

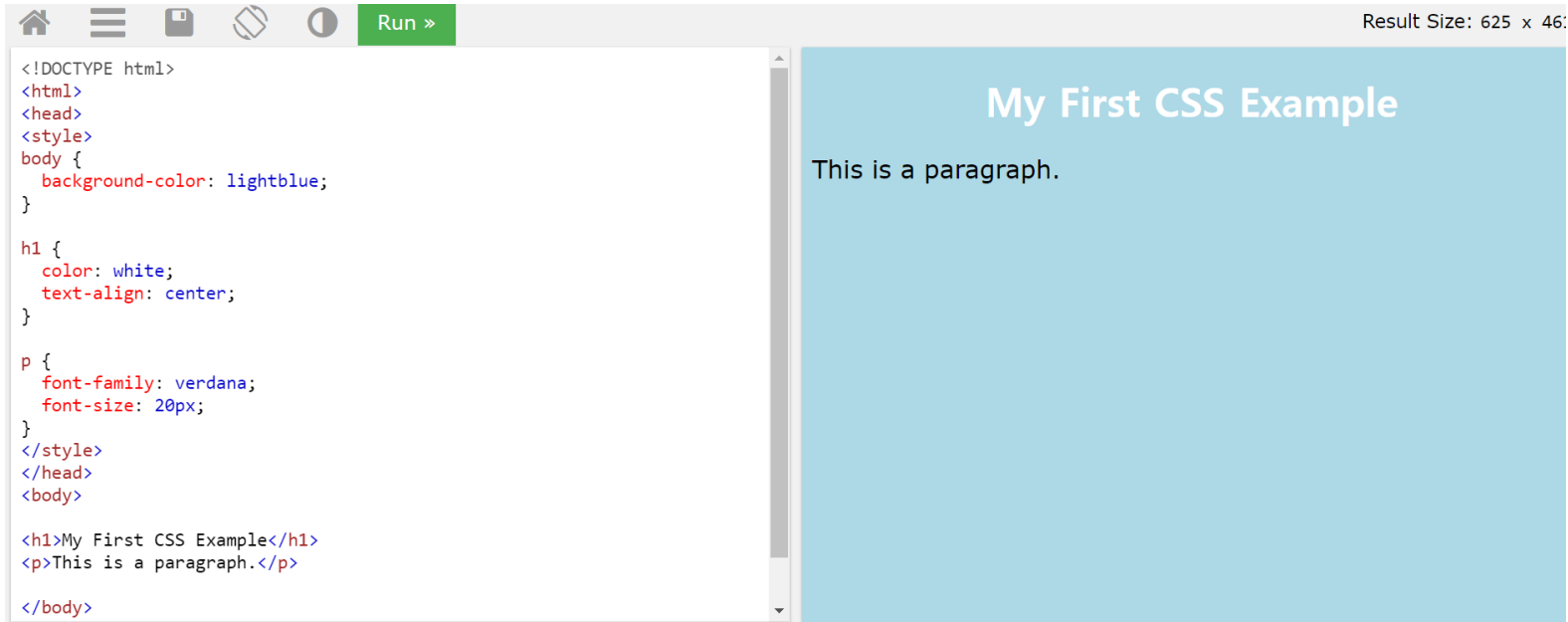
프론트 세미나 Week3

CSS Position&Layout

세미나 Policy!

- 결석, 과제 미제출은 각각 1회까지만 허용됩니다!
- 2회 이상 무단 결석 혹은 2회 이상 과제 미제출 시 세미나를 통과할 수 없습니다!
- 결석을 미리 통보했더라도 녹화본 시청 후 과제를 제출하셔야 출석이 인정됩니다! 과제 미제출 시 무단 결석으로 처리됩니다!

