

- 반응형 웹 지정

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- 속성 지정

```
<div class="col-lg-6 col-md-8 col-sm-10 col-12 mx-auto">

```

- mx, my, mt: margin(외부)의 x 축, y 축, t 축, d 축
- px, py: padding(내부)의 x 축, y 축
- col-lg-6 col-md-8 col-sm-10 col-12
 - col-lg-6: 전체와 main 의 비율이 6:6 으로 절반
 - large(>=992px)는 6을 적용
 - md 는 8을 적용
- mx-auto: 반응형 지정
- img-fluid: 반응형 지정
- [Grid system](#): 반응형 웹

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
.container	100%	540px	720px	960px	1140px	1320px
.container-sm	100%	540px	720px	960px	1140px	1320px
.container-md	100%	100%	720px	960px	1140px	1320px
.container-lg	100%	100%	100%	960px	1140px	1320px
.container-xl	100%	100%	100%	100%	1140px	1320px
.container-xxl	100%	100%	100%	100%	100%	1320px
.container-fluid	100%	100%	100%	100%	100%	100%

[bootstrap - container](#) / [Bootstrap grid examples](#) 참고

- Animation Effect

- [-webkit-animation](#)
- [keyframes](#): CSS 애니메이션에서 구간을 정하고 각 구간별로 어떤 스타일을 적용시킬지 정하는 문법
- [cross-fade](#)
- 크로스 브라우징 이슈를 해결하기 위해 접두사를 붙임
 - 크롬&사파리(apple): -webkit-
 - 파이어폭스: -moz-
 - 오페라: -o-
 - 인터넷 익스플로러: -ms-

- 자바스크립트는 자료형이 없기 때문에 array 로 직접 넣어야 함

- `Document.createElement()`
 - HTML 문서에서, `Document.createElement()` 메서드는 지정한 `tagName`의 HTML 요소를 만들어 반환합니다.
 - `tagName`을 인식할 수 없으면 [HTMLUnknownElement \(en-US\)](#)를 대신 반환합니다.
- `Node.appendChild()` : 한 노드를 특정 부모 노드의 자식 노드 리스트 중 마지막 자식으로 붙입니다.

```
function addAnswer(answerText){
    var a = document.querySelector('.answerBox');
    var answer = document.createElement('button');
    answer.classList.add('answerList');
    answer.classList.add('my-3');
    answer.classList.add('py-3');
    answer.classList.add('mx-auto');
    answer.classList.add('fadeIn');
    a.appendChild(answer);
    answer.innerHTML = answerText;
}
```

- 1번 문제가 부모, 선택지들이 자식

- `disable=true`
 - 클릭시 사라지는 상태가 true

```
answer.addEventListener("click", function(){
    var children = document.querySelectorAll('.answerList');
    for(let i=0; i<children.length; i++){
        children[i].disable=true;
        children[i].style.webkitAnimation="fadeOut 0.5s";
        children[i].style.Animation="fadeOut 0.5s";
    }
})
```

◦

- [ColorZilla](#) : 색상에 그라데이션 부여
- [Google Fonts](#) : 우측 상단 쇼핑백 클릭

- 자주 쓰이는 이벤트들

이벤트명 (표준 모델 호출)	인라인 호출 시	발생 시점
keydown	onKeyDown	키가 눌렸을 때
click	onClick	요소가 클릭되었을 때
doubleclick	onDoubleClick	요소가 더블 클릭되었을 때
mouseover	onMouseover	요소 위에 마우스 포인터가 올라왔을 때
mouseleave	onMouseleave	요소로부터 마우스 포인터가 벗어났을 때
change	onChange	요소의 값이 변경될 때
input	onInput	사용자가 요소에 값을 입력할 때
submit	onSubmit	form 요소의 제출 버튼이 클릭되었을 때
reset	onReset	form 요소의 리셋 버튼이 클릭되었을 때

- keydown

```
<body>
  <p>key down <input id="keydown"></p>
</body>
<script>
  document.addEventListener("DOMContentLoaded",
  function (e){
    documentElement.addEventListener("keydown",
    function(e){console.log(e.key)
      if (e.key === "q"){alert("q pushed")}})})
</script>
```

DOMContentLoaded = onload

누가 : (e) 란
어디에 : keydown이란 곳에
무엇을 : e.key를 출력하는데
어떻게 : e.key가 q 면 메시지 출력

- Mousover & mouseleave

```
<body>
  <button id="btn" onmouseover="red(event)"
    onmouseleave="blue(event)">HERE!</button>
</body>
<script>
  function red(){
    btn.style.backgroundColor = "red"}
  function blue(e){
    e.target.style.backgroundColor = "blue"}
</script>
```

누가 : red(e) 란
어디에 : e의 target 인 button에
무엇을 : 배경색을 re로

- red 는 id를 통해 {기능}에서 지정했고, blue 는 파라미터 e를 통해 e.target 으로 지정

- setAttribute : 해당 속성을 변경

- **textContent** : 태그의 글자 변경, 현재 기재된 값을 가져오거나 새로운 값 할당

```
<body>
  <p>HTML 글자 출력</p>
  <input type="button" value="click">
</body>
<script>
  document.addEventListener("DOMContentLoaded",
    function(e){
      let a = document.querySelector("input")
      let b = document.querySelector("p")
      a.addEventListener("click",
        function(e){
          b.textContent = "HTML 글자 변경"}}))
</script>
```

누가 : 파라미터(e) 란 = 이벤트발생

a 는 input 태그를 지칭
b 는 p 태그를 지칭

input 태그가 클릭, 되면
p 태그의 textContent 를 변경

- **innerHTML** : HTML내 속성 변경, 현재 기재된 값을 가져오거나 새로운 값 할당
- [JavaScript Reserved Words](#)

abstract	arguments	await*	boolean
break	byte	case	catch
char	class*	const	continue
debugger	default	delete	do
double	else	enum*	eval
export*	extends*	false	final
finally	float	for	function
goto	if	implements	import*
in	instanceof	int	interface
let*	long	native	new
null	package	private	protected
public	return	short	static
super*	switch	synchronized	this
throw	throws	transient	true
try	typeof	var	void
volatile	while	with	yield