



# JAVA 웹 개발자 양성과정 HTML&CSS

## 10강 - Css Grid

By SoonGu Hong

# JAVA 웹 개발자 양성과정

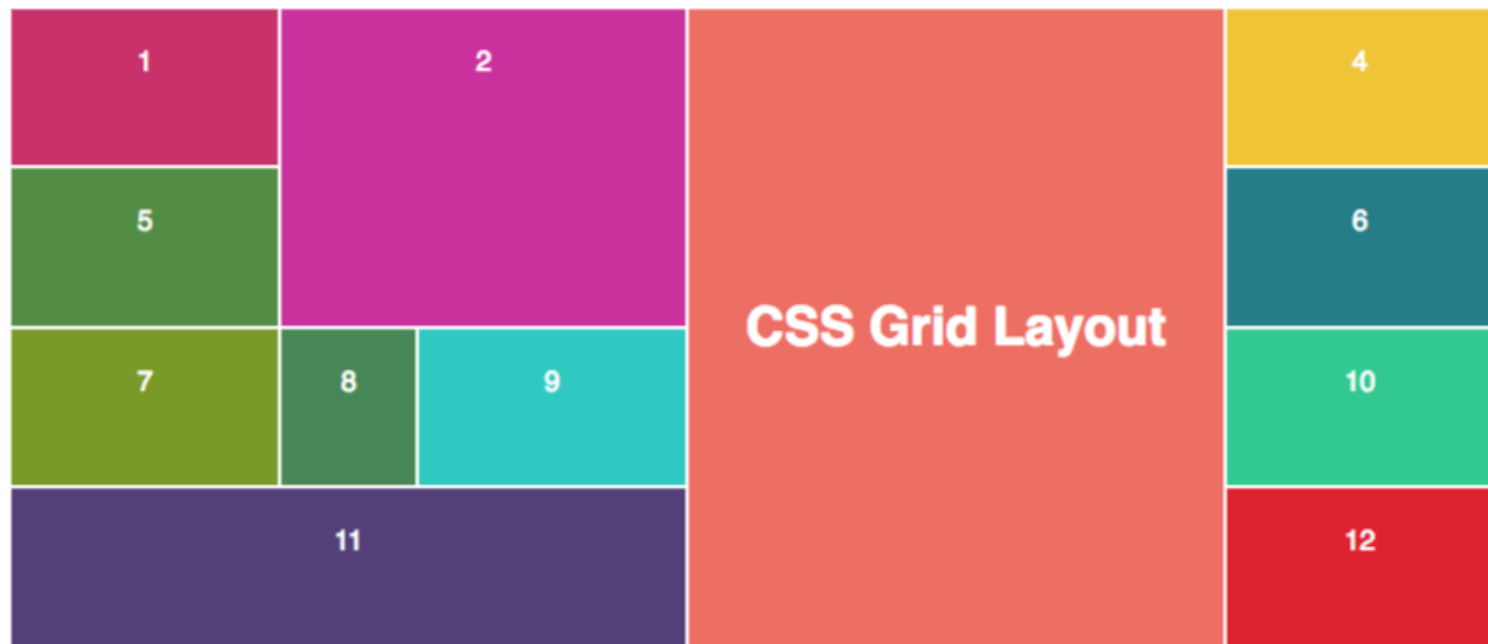
## HTML&CSS

### 1. Grid란?



## \* 좀 더 복잡한 레이아웃 시스템 Grid

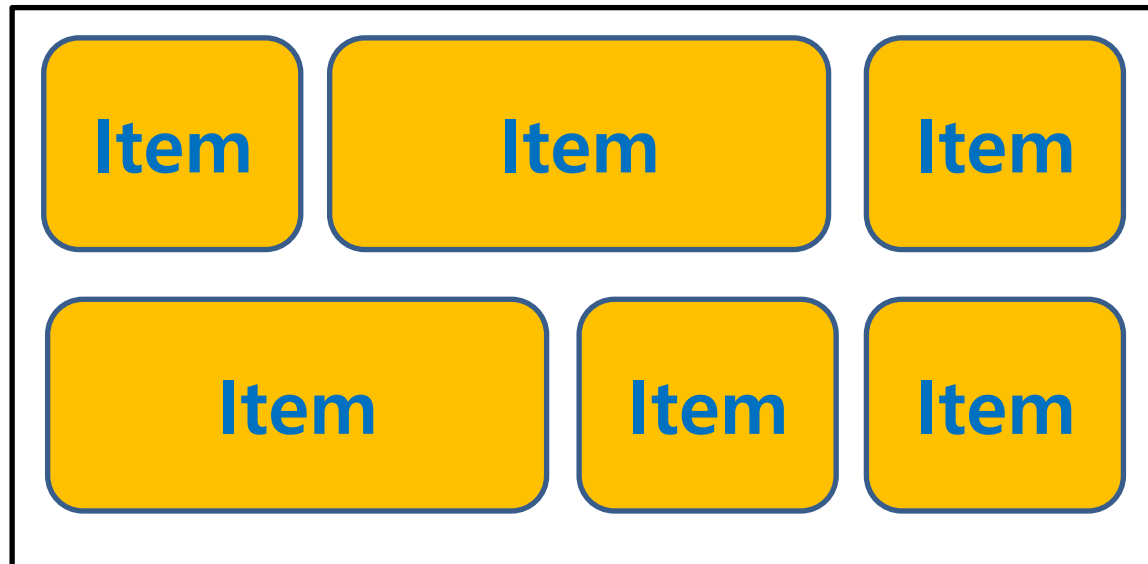
- Css Grid는 2차원의 레이아웃 시스템을 제공합니다.
- Flex Box는 비교적 단순한 1차원 레이아웃에 주로 사용하지만 좀 더 복잡한 레이아웃을 위해서 Grid 시스템이 개발되었습니다.



## \* Grid Container와 Grid Items

- Grid도 Flex와 마찬가지로 **Container**와 **Item**이라는 두 가지 개념으로 구분되어 있습니다.