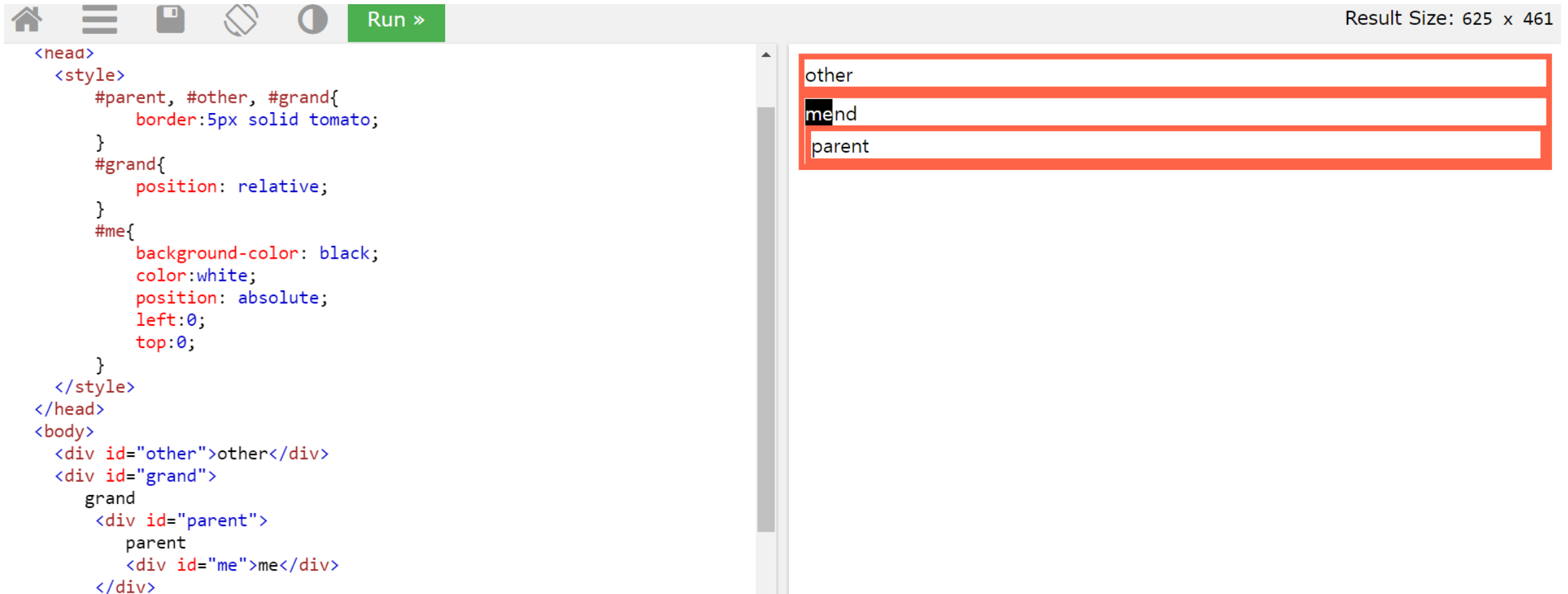
Review



Last lecture: CSS

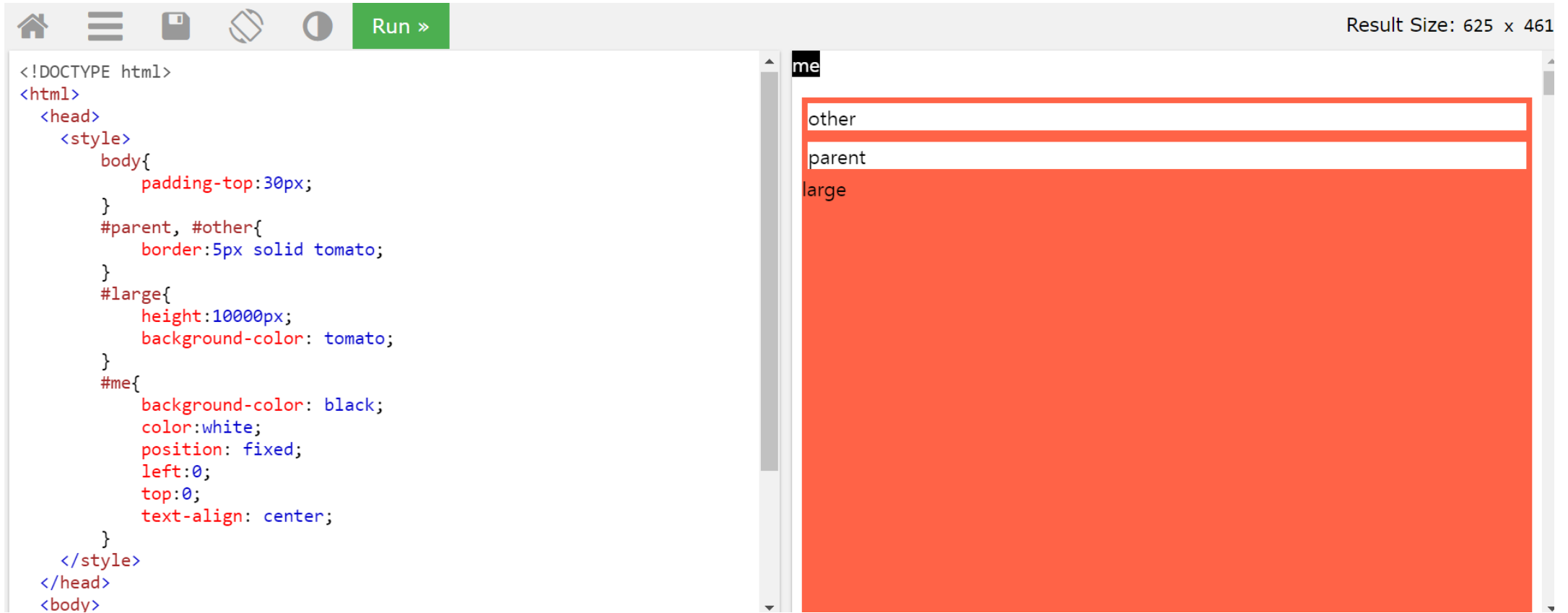
- 디자인을 위한 언어인 CSS
- 각 element들의 위치 지정하는 property 알아보기!(position)
- 각 element들을 조화롭게 배치하는 방법 알아보기!(Layout)

