

07. Array 객체

Array 객체란

Array 객체로 배열 만들기

```
> var numbers = ["one", "two", "three", "four"]; // 리터럴을 사용한 배열
> var numbers = new Array("one", "two", "three", "four"); // Array 객체를 사용한 배열
```

배열에서 for문 사용하기

```
> var seasons = ["봄", "여름", "가을", "겨울"]
> seasons.length
< 4
```

```
> for (var i = 0; i < seasons.length; i++) {
    console.log(seasons[i]);
}
봄
여름
가을
겨울
```

Array 객체란

Array 객체 : 배열에 활용하는 많은 속성과 함수가 미리 정의되어 있는 객체

Array 객체로 배열 만들기

초깃값이 없을 때

```
> var myArray = new Array( ); // Array 객체의 인스턴스를 만듭니다.
```

초깃값이 있을 때

```
> var numbers = ["one", "two", "three", "four"]; //리터럴을 사용한 배열  
> var numbers = new Array("one", "two", "three", "four"); //Array 객체를 사용한 배열
```

Array 객체의 주요 함수

둘 이상의 배열을 연결하는 concat() 함수

- 기존의 배열에 또 다른 배열이나 값을 합쳐서 새로운 배열을 만드는 함수
- concat() 함수는 새로운 배열을 만들기 때문에 기존의 nums나 chars 배열에는 영향을 주지 않음

```
> var nums = ["1", "2", "3"];  
> var chars = ["a", "b", "c", "d"];
```

```
> nums.concat(chars) //nums 배열에 chars 배열 연결  
< ▶ (7) ["1", "2", "3", "a", "b", "c", "d"]  
      └──┬──┘ └──┬──┘  
      nums  chars
```

```
> chars.concat(nums) //chars 배열에 nums 배열 연결  
< ▶ (7) ["a", "b", "c", "d", "1", "2", "3"]  
      └──┬──┘ └──┬──┘  
      chars  nums
```

```
> var nums = [1, 2, 3]  
< undefined  
  
> var chars = ["a", "b", "c", "d"]  
< undefined  
  
> nums.concat(chars)  
< ▶ (7) [1, 2, 3, 'a', 'b', 'c', 'd']  
  
> chars.concat(nums)  
< ▶ (7) ['a', 'b', 'c', 'd', 1, 2, 3]  
  
> nums  
< ▶ (3) [1, 2, 3]  
  
> chars  
< ▶ (4) ['a', 'b', 'c', 'd']
```

Array 객체의 주요 함수

배열 요소를 연결하는 join() 함수

- 지정한 구분 기호를 사용해 배열 요소 연결
- 구분 기호를 지정하지 않으면 쉼표(,)로 구분

```
> nums.join( )    //구분 기호 없이 연결
< "1,2,3"
> nums.join("-")  //구분 기호("-")를 사용해 연결
< "1-2-3"
```

```
> nums.join()
```

```
< '1,2,3'
```

```
> nums.join("-")
```

```
< '1-2-3'
```

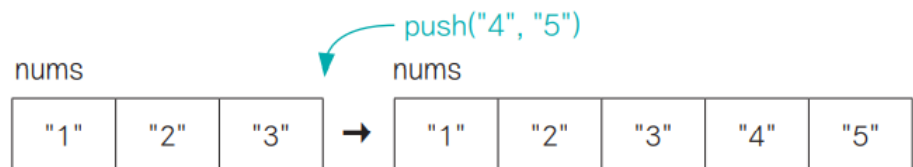
Array 객체의 주요 함수

새로운 요소를 추가하는 push() 함수와 unshift() 함수

- push() 함수 : 배열의 맨 끝에 요소 추가
- unshift() 함수 : 배열의 맨 앞에 요소 추가
- 새 요소가 추가된 후의 요소 개수가 반환됨

