

- CSS 는 순차적용이므로, 앞서 먼저 선언한 내용이 있다면 하위에서 지정해도 미적용
- 익스플로러(IE-edge) 호환성

```
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

html

- letter-spacing : 자간

```
letter-spacing: -0.5px;
```

CSS

- display

```
.grid-3-cols {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  column-gap: 80px;
}
```

CSS

- grid : 테이블이 아닌 격자형태의 배열
- repeat(3, 1fr) : 한 페이지에 3개의 객체 생성
- column-gap : 각 객체간 거리

- button

```
<a class="btn btn--big" href="#">Shop chairs</a>
```

html

```
.btn:link,
.btn:visited{
  background-color: #087f5b;
  color: #fff;
  text-decoration: none;
  text-transform: uppercase;
  font-weight: 500px;
  display: inline-block;
  border-radius: 100px;
}

.btn:hover,
.btn:active{
  background-color: #099268;
}
```

CSS

- btn : 버튼에 관련한 약어 # a href(a:link)와 유사한 태그
- color: #fff : 텍스트를 흰색으로 표현하기 위해
- border-radius : 반원형태

- stroke : 외곽선 처리

```
stroke: #087f5b;
```

```
-webkit-text-stroke: 1px red
```

- 이미지나 블록등에 적용
- 텍스트의 경우 -webkit-text 을 적어야 빠른 적용(한글은 자간과 장평의 적용이 어려움)

- shadow : 음영

```
.chair{
  box-shadow: 0 20px 30px 0 rgba(0,0,0, 0.07);
}
```



### The Laid Back

Leisure and relaxing

Comfortable for 4h

Vegan leather

Weighs 16 kg

250€

ADD TO CART



### The Laid Back

Leisure and relaxing

Comfortable for 4h

Vegan leather

Weighs 16 kg

250€

ADD TO CART

- rgba : red(0~255), green(0~255), blue(0~255), alpha(0~1)
- 상우하좌(시계방향) 순서로 적용

- space : 여백

```
justify-content: space-between;
```

- 빈 공간을 똑같이 나눔을 지정

- font-family : 폰트모음집

```
font-family: sans-serif;
```

| Serif : 바탕체 계열  | Sans-Serif : 고딕체 계열 | Monospace : 가로 폭이 동일한 글꼴                                |
|-----------------|---------------------|---|
| Courier         | Arial               | Courier   |
| Courier New     | Arial Black         | Courier New   |
| Georgia         | Tahoma              |   |
| Times           | Trebuchet MS        | <b>cursive : 필기체 계열</b>                                 |
| Times New Roman | Verdana             | Comic Sans(Windows와 Macintosh에서만 지원하며 Linux에서는 지원되지 않음) |

- family 단위로 지정해야 해당 폰트 속성(하위 글꼴)이 전체 적용

- class의 하위 속성에 CSS 지정

- Text : h2에 새로운 class를 지정하지 않고, **.testimonial-box h2**에 작성해도 적용됨

```
<div class="testimonial-box">
  <h2>"We couldn't live without these chairs anymore"</h2>
```

html

```
.testimonial-box {
  grid-column: 2 / -1;
  align-self: center;
}
.testimonial-box h2 {
  margin-bottom: 24px;
  font-size: 24px;
}
```

CSS

- img : 마찬가지로 **.chair img**도 지정 가능

```
<figure class="chair">
  
```

html

```
.chair{
  box-shadow: 0 20px 30px 0 rgba(0,0,0, 0.07);
  border-radius: 12px;
}
.chair img{
  border-bottom-left-radius: 0;
  border-bottom-right-radius: 0;
}
```

CSS

- li : 리스트도 동일하게 **.chair-details** 뒤에 **li**와 **li:last-child**마다 지정 가능

```
<ul class="chair-details">
  <li>
    <span>Leisure and relaxing</span>
  </li>
```

html

```
.chair-details {
  list-style: none;
  margin-bottom: 24px;
}
.chair-details li {
  display: flex;
  align-items: center;
  gap: 12px;
  margin-bottom: 16px;
}
.chair-details li:last-child {
  margin-bottom: 0;
}
```

CSS

- HTML <Head> 에서 CSS 지정