Position



- absolute: 부모 element 중 처음으로 static이 아닌 position을 갖는 것에 따라 결정
- absolute를 쓸 경우 부모-자식 간의 관계가 사라짐(border tomato 사라진 것 확인, contents 크기 자기 크기만큼만 가지므로 width height 직접 지정 필요)
- 부모 element 중 position이 따로 지정된 것이 없다면 html tag 기준(전체)으로 결정

Position

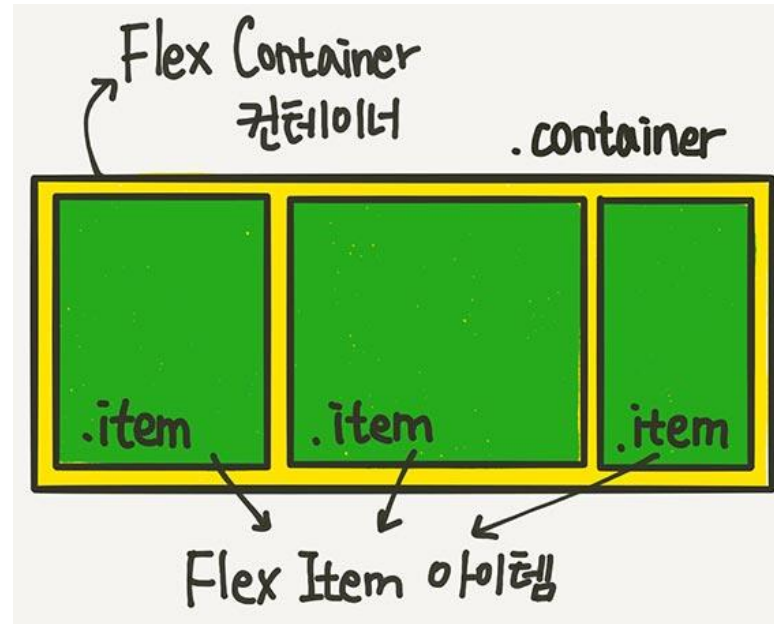


- fixed: 스크롤과 무관하게 배치하고 싶을 때 사용
- absolute와 마찬가지로 부모-자식 간의 관계가 사라짐, contents 크기=자신의 크기

Layout

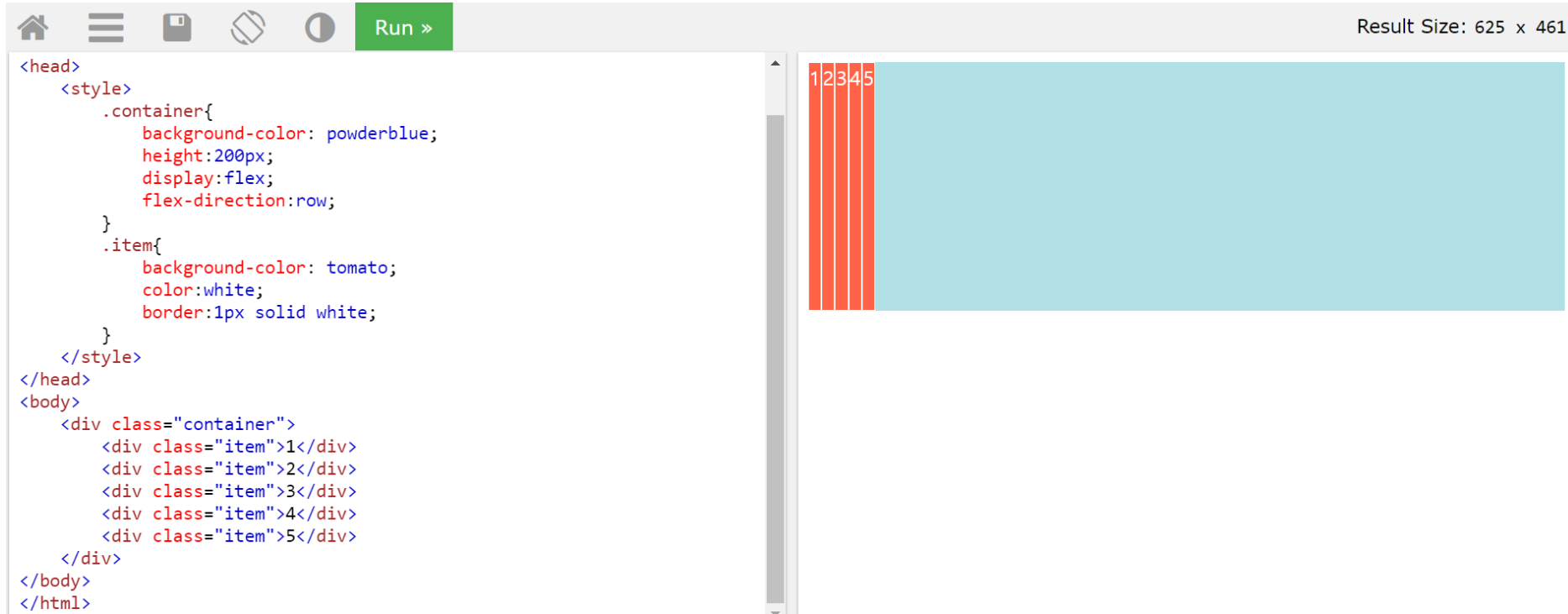
- float과 display, position 등을 적절히 활용하여 element들을 배치할 수는 있지만, 직관적이지 않고 배치를 어떻게 해야할지 많은 고민이 필요
- 이를 개선하기 위한 것이 Layout!
- Flex, Grid