### Container





# JAVA 웹 개발자 양성과정

## HTML&CSS

## 2. Grid Container

### Grid Container

---

속성명	의미
display	grid container를 정의
grid-template-rows	명시적 행의 크기를 정의
grid-template-columns	명시적 열의 크기를 정의
grid-template-area	영역(Area) 이름을 참조하여 템플릿 생성
grid-template	grid-template-xxx 의 단축 속성
row-gap	행과 행 사이의 간격을 정의
column-gap	열과 열 사이의 간격을 정의
gap	xxx-gap 의 단축 속성

## \* Grid Container의 속성들

속성명	의미
grid-auto-rows	암시적인 행의 크기를 정의
grid-auto-columns	암시적인 열의 크기를 정의
grid-auto-flow	자동 배치 알고리즘 방식을 정의
grid	grid-template-xxx와 grid-auto-xxx의 단축 속성
align-content	그리드 콘텐츠를 수직(열 축) 정렬
justify-content	그리드 콘텐츠를 수평(행 축) 정렬
place-content	align-content와 justify-content의 단축 속성
align-items	그리드 아이템들을 수직(열 축) 정렬
justify-items	그리드 아이템들을 수평(행 축) 정렬
place-items	align-items와 justify-items의 단축 속성

## 속성 값

display

- Grid Container를 정의.

값	의미
grid	block특성의 container를 정의
inline-grid	inline특성의 container를 정의



grid-  
template-  
rows

### 속성 값

- 명시적 행의 크기를 정의.
- fr (fraction, 공간 비율) 단위를 사용 가능.
- repeat() 함수를 사용 가능.

```
display: grid;  
grid-template-rows: 1행 크기, 2행크기;
```

---

```
display: grid;  
grid-template-rows: 100px, 200px;
```

grid-  
template-  
columns

### 속성 값

- 명시적 열의 크기를 정의.
- fr (fraction, 공간 비율) 단위를 사용 가능.
- repeat() 함수를 사용 가능.

```
display: grid;  
grid-template-columns: 1열 크기, 2열 크기;
```

---

```
display: grid;  
grid-template-columns: 100px, 200px;
```

# \* grid-template-areas

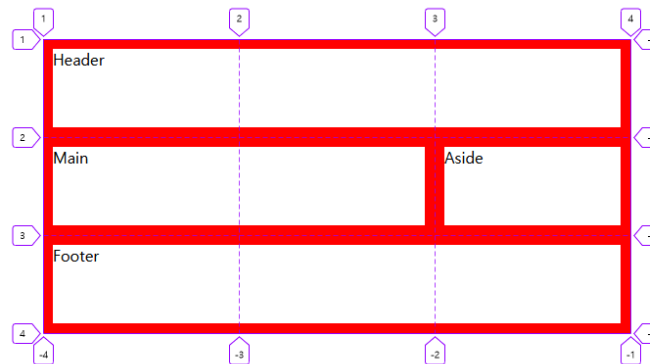
grid-  
template-  
areas

## 속성 값

- 지정된 그리드 영역 이름(grid-area)를 참조해 그리드 템플릿을 생성합니다.

```
.wrapper header {  
  grid-area: header;  
}  
.wrapper section {  
  grid-area: main;  
}  
.wrapper aside {  
  grid-area: aside;  
}  
.wrapper footer {  
  grid-area: footer;  
}
```

```
display: grid;  
grid-template-rows: repeat(3, 100px);  
grid-template-columns: repeat(3, 1fr);  
grid-template-areas:  
  "header header header"  
  "main main aside"  
  "footer footer footer";
```



## \* row-gap, column-gap

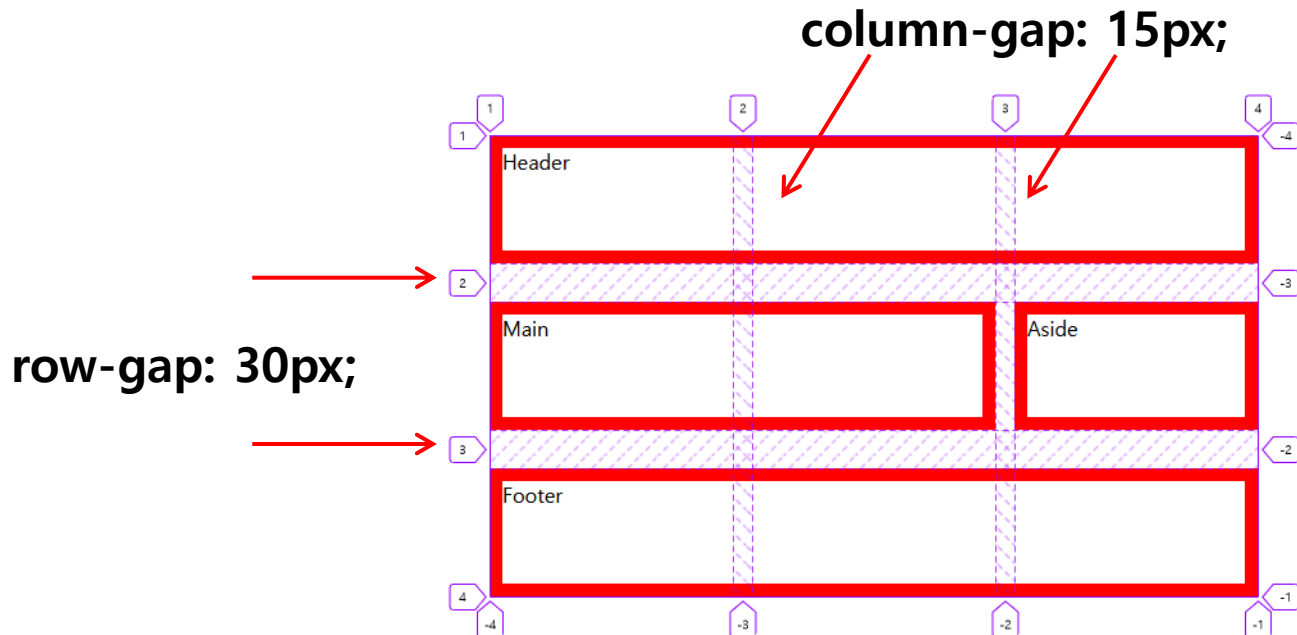
### 속성 값

row-gap

- 각 행과 행 사이의 간격을 지정.

column-gap

- 각 열과 열 사이의 간격을 지정.