```
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="text/html" />
  <style type="text/css">
    div.container{
      width: 900px;
      margin: 0 auto;
    }
    ul.menubar li{
      display: inline-block;
      list-style-type: none;
      min-width: 6em;
      text-align: center;
      padding: 0.2em;
    }
    ul.menubar li:hover{
      background-color: #ddd;
      cursor: pointer;
    }
    ul.menubar ul{
      position: absolute;
      visibility: hidden;
      margin-top: 2px;
      margin-left: 30px;
      background-color: #ffffff;
      border: 1px solid #ddd;
      padding: 0px;
      box-shadow: 4px 4px 4px #eee;
      min-width: 120px;
    }
  </style>
</head>
```

```
<div class="container">
  <ul class="menubar">
    <li>메뉴1
    <li>메뉴2
    <li>메뉴3
      <ul>
        <li>메뉴3-1</li>
        <li>메뉴3-2</li>
        <li>메뉴3-3</li>
      </ul>
    <li>로그인</li>
  </ul>
</div>
```

- class의 내부 class 지정 : **ul.menubar li** / **ul.menubar ul** / **ul.menubar li**
- em : 해당 폰트의 N 배 (ex; 6em = 6배)
  - font-size 에서 px를 지정시 em을 설정한 곳 들은 해당 배수만큼 적용되어 편리
- 상위 <li> 에 </li> 없도록 주의 (ex; <li>메뉴3 뒤에 </li> 작성시 하위 리스트 미적용됨)
- HTML 이 아닌 CSS 로 이동시 **ul.menubar** -> **.menubar** 변경 가능

- 풀 메뉴 구현(하위/세로)

```
ul.menubar{
  border: 1px solid #ddd;
  border-radius: 0.4em;
  background: linear-gradient(#ffffff, #000000);
  padding: 0.3em 5em;
  font-size: 11px;
}
ul.menubar > li{
  display: inline-block;
  list-style-type: none;
  min-width: 6em;
  text-align: center;
  padding: 0.2em;
}
ul.menubar li.right{
  float: right;
}
ul.menubar li:hover{
  background-color: #ddd;
  cursor: pointer;
}
ul.menubar ul{
  position: absolute;
  visibility: hidden;
  margin-top: 2px;
  margin-left: 30px;
  background-color: #ffffff;
  border: 1px solid #ddd;
  padding: 0px;
  box-shadow: 4px 4px 4px #eee;
  min-width: 120px;
}
ul.menubar li:hover ul{
  visibility: visible;
}
ul.menubar ul li{
  display: block;
  text-align: left;
  list-style-type: none;
  min-width: 8em;
  padding: 0.4em;
}
ul.menubar li.seperator{
  height: 1px;
  padding: 0px;
  margin: 4px 0;
  border-top: 1px solid #ccc;
}
```

- ul.menubar ul 에서는 Visibility 를 hidden 적용
  - hover 전에는 하위 메뉴 미등장
- ul.menubar li:hover ul 과 ul.menubar ul:hover 는 Visibility : visible이 적용되어 확인됨



- linear-gradient : 그라데이션 적용
- ul menubar > li : **greater-than sign** 으로 인하여 하위는 미적용, 해당 줄만 적용
- seperator : 구분선 생성

- [display: inline-block](#)

- display : 요소의 크기 결정하며 none, block, inline, inline-block 의 속성값을 가짐
- inline : 전후 줄바꿈 미적용, 한 줄에 다른 요소들과 나란히 배치됨

```

before
<a>A</a>
<span>SPAN</span>
<em>EM</em>
after
      span {
        background: yellow;
        width: 200px;
        height: 50px;
        margin: 20px;
        padding: 10px;
      }
          
```

- ex) <span>, <a>, <em> 등과 유사
- 해당 태그는 마크업하고 있는 콘텐츠의 크기 만큼만 공간을 차지하게 됨
  - 즉, width 와 height 속성을 무시해되 됨
  - 또한, margin 과 padding 속성은 좌우 간격만 반영, 상하는 미반영

- block : 전후 줄바꿈이 적용, 다른 요소들을 다른 줄로 밀어내고 한 줄을 차지