Flex



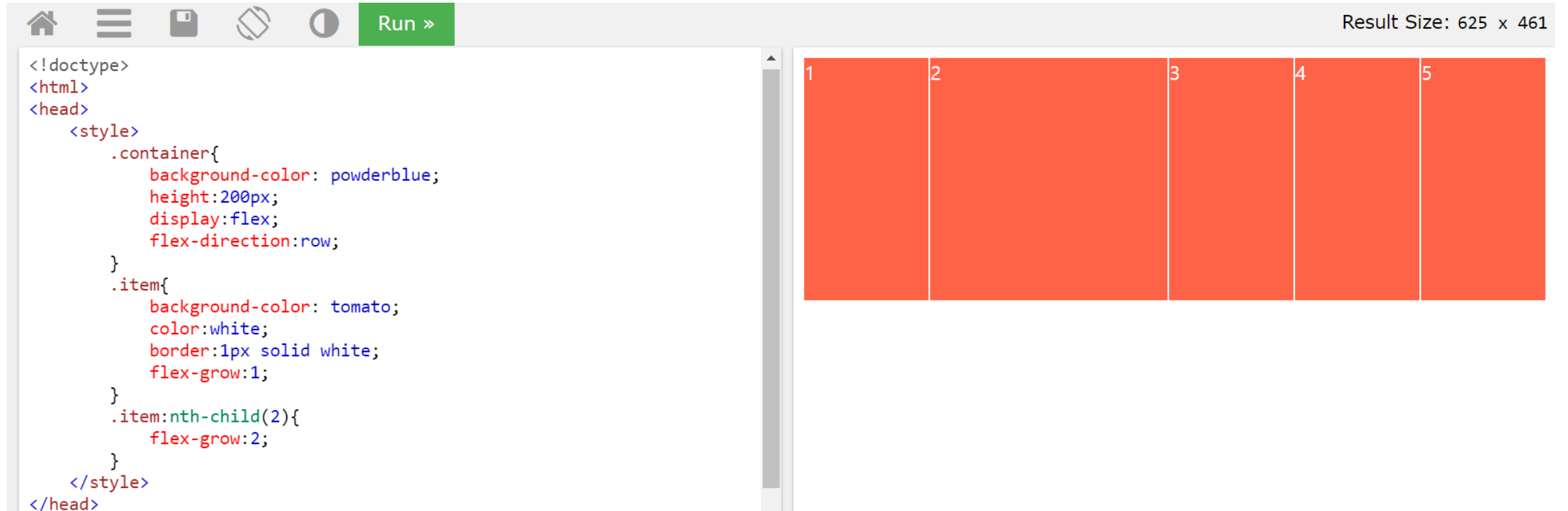
- 각 item들을 감싸는 container 안에 여러 개의 item을 원하는 대로 배치 가능
- flex를 적용하기 위해서는 container의 property로 display:flex를 명시 해주어야 함

Flex



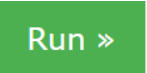





- flex-direction: 각 item들을 어떤 방향으로 정렬할지를 결정
ex) row, row-reverse, column, column-reverse

Flex



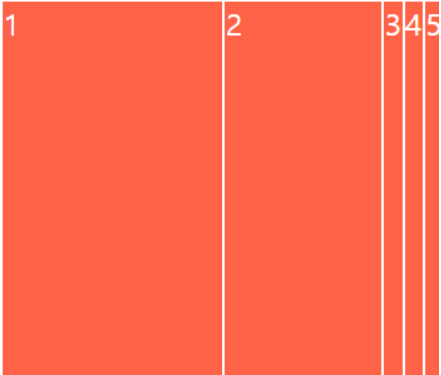
- flex-basis: item의 기본 크기 설정
- flex-grow: 여백을 얼마나 나눠가질지 설정