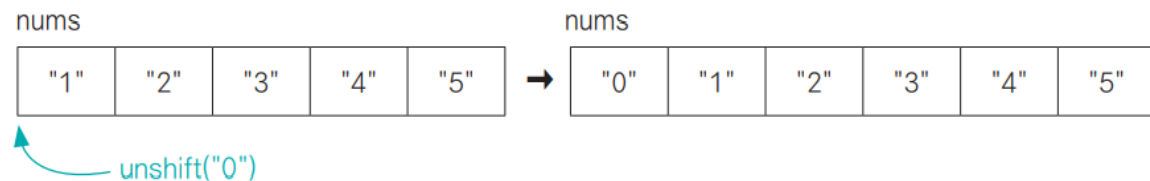
```
> var nums = ["1", "2", "3"]  
> nums.push("4", "5") // nums 배열 맨 끝에 "4"와 "5" 요소 추가  
< 5 // 새 요소가 추가된 후의 배열 요소의 개수가 반환됨
```

```
> nums  
< ▶ (5) ["1", "2", "3", "4", "5"]
```



```
> nums.unshift("0") // nums 배열 맨 앞에 "0" 요소 추가  
< 6 // 새 요소가 추가된 후의 배열 요소의 개수가 반환됨
```

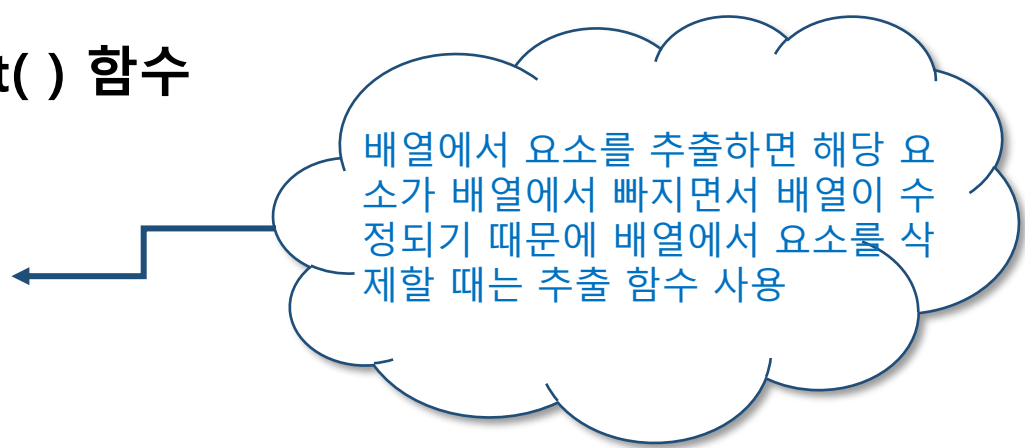
```
> nums  
< ▶ (6) ["0", "1", "2", "3", "4", "5"]
```



Array 객체의 주요 함수

배열에서 요소를 추출하는 pop() 함수와 shift() 함수

- pop() 함수 : 배열의 맨 뒤에 있는 요소 추출
- shift() 함수 : 배열의 맨 앞에 있는 요소 추출
- 추출한 요소가 반환됨



배열에서 요소를 추출하면 해당 요소가 배열에서 빠지면서 배열이 수정되기 때문에 배열에서 요소를 삭제할 때는 추출 함수 사용

```
> var study = ["html", "css", "javascript"]
< study.pop( )
> "javascript"
< study
> ▶ (2) ["html", "css"]
```

```
> var js = ["es6+", "node", "react", "angular", "vue"]
> js.shift( )
< "es6+"
```

Array 객체의 주요 함수

원하는 위치의 요소를 삭제/추가하는 splice() 함수

- 괄호 안에 들어 있는 인수에 따라 일정 구간의 요소를 삭제하고 새로운 요소를 추가하는 함수
- splice() 함수를 실행하면 삭제한 구간의 요소들로 이루어진 새로운 배열이 곱셈으로 표시됨.

