

1) createElement()

- 요소 생성 : HTML 문서에서 지정한 태그의 요소 생성, 위치 미지정 상태이므로 보이지 않음
 - 구조 : 변수 = 요소를 생성한다 '특정 태그를'

```
var answer = document.createElement('button');
```

- 요소 추가 : 새로운 요소를 자식 요소로 추가하는 것
 - 구조 : 부모 노드의 요소명.appendChild(자식요소)

```
a.appendChild(answer);
```

- appendchild : 특정 부모 노드의 자식 노드 리스트 중 마지막 자식으로 추가

- 자식 요소값 설정 : 자식요소의 메서드를 수행하거나 속성 값 설정
 - 구조 : 자식요소.메서드 혹은 자식요소.속성 = 속성 값

```
answer.innerHTML = answerText;
```

```
window.onload = function() {
  document.body.appendChild(newElement);
  // create element
  const newElement = document.createElement("div");

  // input text
  newElement.innerHTML = "add body child : world!"

  // append child element, parent : body
  document.body.appendChild(newElement);
}
```

전체 구조 예시

- classList : 자바스크립트에서 HTML 요소의 Class 추가, 제거 등, element의 읽기 전용 속성
 - 속성 : add, remove, contains(요소에 특정 클래스가 포함되어 있는지 테스트), toggle(미속성값시 추가, 추가 제거), replace(이름변경)
 - HTML 에서 class 를 여러개 늘려도 되지만 JS 에서 늘리는 방식을 채용
 - ('my-3 , py-3 , mx-auto') : 한 줄로 작성가능

```
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;
```

addAnswer에게 answerText 전달
a 는 answerBox 태그를 지칭
answer 는 button 태그를 지칭

answer의 class 추가

a의 자식요소로 answer 요소 추가
(answerBox 부모, answer 자식)
answer 를 answerText 로 지정

2) createElement 상세 정리

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

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

        setTimeout(()=>{
            for(let i=0; i<children.length; i++){
                children[i].style.display='none';
            }
            goNext(++qIdx);}, 400)
    }, false);
}
```

전체 문단

- 요소 생성(class의 선언) 및 속성 부여
 - `answerList my-3 py-3 mx-auto fadeIn` 라는 속성을 주목하자

```
<button type="button" class="answerList my-3 py-3 mx-auto fadeIn"
onclick="js:begin()">시작하기</button>
```

html

```
var answer = document.createElement('button');
answer.classList.add('answerList', 'my-3', 'py-3', 'mx-auto', 'fadeIn');
```

js

- 둘의 기능적 차이는 없다.
- 다만, HTML 의 코드를 js에서 구현하기 위해서이다.

- 요소의(선언된 class) 의 호출 : `document.querySelectorAll('.answerList')`

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

js

- 앞서 button 에 createElement 라는 요소를 생성했다.
- querySelectorAll 은 answerList 이 포함된 모든 곳에 적용된다는 의미이다.
 - 그러나, button이 두개라면 첫번째 button 에만 적용되는 것에 주의
 - answerList 가 지정된 곳에 해당 for 문이 전부 적용된다.

- querySelectorAll 의 활용
 - for 문 or 파라미터를 여러개 지정하는 용도로 사용

```
const sections = document.querySelector("#sections , #sections .section"js)
console.log(sections.constructor.name);
for (const i = 0; i < sections.length; i++) {
  const item = sections.item(i);
  item.style.border = "1px solid #ff0000";}
```