Flex



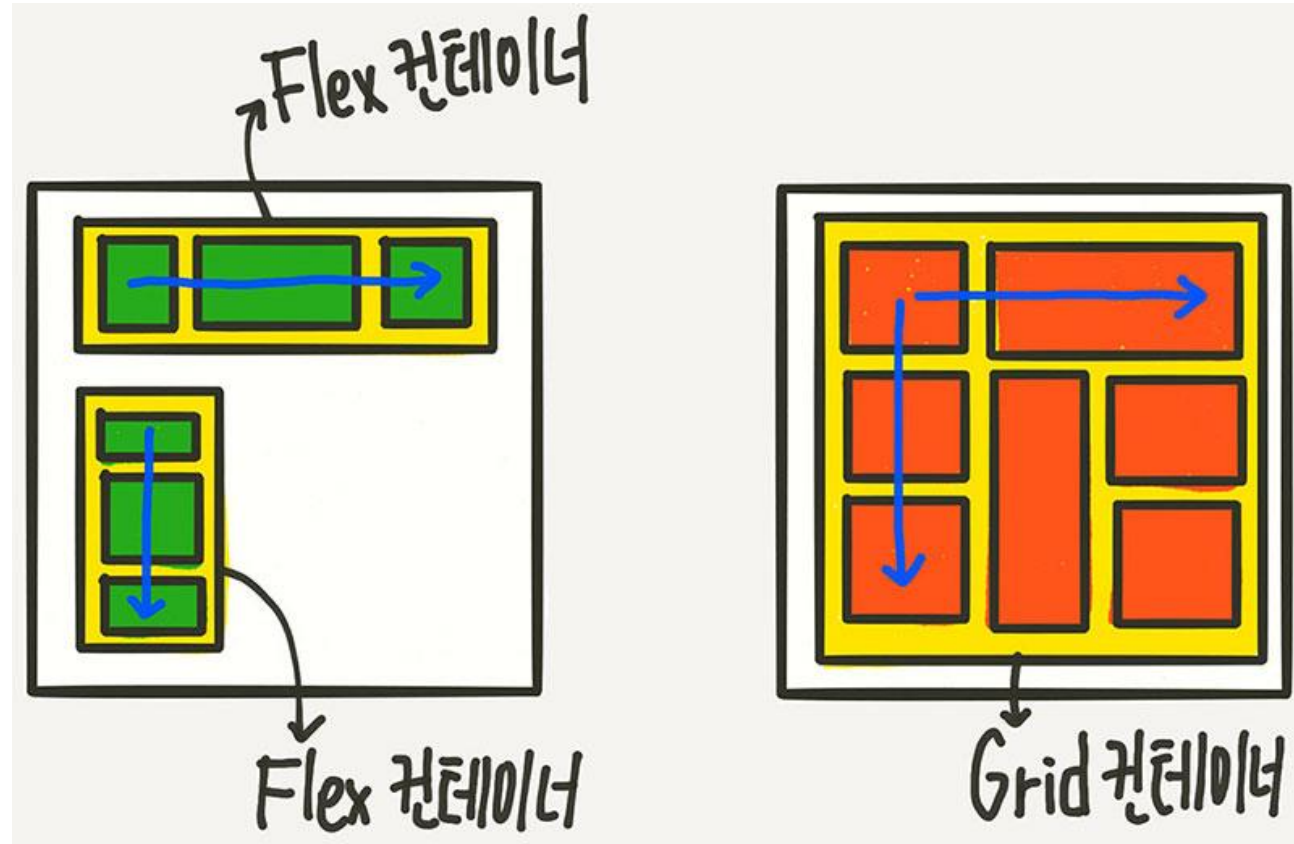
Result Size: 251 x 461

```
<!doctype>
<html>
<head>
  <style>
    .container{
      background-color: powderblue;
      height:200px;
      display:flex;
      flex-direction:row;
    }
    .item{
      background-color: tomato;
      color:white;
      border:1px solid white;
    }
    .item:nth-child(1){
      flex-basis: 150px;
      flex-shrink: 1;
    }
    .item:nth-child(2){
      flex-basis: 150px;
      flex-shrink: 2;
    }
  
```








- flex-shrink: item이 얼마나 줄어들지를 설정

Grid



- flex는 1차원이지만, grid는 2차원으로 더 복잡한 layout 가능!
- container의 property로 display: grid를 사용

Grid



Run »

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto| auto auto;
  background-color: #2196F3;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
</head>
<body>