```

before
<h1>H1</h1>
<div>DIV</div>
<p>P</p>
after
div {
  background: yellow;
  width: 200px;
  height: 50px;
  margin: 20px;
  padding: 10px;
}
          
```

- ex) <div>, <p>, <h1>, <br> 등과 유사, 특히 <div>가 갖는 기본값과 동일
- inline과 다르게 width, height, margin, padding 속성이 반영됨

- inline-block : 전후 줄바꿈 미적용, width, height, margin, padding 속성이 반영

```

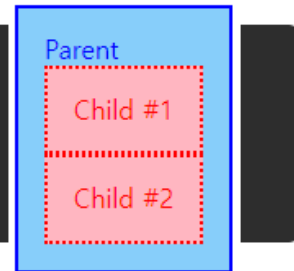
before
<a>A</a>
<span>SPAN</span>
<em>EM</em>
after
span {
  display: inline-block;
  background: yellow;
  width: 200px;
  height: 50px;
  margin: 20px;
  padding: 10px;
}
          
```

- ex) <button>, <input>, <select> 등과 유사
- 내부적으로는 block 요소의 규칙과 외부적으로는 inline 요소의 규칙을 받는 것
- 여러 개의 요소를 한 줄에 정확히 원하는 너비만큼 배치할 수 있기에 유용

- [position : absolute](#)

- position : 요소(elements)가 브라우저 화면에 어떻게 배치되는가를 결정
- absolute
  - 부모 요소에 속박되지 않으며 독립된 배치 문맥을 가짐(position context),
  - 해당 요소를 브라우저 화면(viewport)상에서 원하는 위치에 자유롭게 배치
  - 부모 요소 위에 겹쳐서 배치도 가능
  - 단, 상위 요소중 position: relative인 요소가 있다면, 그 중 가장 가까운 요소의 내부에서만 요소를 자유롭게 배치 가능. 즉, 전체 화면이 아닌 해당 상위 요소를 기준으로 offset 속성(top, left, bottom, right)이 적용됨
- 하나의 Parent(상위요소) 와 두 개의 Child(하위요소)가 있을 경우

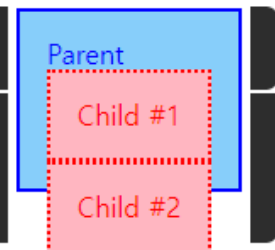
```
<div class="parent">
  Parent
  <div class="child">Child #1</div>
  <div class="child">Child #2</div>
</div>
```



- 하나의 Child 에 absolute 지정시

```
<div class="child abs">Child #2</div>
```

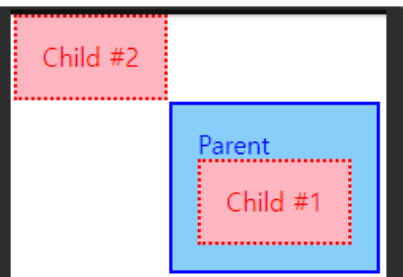
```
.abs {
  position: absolute;
}
```



- 테두리 밖으로는 밀어내지지만, offset 속성을 명시하지 않았을 경우, 기본값 auto(=Position: Static) 가 적용되어 있는 경우와 동일

- Offset(TRBL) 속성을 지정할 경우

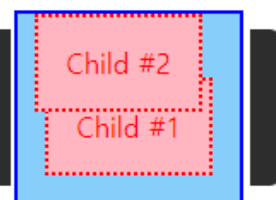
```
.abs {
  position: absolute;
  top: auto;
  left: auto;
  bottom: auto;
  right: auto;
}
```



- Parent(상위요소)로부터 독립(absolute)된 상태가 적용

- Position : relative 가 존재시

```
<div class="parent" style="position: relative">
  <!-- 생략 -->
</div>
```



- Position : relative 의 상위요소인 Parent 내부로 귀속됨
- 즉, positioning context가 전체 화면(viewport)에서 부모 요소로 변경