미지가 안여백의 좌측 상단 모퉁이에서 시작함 (기본값)
content-box	배경 이미지가 콘텐츠의 좌측 상단부터 시작함

## CSS background - Shorthand property

background: #ffffff url("이미지") no-repeat right top;

background-color

background-image

background-repeat

background-attachment

background-position

## 배경에 gradient 적용하기

background: linear-gradient(red, yellow, green);

background: linear-gradient(to top, red, yellow, green);

background: linear-gradient(to right, red, yellow, green);

background: linear-gradient(to bottom right, red, yellow, green);

background: linear-gradient(to right, red, yellow, green);

background: linear-gradient(45deg, red, yellow, green);

background: radial-gradient(red, yellow, green);

background: radial-gradient(circle, red, yellow, green);

background: repeating-linear-gradient(red, green 10%, blue 20%);

background: repeating-radial-gradient(red, green 10%, blue 20%);

## BOX 관련 속성

속성 값	속성 설명
width: 800px;	요소의 가로 폭을 800px로 지정함
height: 100px;	요소의 높이 값을 100px로 지정함
min-width: 1024px;	요소의 가로폭을 1024px 이상으로 지정함
min-height: 200px;	요소의 세로폭을 200px 이상으로 지정함
max-width:400px;	요소의 너비 값을 400px 이하로 지정함
max-height: 500px;	요소의 높이 값을 500px 이하로 지정함

margin : 요소의 바깥 여백을 지정하는 속성

border : 박스의 테두리를 설정하는 속성

border-radius : 박스의 네 모서리를 둥글게 만들 수 있는 속성

box-sizing : width, height 값에 padding 값이나 border 값은 포함되지 않게 하는 속성

box-shadow : 요소에 그림자를 발생시키는 속성

box-shadow: 8px 15px 10px 7px inset rgba(255,255,255,0.4);

속성 값	속성 설명
8px	그림자가 원본과 가로로 떨어진 간격
15px	그림자가 원본과 세로로 떨어진 간격
10px	그림자의 흐릿한 정도 (blur)
7px	그림자의 확장(spread-양수는 확장, 음수는 축소)
inset	그림자가 요소의 안쪽에 나타남
rgba(255,255,255,0.4);	그림자의 색상

## 기타 CSS3속성

### 2D transform

속성 값	속성 설명
translate(20px, 30px)	우측으로 20px, 아래로 30px 이동 translateX(), translateY() 둘로 사용 가능
rotate(30deg)	30도 회전
scale(0.7, 1.3)	가로 70%로 축소, 세로 130% 확대
skew(30deg, 20deg)	가로 30도 찌그러뜨림, 세로 20도 찌그러뜨림 skewX(), skewY() 둘로 사용 가능

transition : 속성 값을 지정한 시간 동안 부드럽게 변경하는 속성

animation : 애니메이션을 사용하여 요소가 한 스타일에서 다른 스타일로 점진적으로 변경하는 속성

### 다단편집(multi-column)

속성 값	속성 설명
column-count	가로로 나열될 텍스트의 단의 개수
column-gap	단과 단 사이의 여백
column-rule: 1px solid red	단과 단 사이의 구분선

# SVG

SVG는 확장 가능한 벡터 그래픽의 약자입니다.

SVG는 XML 형식의 벡터 기반 그래픽을 정의합니다.

SVG는 웹용 벡터 기반 그래픽을 정의하는 데 사용됩니다.

SVG 파일의 모든 요소와 모든 속성은 애니메이션 가능 SVG는 W3C 권장 사항입니다.

SVG에는 개발자가 사용할 수 있는 몇 가지 미리 정의된 모양 요소가 있습니다.

Rectangle <rect>  
Circle <circle>  
Ellipse <ellipse>  
Line <line>  
Polyline <polyline>  
Polygon <polygon>  
Path <path>

SVG 이미지는 <svg> 요소로 시작합니다.

<svg> 요소의 너비 및 높이 속성은 SVG 이미지의 너비와 높이를 정의합니다.

## SVG <rect>

<rect> 요소의 너비 및 높이 속성은 사각형의 높이와 너비를 정의합니다.

style 속성은 사각형의 CSS 속성을 정의하는 데 사용됩니다.

CSS 채우기 속성은 사각형의 채우기 색상을 정의합니다.

CSS stroke-width 속성은 사각형 테두리의 너비를 정의합니다.

CSS 획 속성은 사각형 테두리의 색상을 정의합니다.

```
<svg width="400" height="110">  
  <rect width="300" height="100" style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />  
</svg>  
  
<svg width="400" height="110">  
  <rect width="300" height="100" style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />  
</svg>
```

## SVG <circle>

<circle> 요소는 원을 그리는 데 사용됩니다.

cx 및 cy 속성은 원 중심의 x 및 y 좌표를 정의합니다. cx 및 cy가 설정되지 않은 경우 원의 중심은 (0, 0)으로 설정됩니다.

r 속성은 원의 반지름을 정의합니다.

획 및 획 너비 속성은 모양의 윤곽선이 표시되는 방식을 제어합니다. 원의 윤곽을 4px 녹색 "테두리"로 설정합니다.

채우기 속성은 원 안의 색상을 나타냅니다. 채우기 색상을 노란색으로 설정합니다.