<h1>Grid Elements</h1>

<p>A Grid Layout must have a parent element with the <em>display</em>
property set to <em>grid</em> or <em>inline-grid</em>.</p>
```

Result Size: 625 x 461

Grid Elements






A Grid Layout must have a parent element with the *display* property set to *grid* or *inline-grid*.

Direct child element(s) of the grid container automatically becomes grid items.

1	2	3
4	5	6
7	8	9

- grid-template-columns/grid-template-rows : grid 형태 지정
- auto: 크기 자동으로 최대 크기만큼
- 직접 픽셀 지정 가능
- fr: 비율 지정, 1fr 2fr 2fr일 경우 1:2:2로 비율 맞춤

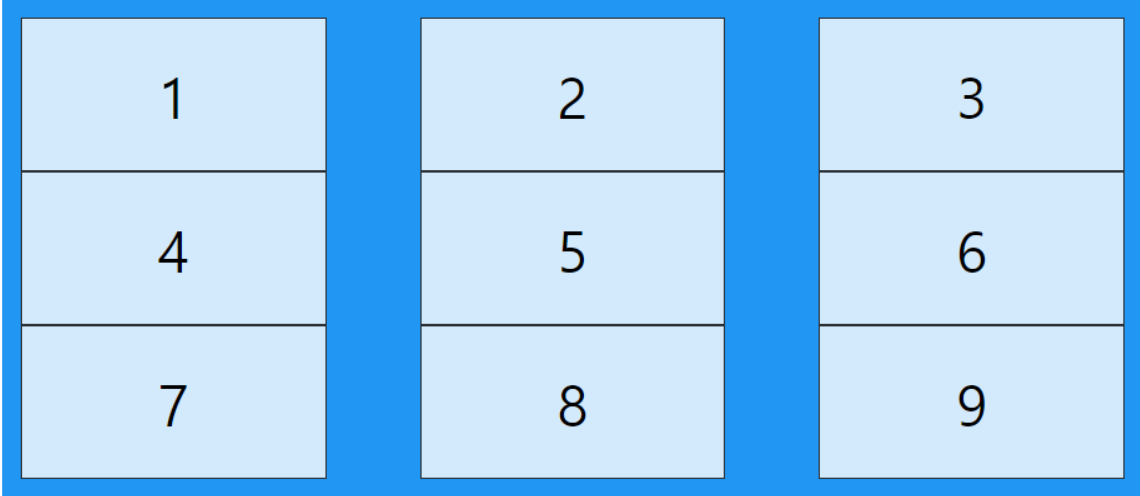
Grid



Run »

Result Size: 625 x 461

The grid-column-gap Property:



Use the *grid-column-gap* property to adjust the space between the columns.

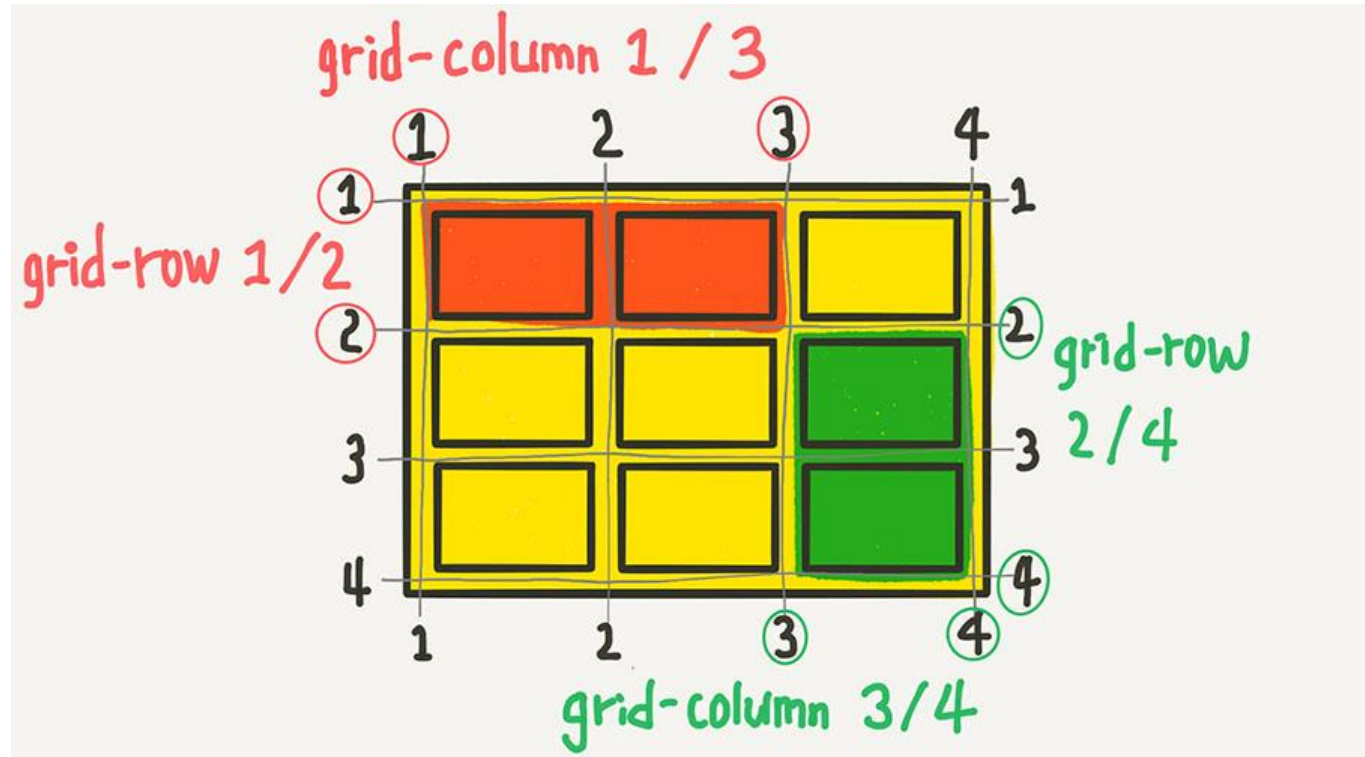
```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-column-gap: 50px;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}

