



grid

# 그리드 레이아웃(grid layout)

---



그리드 레이아웃이란 이름 그대로 격자 형태의 레이아웃을 만드는 2차원 레이아웃 방식을 말합니다.

그리드 레이아웃 방식에서, 그리드 아이템의 배치와 정렬은 그리드 컨테이너 내부의 행과 열의 상호작용을 통해 결정됩니다.

그리드 컨테이너 : 그리드 방식으로 레이아웃을 결정할 요소

그리드 아이템 : 그리드 컨테이너 내부에서 그리드 방식으로 배치되는 요소

# display: grid;

그리드 레이아웃 방식을 사용하기 위해  
그리드 컨테이너를 만들어요!



# display: inline-grid;

그리드 컨테이너가  
인라인 레벨 요소이길 원한다면?



# 이것부터 알고 시작하자



그리드 방식은 컨테이너에 행(row)과 열(column)을 제공합니다. 행과 열을 동시에 제어해 요소를 배치할 수 있으며, 컨테이너의 구조는 다음과 같습니다.

grid-container

grid-item	grid-item	grid-item
grid-item	grid-item	grid-item
grid-item	grid-item	grid-item

# 이것부터 알고 시작하자



그리드 방식은 컨테이너에 행(row)과 열(column)을 제공합니다. 행과 열을 동시에 제어해 요소를 배치할 수 있으며, 컨테이너의 구조는 다음과 같습니다.

grid-container

grid-item	grid-item	grid-item
grid-item	grid-item	grid-item
grid-item	grid-item	grid-item

# 이것부터 알고 시작하자



그리드 방식은 컨테이너에 행(row)과 열(column)을 제공합니다. 행과 열을 동시에 제어해 요소를 배치할 수 있으며, 컨테이너의 구조는 다음과 같습니다.

grid-container

grid-item	grid-item	grid-item
grid-item	grid-item	grid-item
grid-item	grid-item	grid-item

- grid-template-columns
- grid-template-rows
- gap(grid-gap)
- 트랙 관련 함수
- grid-column & row
- grid-template-areas
- grid-area
- align-items
- align-self
- justify-items
- justify-self
- align-content
- justify-content

grid layout 관련 속성들!





- grid-template-columns
- grid-template-rows
- gap(grid-gap)
- 트랙 관련 함수
- grid-column & row
- grid-template-areas
- grid-area
- align-items
- align-self
- justify-items
- justify-self
- align-content
- justify-content

이번 시간에는?



# grid-template-columns



그리드 컨테이너의 트랙 중 열(column) 트랙 내 아이템들의 크기를 지정할 수 있는 속성입니다. 그리드 컨테이너에서 트랙이란 행 또는 열을 뜻합니다.

속성값	의미
none	기본값. 명시한 값이 없으므로 암묵적으로 값이 정해집니다.
수치	길이를 나타내는 음수가 아닌 값을 지정합니다.
그 외	다양한 키워드나 CSS 함수를 사용해 지정하기도 합니다.

실습을 통해서 알아볼까요!?

# grid-template-rows



그리드 컨테이너의 트랙 중 행(row) 트랙 내 아이템들의 크기를 지정할 수 있는 속성입니다.

속성값	의미
none	기본값. 명시한 값이 없으므로 암묵적으로 값이 정해집니다.
수치	길이를 나타내는 음수가 아닌 값을 지정합니다.
그 외	다양한 키워드나 CSS 함수를 사용해 지정하기도 합니다.

실습을 통해서 알아보까요!?

# gap(grid-gap)



그리드 아이템 사이의 간격을 지정하는 속성입니다. 행에서의 간격과 열에서의 간격을 똑같이 지정할 수도 있고, 각자 따로 지정할 수도 있습니다.  
row-gap과 column-gap의 단축속성입니다.

속성값	의미
normal	기본값. 명시한 값이 없는 기본적인 상태.
수치	길이를 나타내는 값을 지정합니다. 다양한 단위 사용 가능.

실습을 통해서 알아보까요!?

- grid-template-columns
- grid-template-rows
- gap(grid-gap)
- **트랙 관련 함수**
- grid-column & row
- grid-template-areas
- grid-area
- align-items
- align-self
- justify-items
- justify-self
- align-content
- justify-content

이번 시간에는?



# 트랙 관련 함수



그리드 컨테이너의 트랙(행과 열) 크기를 지정할 때 사용할 수 있는 유용한 함수들이 있습니다.

함수	기능
repeat()	반복되는 값을 자동으로 처리할 수 있는 함수.
minmax()	최솟값과 최대값을 각각 지정할 수 있는 함수.
auto-fill & auto-fit	반응형을 고려해 사용할 수 있는 키워드들 (함수X)

실습을 통해서 알아보까요!?



- grid-template-columns
- grid-template-rows
- gap(grid-gap)
- 트랙 관련 함수
- **grid-column & row**
- grid-template-areas
- grid-area
- align-items
- align-self
- justify-items
- justify-self
- align-content
- justify-content

이번 시간에는?