닫는 </svg> 태그는 SVG 이미지를 닫습니다.

```
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />  
</svg>
```

## SVG <ellipse>

<ellipse> 요소는 타원을 만드는 데 사용됩니다.

타원은 원과 밀접한 관련이 있습니다. 차이점은 타원은 x 및 y 반경이 서로 다른 반면 원은 x 및 y 반경이 동일하다는 것입니다.

cx 속성은 타원 중심의 x 좌표를 정의합니다.

cy 속성은 타원 중심의 y 좌표를 정의합니다.

rx 속성은 수평 반경을 정의합니다.

ry 속성은 수직 반경을 정의합니다.

```
<svg height="140" width="500">
  <ellipse cx="200" cy="80" rx="100" ry="50"
    style="fill:yellow;stroke:purple;stroke-width:2" />
</svg>
```

## SVG Line - <line>

```
<svg height="210" width="500">
  <line x1="0" y1="0" x2="200" y2="200" style="stroke:rgb(255,0,0);stroke-width:2" />
</svg>
```

x1 속성은 x축에서 선의 시작을 정의합니다.

y1 속성은 y축에서 선의 시작을 정의합니다.

x2 속성은 x축에서 선의 끝을 정의합니다.

y2 속성은 y축에서 선의 끝을 정의합니다.

## SVG <polygon>

<polygon> 요소는 최소한 세 면을 포함하는 그래픽을 만드는 데 사용됩니다.

다각형은 직선으로 이루어지며 모양은 "닫힌" 상태입니다(모든 선이 연결됨).

```
<svg height="210" width="500">
  <polygon points="200,10 250,190 160,210" style="fill:lime;stroke:purple;stroke-width:1" />
</svg>
```

```
<svg height="250" width="500">
  <polygon points="220,10 300,210 170,250 123,234"
    style="fill:lime;stroke:purple;stroke-width:1" />
</svg>
```

## SVG <polyline>

<polyline> 요소는 직선으로만 구성된 모양을 만드는 데 사용됩니다(여러 점에서 연결됨)

포인트 속성은 폴리라인을 그리는 데 필요한 포인트 목록(x 및 y 좌표 쌍)을 정의합니다.

```
<svg height="200" width="500">
  <polyline points="20,20 40,25 60,40 80,120 120,140 200,180"
    style="fill:none;stroke:black;stroke-width:3" />
</svg>
```

## SVG <path>

<path> 요소는 경로를 정의하는 데 사용됩니다.

M = moveto

L = lineto

H = horizontal lineto

V = vertical lineto

C = curveto

S = smooth curveto

Q = quadratic Bézier curve

T = smooth quadratic Bézier curveto

A = elliptical Arc

Z = closepath

위의 모든 명령어는 소문자로도 표현할 수 있습니다. 대문자는 절대 위치를 의미하고 소문자는 상대적 위치를 의미합니다.

```
<svg height="210" width="400">  
  <path d="M150 0 L75 200 L225 200 Z" />  
</svg>
```

## SVG <text>

<text> 요소는 텍스트를 정의하는 데 사용됩니다.

```
<svg height="30" width="200">  
  <text x="0" y="15" fill="red">I love SVG!</text>  
</svg>
```

## SVG Stroke Properties

stroke

stroke-width

stroke-linecap

stroke-dasharray

g태그는 그룹을 의미하는 태그

stroke 속성은 요소의 선, 텍스트 또는 윤곽선의 색상을 정의합니다.

```
<svg height="80" width="300">  
  <g fill="none">  
    <path stroke="red" d="M5 20 l215 0" />  
    <path stroke="black" d="M5 40 l215 0" />  
    <path stroke="blue" d="M5 60 l215 0" />  
  </g>  
</svg>
```

stroke-width 속성은 요소의 선, 텍스트 또는 윤곽선의 두께를 정의합니다.

```
<svg height="80" width="300">  
  <g fill="none" stroke="black">
```



```
<path stroke-width="2" d="M5 20 l215 0" />
<path stroke-width="4" d="M5 40 l215 0" />
<path stroke-width="6" d="M5 60 l215 0" />
</g>
</svg>
```

stroke-linecap 속성은 열린 경로에 대한 다양한 유형의 끝을 정의합니다.

```
<svg height="80" width="300">
  <g fill="none" stroke="black" stroke-width="6">
    <path stroke-linecap="butt" d="M5 20 l215 0" />
    <path stroke-linecap="round" d="M5 40 l215 0" />
    <path stroke-linecap="square" d="M5 60 l215 0" />
  </g>
</svg>
```

stroke-dasharray 속성은 점선을 만드는 데 사용됩니다.

```
<svg height="80" width="300">
  <g fill="none" stroke="black" stroke-width="4">
    <path stroke-dasharray="5,5" d="M5 20 l215 0" />
    <path stroke-dasharray="10,10" d="M5 40 l215 0" />
    <path stroke-dasharray="20,10,5,5,5,10" d="M5 60 l215 0" />
  </g>
</svg>
```

stroke-dashoffset, svg을 나타낼 지점 지정하기

svg가 어떤 지점에서부터 시작할지 정해주는 속성이다.

stroke-dasharray는 점선을 만들어주고

stroke-dashoffset은 어디부터 시작할 것인지 정해줍니다.