.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
</head>
<body>

<h1>The grid-column-gap Property:</h1>
```

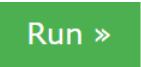





- grid-row-gap/grid-column-gap: 행/열 사이의 간격 지정

Grid



- grid-column-start, grid-column-end, grid-row-start, grid-row-end:
각 item들의 시작 행/열과 끝 행/열을 지정

Grid



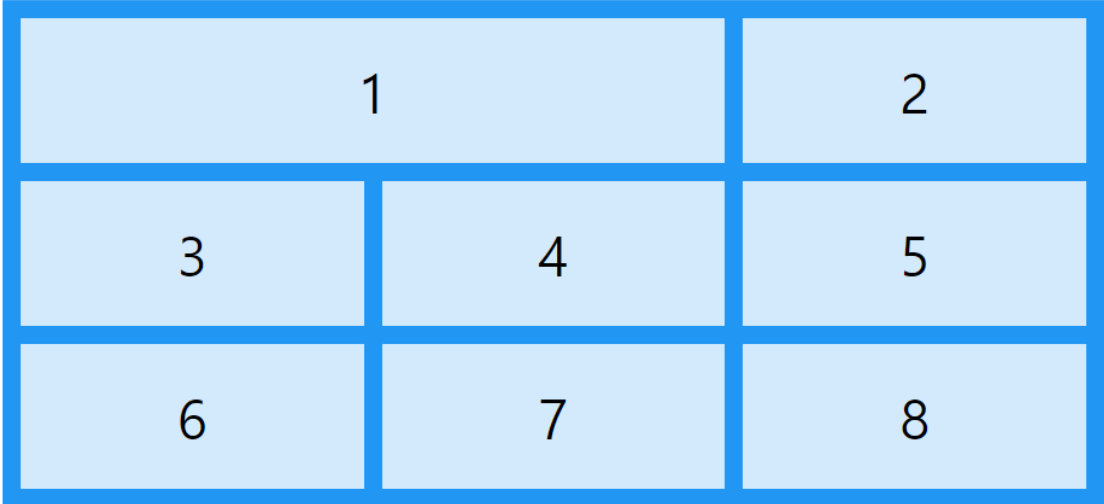
Result Size: 625 x 461

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
  grid-column-start: 1;
  grid-column-end: 3;
}
</style>
```

Grid Lines



You can refer to line numbers when placing grid items.

- grid-column-start, grid-column-end, grid-row-start, grid-row-end:
각 item들의 시작 행/열과 끝 행/열을 지정

Grid

Run »

Result Size: 625 x 46

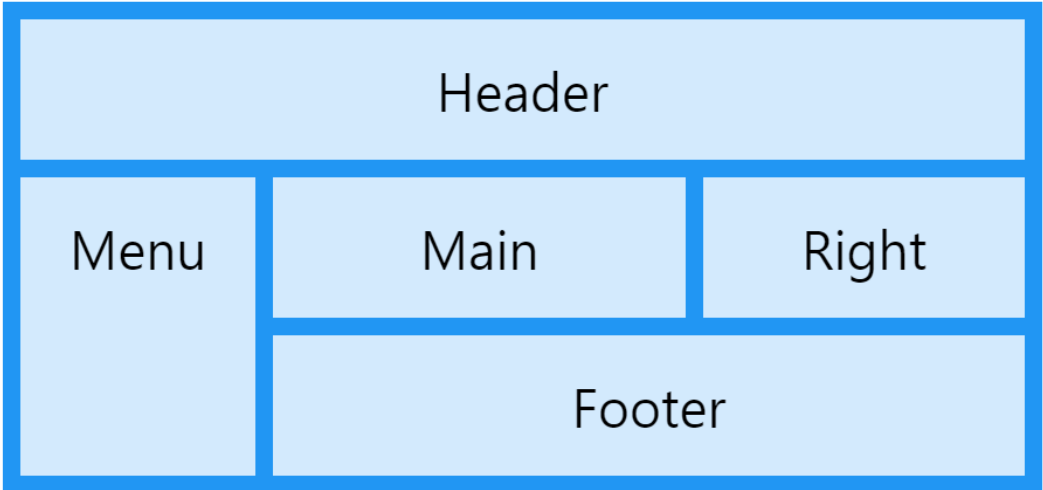
```
.item2 { grid-area: menu; }  
.item3 { grid-area: main; }  
.item4 { grid-area: right; }  
.item5 { grid-area: footer; }  
  
.grid-container {  
  display: grid;  
  grid-template-areas:  
    'header header header header header header'  
    'menu main main main right right'  
    'menu footer footer footer footer footer';  
  grid-gap: 10px;  
  background-color: #2196F3;  
  padding: 10px;  
}  
  
.grid-container > div {  
  background-color: rgba(255, 255, 255, 0.8);  
  text-align: center;  
  padding: 20px 0;  
  font-size: 30px;  
}  
</style>  
</head>  
<body>
```

The grid-area Property

You can use the *grid-area* property to name grid items.