## 속성 값

gap

- row-gap과 column-gap의 단축 속성.

```
gap: row-gap column-gap;
```

```
gap: 20px 10px;
```

```
/* 하나의 값으로 통일 가능 */
```

```
gap: 20px; /* row:20px, col:20px */
```

```
/* 한쪽만 주고 싶다면 다음과 같이 사용 */
```

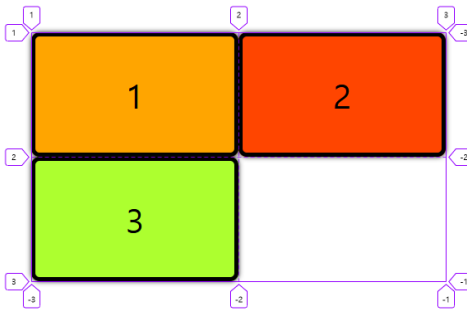
```
gap: 20px 0; /* row:20px, col: 0px */
```

## 속성 값

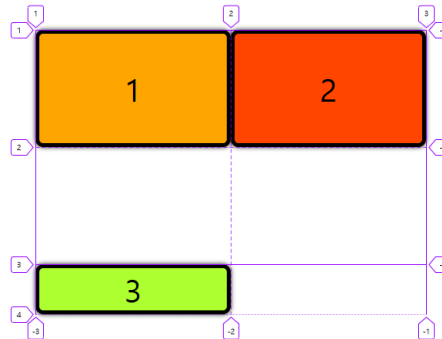
grid-auto-rows

- 암시적 행의 크기를 정의.
- 암시적 크기가 적용된 행과 열은 양수 라인 번호만 사용할 수 있음. (음수 불가)

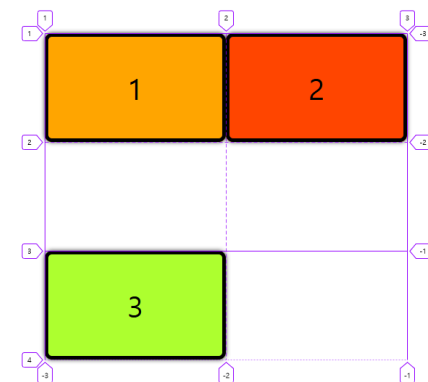
명시적으로 2x2 정의



3번 아이템을 암시적 행에 배치  
`.item3 { grid-row: 3 / 4; }`



암시적 행의 크기를 정의  
`grid-auto-rows: 150px`

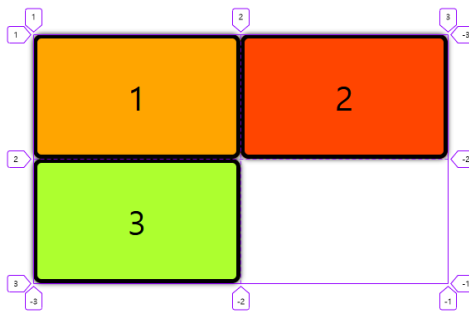


## 속성 값

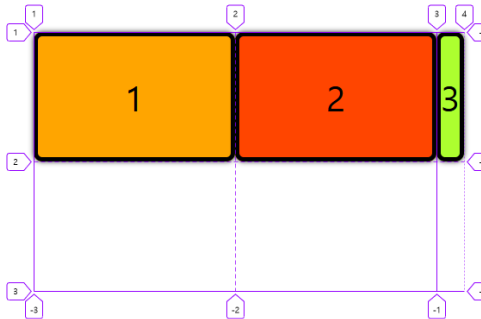
grid-auto-columns

- 암시적 행의 크기를 정의.
- 암시적 크기가 적용된 행과 열은 양수 라인 번호만 사용할 수 있음. (음수 불가)

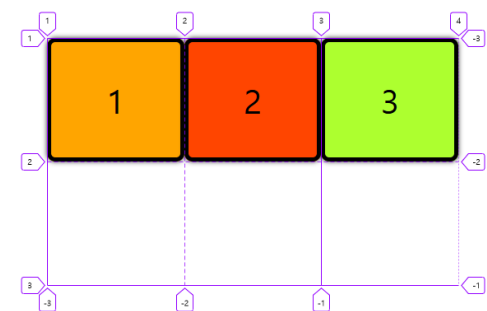
명시적으로 2x2 정의



3번 아이템을 암시적 열에 배치  
`.item3 { grid-column: 3 / 4; }`



암시적 열의 크기를 정의  
`grid-auto-columns: 1fr;`



## 속성 값

grid-auto-flow

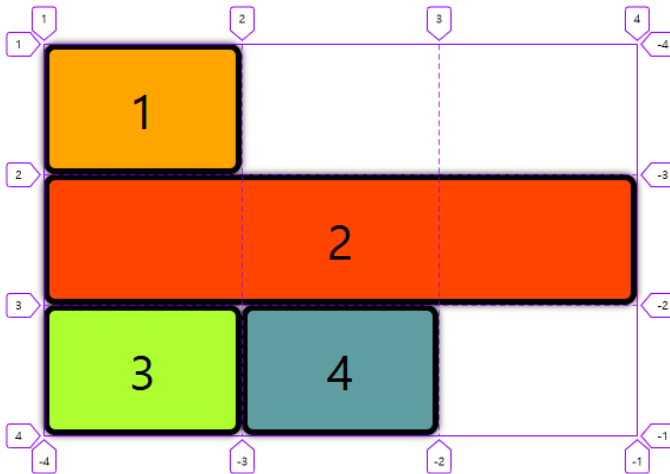
- 배치하지 않은 아이템을 자동배치 알고리즘으로 처리

값	의미
row	각 행 축을 따라 차례로 배치 (기본값)
column	각 열 축을 따라 차례로 배치
row dense	각 행 축을 따라 차례로 배치, 빈 영역을 메움.
column dense	각 열 축을 따라 차례로 배치, 빈 영역을 메움.