1) 인수 1개 : 인수가 가리키는 인덱스의 요소부터 배열의 끝 요소까지 삭제

```
> var numbers = [0, 1, 2, 3, 4, 5]
> numbers.splice(2) //인덱스 2(세 번째 요소) 이후 끝까지 삭제
< ▶ (4) [2, 3, 4, 5] //삭제된 요소로 이루어진 배열
> numbers
< ▶ (2) [0, 1] //수정된 원래 배열
```


Array 객체의 주요 함수

원하는 위치의 요소를 삭제/추가하는 splice() 함수

2) 인수 2개 : 첫번째 인수는 인덱스이고 두번째 인수는 삭제할 개수

```
> var study = ["html", "css", "web", "jquery"]
> study.splice(2,1) //인덱스 2에서 한 개 삭제
< ▶ ["web"] //삭제된 요소로 이루어진 배열
> study
< ▶ (3) ["html", "css", "jquery"] //수정된 원래 배열
```

Array 객체의 주요 함수

원하는 위치의 요소를 삭제/추가하는 splice() 함수

3) 인수 3개 : 첫 번째 인수는 해당 배열에서 삭제를 시작할 위치, 두 번째 인수는 삭제할 개수, 제한 위치에 새로 추가할 요소 . 기존 배열의 요소를 삭제하지 않고 새로운 요소를 추가하고 싶다면 삭제할 개수를 지정하는 두 번째 인수에 0을 넣는다.

```
> study.splice(2, 1, "js")
< ▶ ["jquery"] //인덱스 2에서 1개 삭제
> study
< ▶ (3) ["html", "css", "js"] //삭제한 자리에 새로운 요소를 추가
```

```
> var study = ["html", "css", "web", "jquery"]
< undefined
> study.splice (1, 2, "hello")
< ▶ (2) ['css', 'web']
> study
< ▶ (3) ['html', 'hello', 'jquery']
```

```
> study.splice(2, 0, "jquery")
< ▶ [] //삭제한 요소가 없기 때문에 실행 결과는 빈 배열
> study
< ▶ (3) ["html", "css", "jquery", "js"] //새로운 요소 추가
```

```
> study = ["html", "hello", "jquery"]
< ▶ (3) ['html', 'hello', 'jquery']
> study.splice(1, 0, "css", "javascript")
< ▶ []
> study
< ▶ (5) ['html', 'css', 'javascript', 'hello', 'jquery']
```

Array 객체의 주요 함수

원하는 위치의 요소들을 추출하는 slice() 함수

- 여러 개의 요소를 추출하는 함수

- 1) 시작 인덱스만 지정할 경우 시작 인덱스부터 끝까지 모두 추출
- 2) 시작 인덱스와 끝 인덱스를 지정하면 시작 인덱스부터 끝 인덱스 직전(n-1)까지 추출

```
> var colors = ["red", "green", "blue", "white", "black"]
> colors.slice(2)
< ▶ (3) ["blue", "white", "black"] //인덱스 2부터 끝까지 추출
> colors
< ▶ (5) ["red", "green", "blue", "white", "black"] //원래 배열은 변경되지 않음
```

```
> var colors2 = colors.slice(1, 4)
> colors2
< ▶ (3) ["green", "blue", "white"]
> colors
< ▶ (5) ["red", "green", "blue", "white", "black"]
```