You can refer to the name when you set up the grid layout, by using the *grid-template-areas* property on the grid container.

This grid layout contains six columns and three rows:



- grid-template-areas 이용하는 방법: 아주 직관적임!
- 각 item들이 어떻게 배치될지를 써주기만 하면 완성

Reference

position

- <https://www.opentutorials.org/course/2418/13414>
- https://www.w3schools.com/cssref/pr_class_position.asp

flex

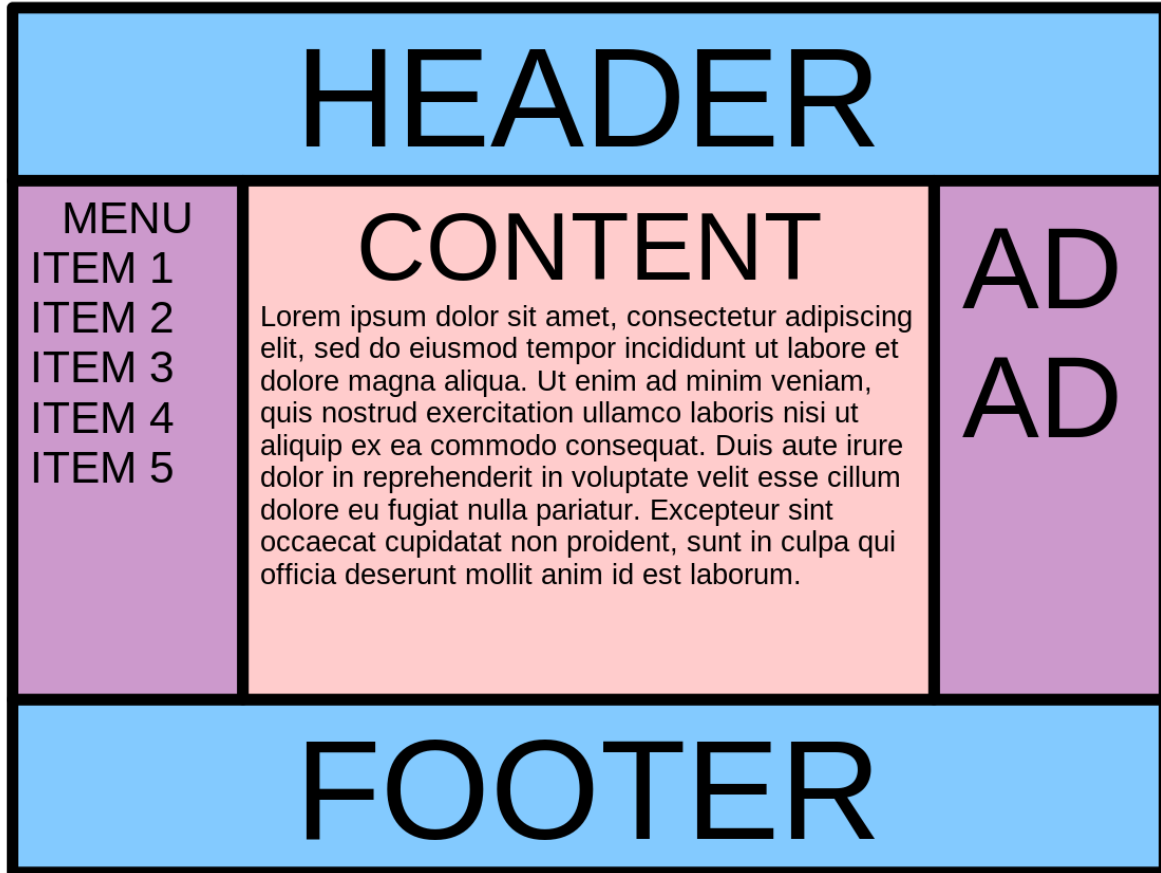
- <https://studiomeal.com/archives/197>
- <https://opentutorials.org/course/2418/13526>
- https://www.w3schools.com/cssref/css3_pr_flex.asp

grid

- <https://studiomeal.com/archives/533>
- <https://opentutorials.org/course/3086/18322>
- https://www.w3schools.com/css/css_grid.asp

- 이외에도 flex, grid와 관련해서는 많은 property들이 존재하므로 구글링해서 찾아보면 도움이 될 것 같습니다!
- position과 flex의 예시 코드는 생활코딩 님의 강좌를 참고하였습니다!

과제



- flex나 grid를 이용하여 간단한 holy grail layout page 작성
- 내용은 자유롭게 구성 가능
- <https://www.youtube.com/watch?v=8DJBCHro-3Q> 의 생활코딩 강좌를 참고하면 도움이 됩니다!
- 화면 캡처 or 녹화 후 카톡으로 보내주세요!
- 기한: 다음 수업 전까지!
- 향후 2주간은 시험 주간이므로 수업이 없을 예정입니다!