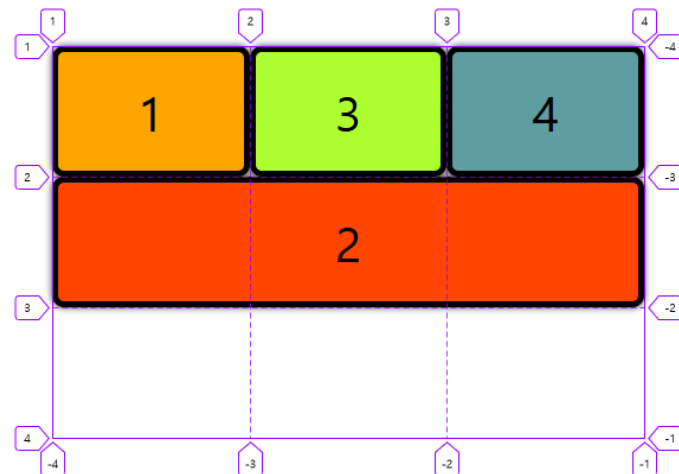


## \* grid-auto-flow

- 명시적으로 3x3 정의
- 2번 아이템에 grid-column: span 3; 적용



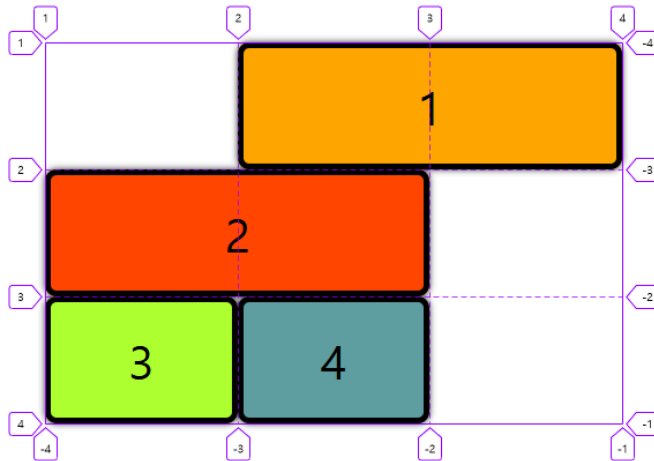
grid-auto-flow: **row;**



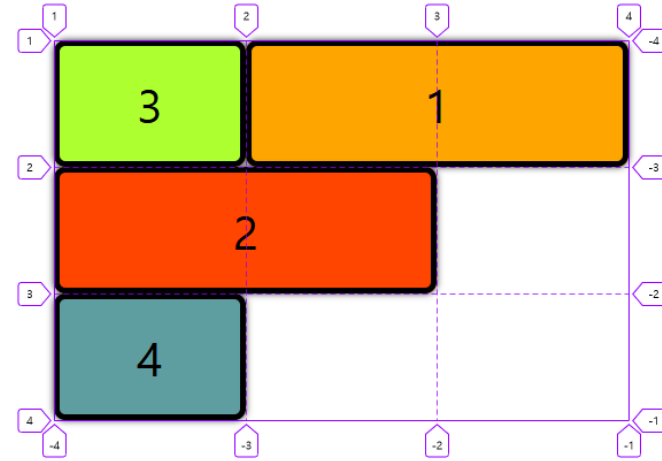
grid-auto-flow: **row dense;**

## \* grid-auto-flow

- 명시적으로 3x3 정의
- 1번 아이템에 grid-column: 2 /span 2; 적용
- 2번 아이템에 grid-column: span 2; 적용



grid-auto-flow: **column**;



grid-auto-flow: **column dense**;

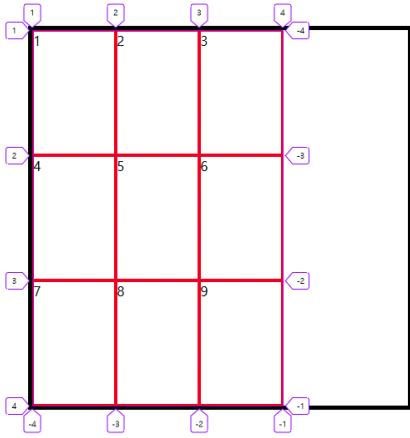
## 속성 값

align-content

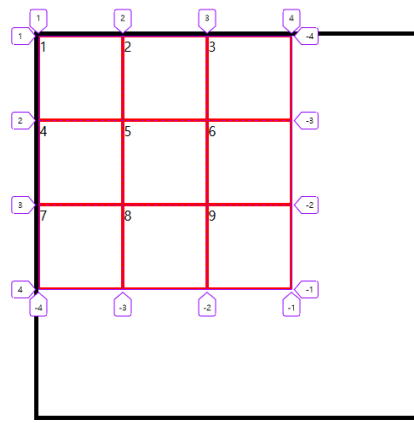
- 그리드 콘텐츠를 수직(열 축) 정렬함.
- 그리드 콘텐츠의 세로 너비가 그리드 컨테이너보다 작아야 함.