[미리보기] 여행 준비물 점검 프로그램

여행 준비물 점검 목록

충전 케이블	추가
필기 도구	X
노트북	X

여행 준비물 점검 목록

	추가
필기 도구	X
노트북	X
충전 케이블	X

여행 준비물 점검 목록

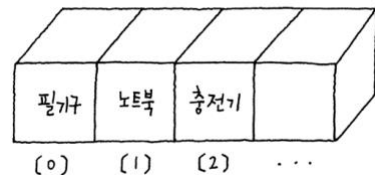
	추가
필기 도구	X
노트북	X
충전 케이블	X

여행 준비물 점검 목록

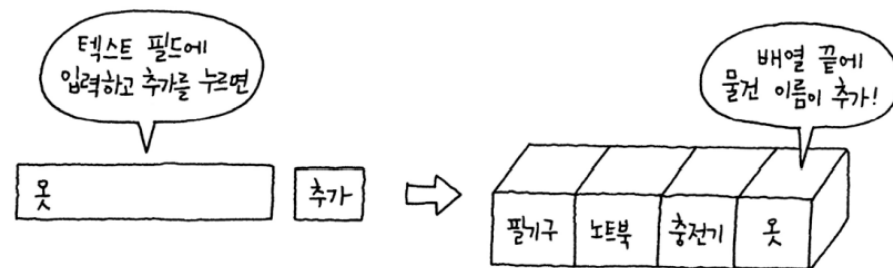
	추가
필기 도구	X
충전 케이블	X

생각해 보기

1) 배열 만들기

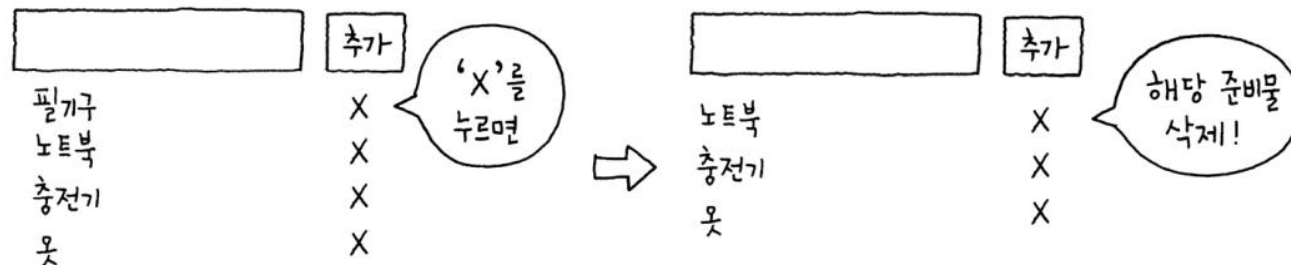


2) 창길 물건 배열에 추가하기 - 텍스트 필드 사용



3) 추가한 내용, 목록 형태로 화면에 표시하기 - for문 사용

4) 창겼다면 목록에서 삭제하기



입력 필드 만들기

실습 파일 : 07Winindex.html

```
7 <link rel="stylesheet" href="css/input.css">
```

```
9 <div id="wrapper">
10   <h2>여행 준비물 점검 목록</h2> //제목
11   <form>
12     <input type="text" id="item" autofocus="true"> //입력용 텍스트 필드
13     <button type="button" id="add" class="addBtn">추가</button>
14   </form>
15   <div id="itemList"></div> //준비물 목록이 표시될 영역
16 </div>
```

항목 추가하기 & 표시하기

실습 파일 :
07WjsWcheckboxlist.js

```
1  var itemList = [];  
2  
3  var addBtn = document.querySelector("#add"); // '추가' 버튼  
4  addBtn.addEventListener("click", addList); // addBtn을 클릭하면 addList 함수 실행  
5  
6  function addList() {  
7      var item = document.querySelector("#item").value; // 텍스트 필드 내용 가져옴  
8      if (item != null) {  
9          itemList.push(item); // itemList 배열의 끝에 item 변수 값 추가  
10         document.querySelector("#item").value = ""; // id="item"인 요소의 값을 지움  
11         document.querySelector("#item").focus(); // 텍스트 필드에 focus( ) 메서드 적용  
12     }  
13     showList(); // 목록을 표시하는 showList( ) 함수 실행  
14 }  
15  
16 function showList() {  
17     var list = "<ul>"; // 목록을 시작하는 <ul> 태그 저장  
18     for (var i=0; i<itemList.length; i++) { // 배열 요소마다 반복  
19         list += "<li>" + itemList[i] + "</li>"; // 요소를 <li>~</li>로 묶음  
20     }  
21     list += "</ul>"; // 목록을 끝내는 </ul> 태그 저장  
22  
23     document.querySelector('#itemList').innerHTML = list; // list 내용 표시  
24 }
```

클릭하면 항목 삭제하기

실습 파일 :
07WjsWcheckboxlist.js

```
1  var itemList = [];  
2  
3  var addBtn = document.querySelector("#add");  
4  addBtn.addEventListener("click", addList); // addBtn.onclick = addList; 라고 해도 됨  
5  
6  function addList() { ...  
14 }  
15  
16 function showList() { ...  
29 }  
30  
31 function removeList() {  
32     var id = this.getAttribute("id"); // this(클릭한 삭제 버튼)의 id 값 가져와 id 변수에 저장  
33     itemList.splice(id, 1); // itemList 배열에서 인덱스 값이 id인 요소 1개 삭제  
34     showList(); // 변경된 itemList 배열을 다시 화면에 표시  
35 }
```