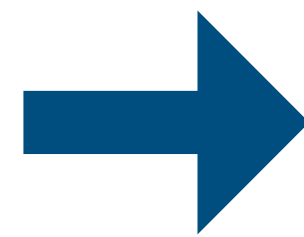


# grid-column & row



grid-column과 grid-row 속성을 이용하면 그리드 컨테이너의 줄 번호를 이용해 아이템을 배치할 수 있습니다.

grid-container 행, 열  
각각의 줄 번호는 이렇게



	1	2	3	4
1	grid-item	grid-item	grid-item	
2	grid-item	grid-item	grid-item	
3	grid-item	grid-item	grid-item	
4				



- grid-template-columns
- grid-template-rows
- gap(grid-gap)
- 트랙 관련 함수
- grid-column & row
- **grid-template-areas**
- **grid-area**
- align-items
- align-self
- justify-items
- justify-self
- align-content
- justify-content

이번 시간에는?



# grid-template-areas



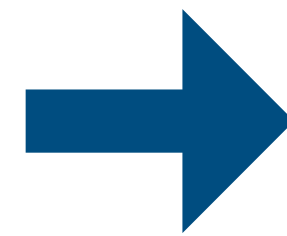
그리드 영역(아이템)의 이름을 이용해 레이아웃의 형태를 정의할 수 있습니다.

```
grid-template-areas:
```

```
"거 거 호"
```

```
"다 다 청"
```

```
"고 고 고";
```



# grid-area



그리드 영역(아이템)의 이름을 지정할 때 사용하는 속성입니다.

```
li:nth-child(1){ grid-area: 거; }  
li:nth-child(2){ grid-area: 호; }  
li:nth-child(3){ grid-area: 다; }  
li:nth-child(4){ grid-area: 청; }  
li:nth-child(5){ grid-area: 고; }
```

li 요소 하나가  
그리드 아이템 하나!

- grid-template-columns
- grid-template-rows
- gap(grid-gap)
- 트랙 관련 함수
- grid-column & row
- grid-template-areas
- grid-area
- align-items
- align-self
- justify-items
- justify-self
- align-content
- justify-content

이번 시간에는?



# align-items



플렉스박스 방식에서와 유사한 역할을 합니다. 그리드 컨테이너 행 트랙의 높이를 기준으로 그리드 아이템의 배치를 결정합니다.

속성값	의미
stretch	기본값. 그리드 아이템들이 트랙 높이만큼 확장됩니다.
start	트랙 상단으로부터 하단을 향해 배치.
end	트랙 하단으로부터 상단을 향해 배치.
center	할당된 공간의 중심부에 배치.

사용 가능한 모든 속성값이 아닌,  
대표적인 속성값 위주로 정리했습니다!

# align-self



각각의 그리드 아이템이 어떤 식으로 배치될 것인지를 스스로 결정합니다.

속성값	의미
stretch	기본값. 그리드 아이템들이 트랙 높이만큼 확장됩니다.
start	트랙 상단으로부터 하단을 향해 배치.
end	트랙 하단으로부터 상단을 향해 배치.
center	할당된 공간의 중심부에 배치.

사용 가능한 모든 속성값이 아닌,  
대표적인 속성값 위주로 정리했습니다!



# justify-items



수평축(행)을 따라 그리드 아이টে을 정렬하고자 할 때 사용할 수 있는 속성으로, 그리드 컨테이너에 지정합니다. 아이টে에 할당된 열 방향 너비가 기준이 됩니다.

속성값	의미
stretch	기본값. 그리드 아이টে들이 트랙의 너비만큼 확장됩니다.
start	트랙 시작점으로부터 끝점을 향해 배치.
end	트랙 끝점으로부터 시작점을 향해 배치.
center	할당된 공간의 중심부에 배치.

사용 가능한 모든 속성값이 아닌,  
대표적인 속성값 위주로 정리했습니다!

# justify-self



수평축(행)을 따라 그리드 아이টে을 정렬하고자 할 때 사용할 수 있는 속성으로, 각각의 그리드 아이টে에 지정합니다. align-items & align-self의 관계와 유사.

속성값	의미
stretch	기본값. 그리드 아이টে들이 트랙의 너비만큼 확장됩니다.
start	트랙 시작점으로부터 끝점을 향해 배치.
end	트랙 끝점으로부터 시작점을 향해 배치.
center	할당된 공간의 중심부에 배치.

사용 가능한 모든 속성값이 아닌,  
대표적인 속성값 위주로 정리했습니다!



- grid-template-columns
- grid-template-rows
- gap(grid-gap)
- 트랙 관련 함수
- grid-column & row
- grid-template-areas
- grid-area
- align-items
- align-self
- justify-items
- justify-self
- align-content
- justify-content

이번 시간에는?



# align-content & justify-content



플렉스박스 방식과 유사하게 사용합니다. 그리드 컨테이너의 수직축과 수평축에서의 아이템 정렬 방식을 결정합니다. 컨테이너에 여유 공간이 있을 때 사용할 수 있습니다.

속성값	의미
start	트랙의 시작점으로부터 끝점을 향해 배치
end	트랙의 끝점으로부터 시작점을 향해 배치
center	트랙의 중심부에 배치
space-between	트랙에서 일정한 간격을 둔 채 양끝 정렬 배치

사용 가능한 모든 속성값이 아닌,  
대표적인 속성값 위주로 정리했습니다!