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

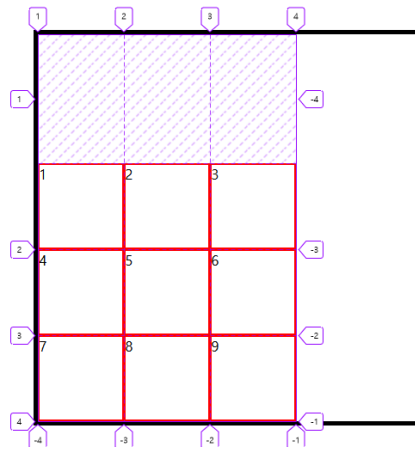
# \* align-content



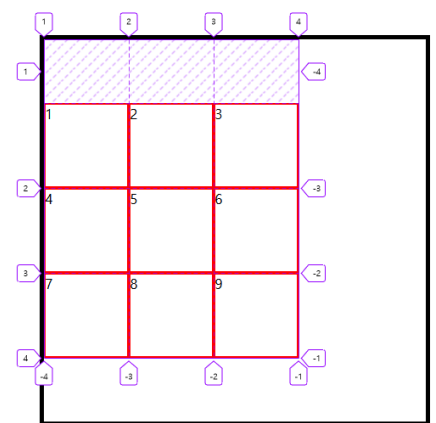
**normal, stretch**



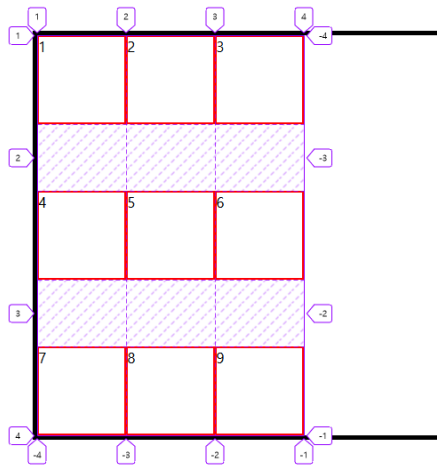
**start**



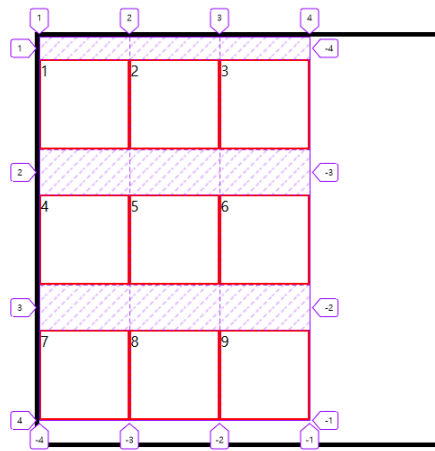
**end**



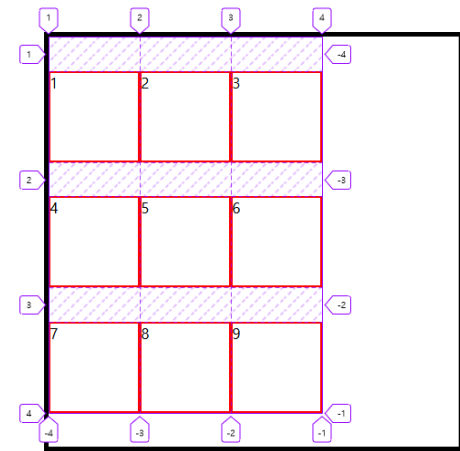
**center**



**space-between**



**space-around**



**space-evenly**

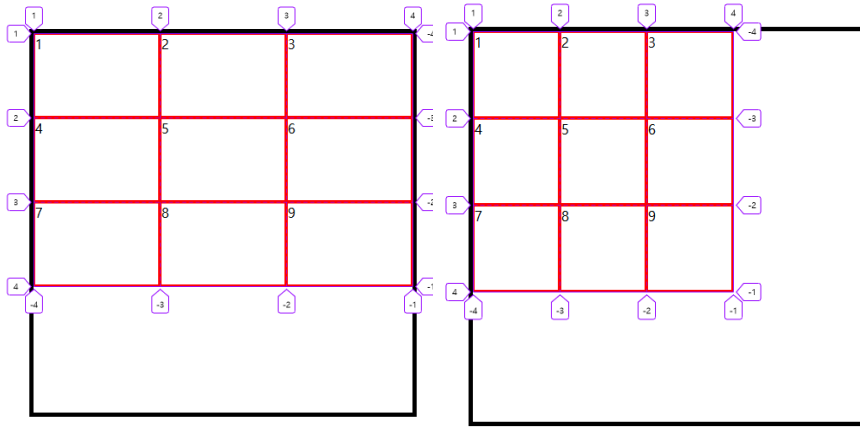
## 속성 값

justify-content

- 그리드 콘텐츠를 수평(행 축) 정렬함.
- 그리드 콘텐츠의 가로 너비가 그리드 컨테이너보다 작아야 함.

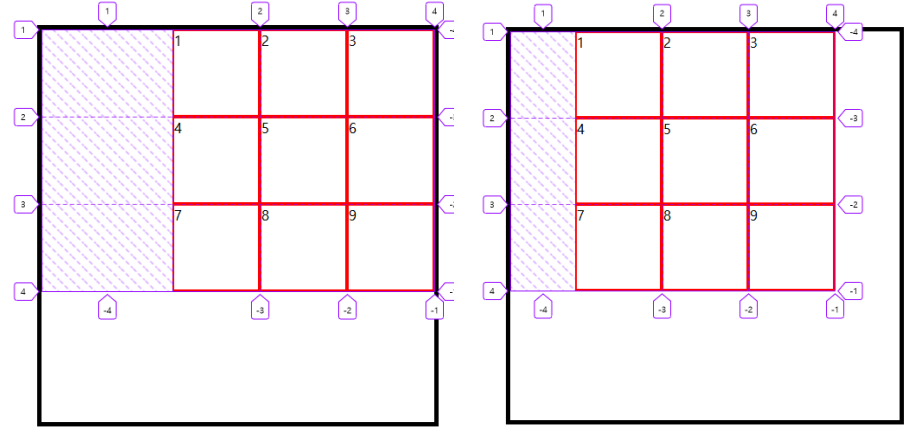
값	의미
normal	stretch와 같음 (기본값)
stretch	열 축을 채우기 위해 그리드 콘텐츠를 늘림
start	시작점(왼쪽) 부터 정렬
end	끝점(오른쪽) 부터 정렬
center	수평 가운데 정렬
space-between	첫 열은 시작점, 끝 열은 끝점, 나머지는 여백으로 고르게 정렬
space-around	각 열 좌우에 여백을 고르게 정렬
space-evenly	모든 여백을 고르게 정렬

# \* justify-content



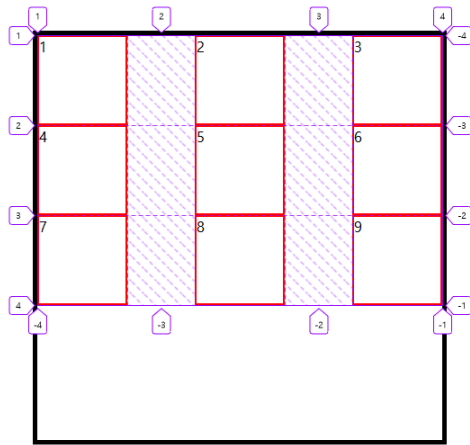
**normal, stretch**

**start**

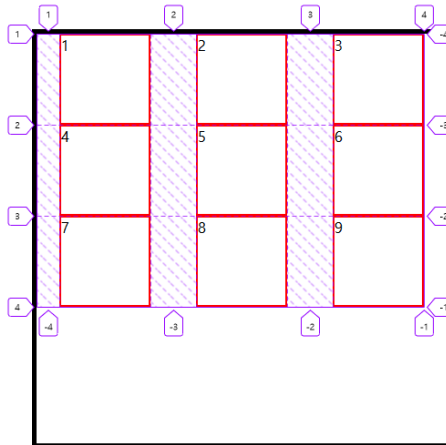


**end**

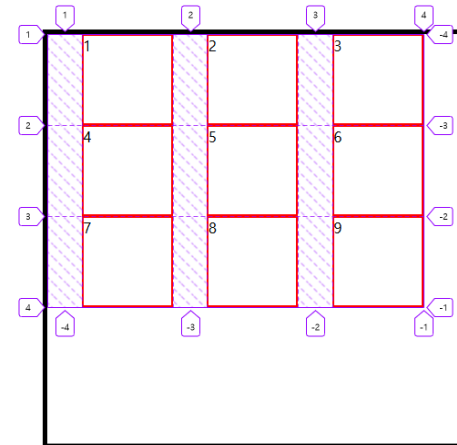
**center**



**space-between**



**space-around**



**space-evenly**

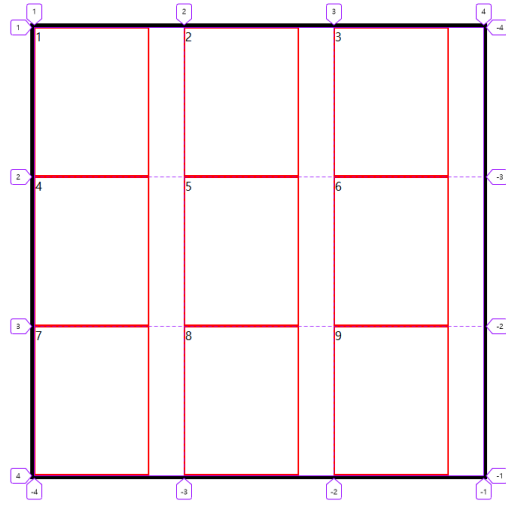
## 속성 값

align-items

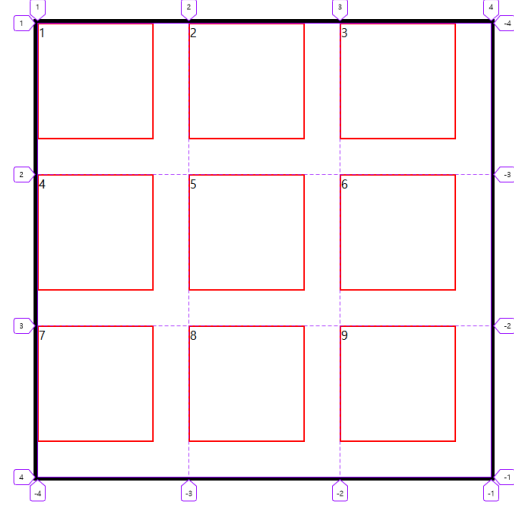
- 그리드 아이тем들을 수직(열 축) 정렬함.
- 그리드 아이тем의 세로 너비가 자신이 속한 그리드 행보다 작아야 함.

값	의미
normal	stretch와 같음 (기본값)
stretch	열 축을 채우기 위해 그리드 아이тем을 늘림
start	시작점(위쪽) 정렬
end	끝점(아래쪽) 정렬
center	수직 가운데 정렬

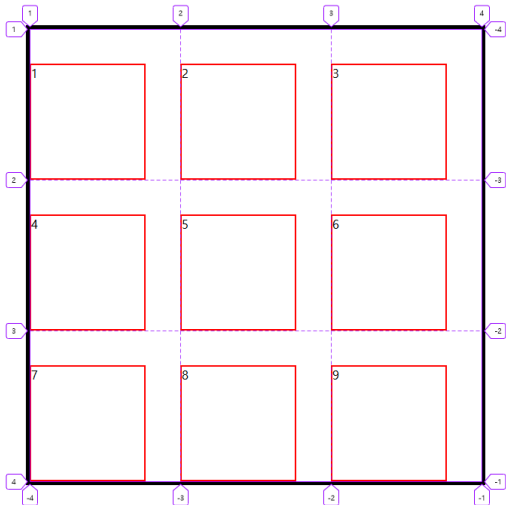
# \* align-items



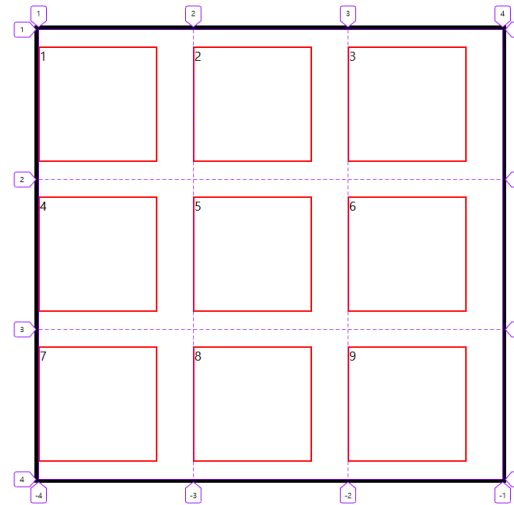
**normal, stretch**



**start**



**end**



**center**



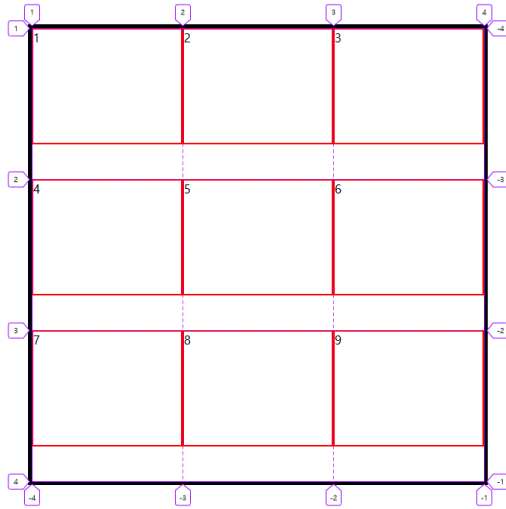
## 속성 값

justify-items

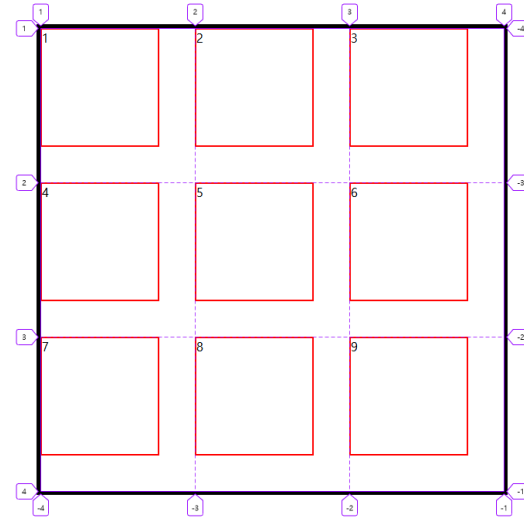
- 그리드 아이тем들을 수평(행 축) 정렬함.
- 그리드 아이тем의 가로 너비가 자신이 속한 그리드 열보다 작아야 함.

값	의미
normal	stretch와 같음 (기본값)
stretch	행 축을 채우기 위해 그리드 아이тем을 늘림
start	시작점(왼쪽) 정렬
end	끝점(오른쪽) 정렬
center	수직 가운데 정렬

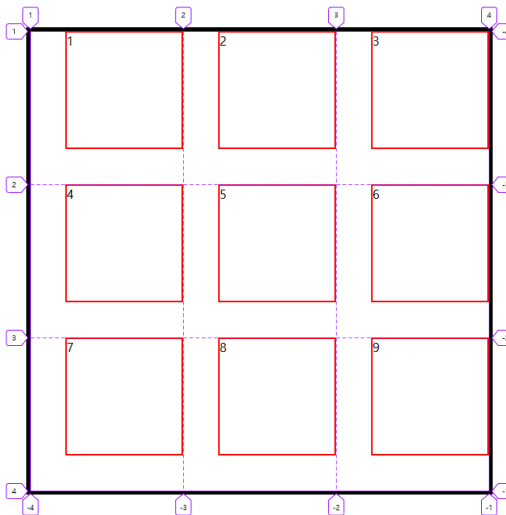
# \* justify-items



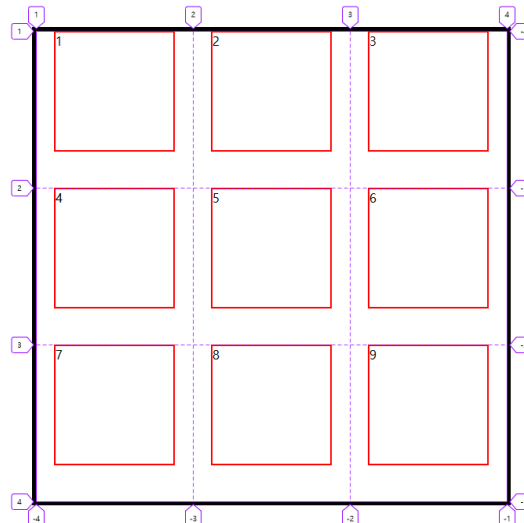
**normal, stretch**



**start**



**end**

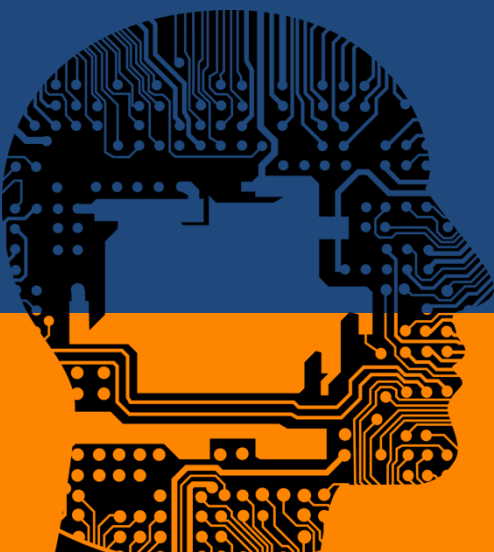


**center**

# JAVA 웹 개발자 양성과정

## HTML&CSS

### 3. Grid Items



### Grid Items

속성명	의미
grid-row-start	그리드 아이템의 행 시작 위치 지정
grid-row-end	그리드 아이템의 행 끝 위치 지정
grid-row	grid-row-xxx 의 단축 속성
grid-column-start	그리드 아이템의 열 시작 위치 지정
grid-column-end	그리드 아이템의 열 끝 위치 지정
grid-column	grid-column-xxx 의 단축 속성
grid-area	영역 이름을 설정
align-self	단일 그리드 아이템을 수직 정렬
justify-self	단일 그리드 아이템을 수평 정렬
place-self	align-self와 justify-self의 단축 속성
order	그리드 아이템의 배치 순서를 지정
z-index	그리드 아이템이 쌓이는 순서를 지정

### 속성 값

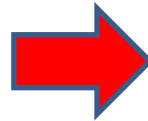
grid-row

- grid-row-start와 grid-row-end의 단축 속성
- span은 확장의 개념

```
grid-row-start: 2;  
grid-row-end: 5;
```

```
grid-row: 2 / 5;
```

```
grid-row: 2 / span 3;
```



모두 같은 의미!

## 속성 값

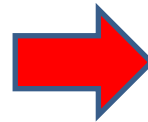
grid-column

- grid-column-start와 grid-column-end의 단축 속성
- span은 확장의 개념

```
grid-column-start: span 3;  
grid-column-end: 4;
```

```
grid-column: span 3 / 4;
```

```
grid-column: 1 / 4;
```



모두 같은 의미!

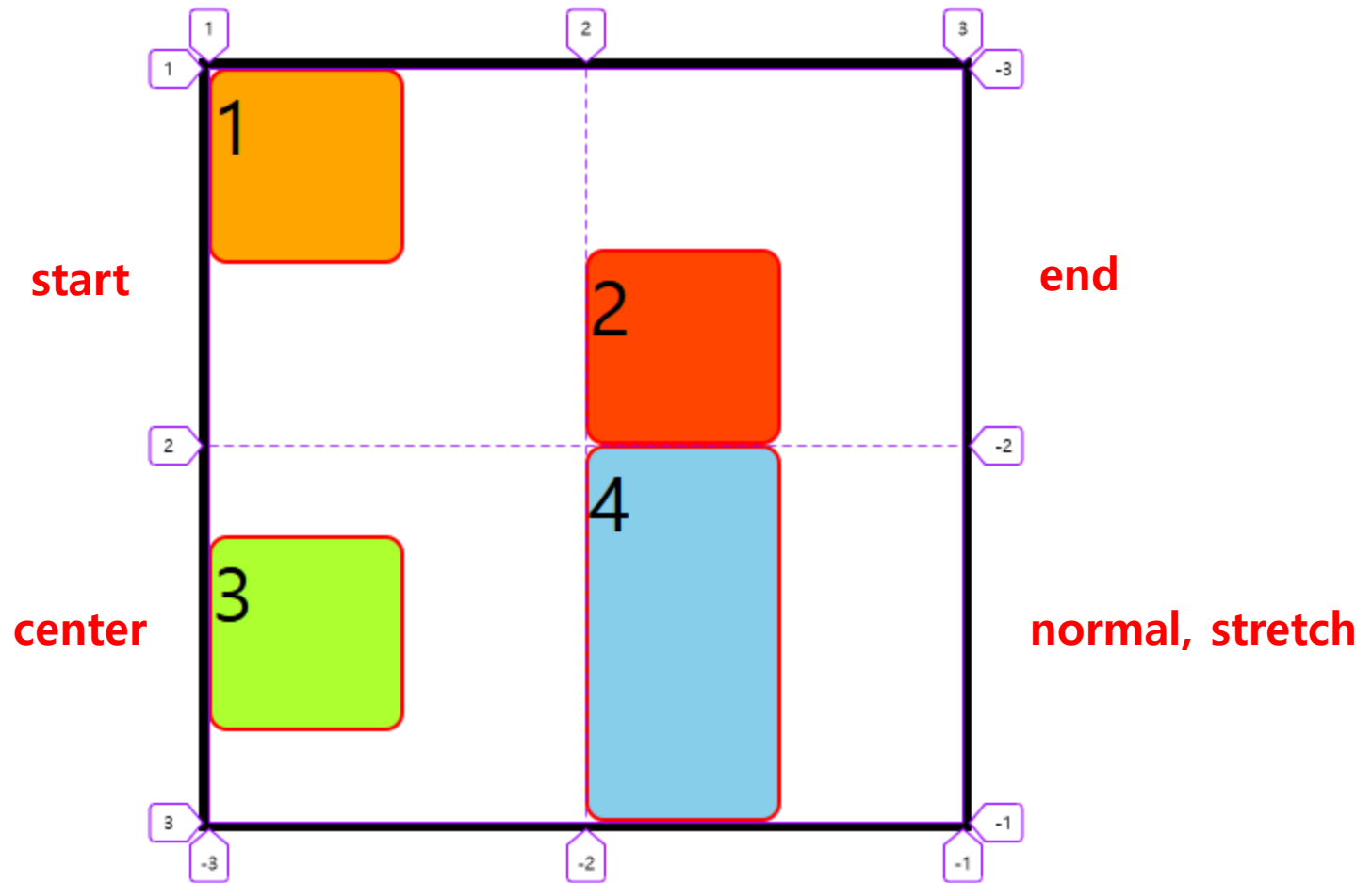
## 속성 값

align-self

- 단일 그리드 아이템들을 수직(열 축) 정렬함.
- 그리드 아이템의 세로 너비가 자신이 속한 그리드 행보다 작아야 함.

값	의미
normal	stretch와 같음 (기본값)
stretch	열 축을 채우기 위해 그리드 아이템을 늘림
start	시작점(위쪽) 정렬
end	끝점(아래쪽) 정렬
center	수직 가운데 정렬

## \* align-self





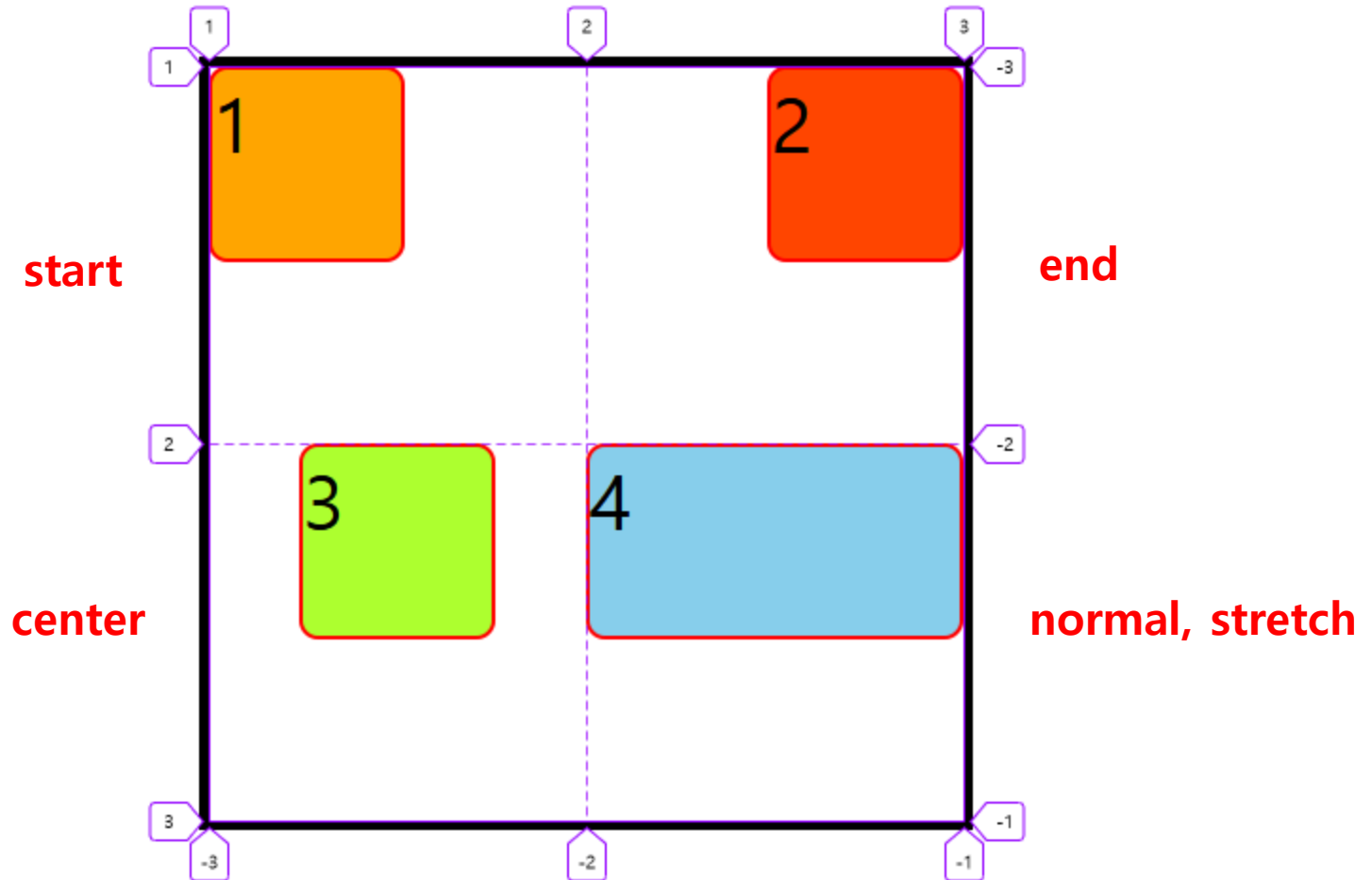
## 속성 값

justify-self

- 단일 그리드 아이템들을 수평(행 축) 정렬함.
- 그리드 아이템의 가로 너비가 자신이 속한 그리드 열보다 작아야 함.

값	의미
normal	stretch와 같음 (기본값)
stretch	행 축을 채우기 위해 그리드 아이템을 늘림
start	시작점(왼쪽) 정렬
end	끝점(오른쪽) 정렬
center	수직 가운데 정렬

# \* justify-self



### 속성 값

order

- 그리드 아이템이 자동 배치되는 순서를 변경함.
- 숫자가 작을수록 앞에 배치됨. (기본값 0)

z-index

- 아이템이 쌓이는 순서를 변경.

**감사합니다**  
**THANK YOU**