

Array 오브젝트(**ES3**)

Array 오브젝트 개요

- 빌트인 오브젝트
- Array(배열) 형태
 - [123, "ABC", "가나다"]
 - 대괄호 안에 콤마로 구분하여 값 작성
- 배열 엘리먼트Element
 - [123, "ABC"]에서 123, "ABC" 각각을 엘리먼트 또는 요소라고 부름
 - 강좌에서는 엘리먼트로 표기
 - 2의 32승(4,294,967,296) - 1개 ++

배열 생성 방법

- `new Array()`로 생성
 - `var book = new Array();`
- `Array()`로 생성
 - `var book = Array();`
- 대괄호로 생성
 - `var book = [];`
 - 일반적으로 이 형태를 사용

엘리먼트 작성 방법

- `var book = ["책1", '책2'];`
- 대괄호 안에 콤마로 구분하여 다수 작성 가능
- `String` 타입은 큰따옴표, 작은따옴표 모두 가능
- `JS`의 모든 타입의 값, 오브젝트 사용 가능
- 값을 작성하지 않고 콤마만 작성하면 `undefined`가 설정됨

배열 차원

- 1차원 배열
 - 대괄호 하나에 엘리먼트 작성
 - [12, 34, 56] 형태 [코드 1](#)
- 2차원 배열
 - 배열 안에 1차원 배열을 작성
 - [[12, 34, 56]] [코드 2](#)
- 3차원 배열
 - 배열 안에 2차원 배열을 작성
 - [[[12, 34, 56]]] [코드 3](#)

Array 오브젝트 프로퍼티(**ES3**)

ES3 프로퍼티 리스트

이름	개요
new Array()	인스턴스 생성
Array()	인스턴스 생성
Array 프로퍼티	
length	배열의 엘리먼트 수 반환
Array.prototype	
constructor	생성자
unshift()	배열 처음에 엘리먼트 삽입
push()	배열 끝에 엘리먼트 첨부
concat()	배열 끝에 값을 연결
slice()	인덱스 범위의 엘리먼트를 복사
join()	엘리먼트와 분리자를 결합하여 반환
toString()	엘리먼트를 문자열로 연결하여 반환
toLocaleString()	엘리먼트를 지역화 문자로 변환하고 문자열로 연결하여 반환
shift()	첫 번째 엘리먼트를 삭제하고 삭제한 엘리먼트 반환
pop()	마지막 엘리먼트를 삭제하고 삭제한 엘리먼트 반환
splice()	엘리먼트를 삭제하고 새로운 엘리먼트를 삽입, 삭제한 엘리먼트 반환
sort()	엘리먼트 값을 Unicode 순서로 분류하여 반환
reverse()	엘리먼트 위치를 역순으로 바꾸어 반환

Array 인스턴스 생성

new Array()

- Array 인스턴스 생성, 반환
- 배열 생성 기준
 - 파라미터에 따라 배열 생성 기준이 다름
 - 파라미터를 작성하지 않으면 빈 배열 [코드 1](#)
 - 작성한 순서로 엘리먼트에 설정 [코드 2](#)
 - **new Array(3)**처럼 파라미터에 숫자를 작성하면 3개의 엘리먼트 생성 [코드 3](#)

Array()

- Array 인스턴스 생성, 반환
 - new Array()와 생성 방법 및 기능 같음
- 인스턴스 생성 논리
 - new Array()는 new 연산자에서 생성자 함수를 호출하여 인스턴스 생성
 - Array()는 직접 생성자 함수를 호출하여 인스턴스 생성

length 프로퍼티

- 배열 [1, 2, 3]에서
 - length 값은 3
 - Array 오브젝트에 {length: 3} 형태로 설정 [코드 1](#)
 - 처음 인덱스는 0, 마지막 인덱스는 2
- 열거/삭제는 할 수 없지만, 변경은 가능
- length 값을 변경하면
배열의 엘리먼트 수가 변경됨 [코드 2](#) [코드 3](#)

엘리먼트 추가, 삭제 메커니즘

엘리먼트 추가

- 배열에 엘리먼트를 추가하는 방법
 - 삽입할 위치에 인덱스 지정 [코드 1](#)
 - 표현식으로 인덱스 지정 [코드 2](#)

delete 연산자

구분	데이터(값)
object	매치 대상
프로퍼티	삭제할 프로퍼티 이름
인덱스	배열의 인덱스
반환	삭제 성공: true, 실패: false

- var 변수는 삭제 불가 [코드 1](#)
- 글로벌 변수는 삭제 가능 [코드 2](#)
- {name: value} 삭제 방법
 - 삭제할 프로퍼티 이름 작성 [코드 3](#) [코드 4](#)
 - ES5에서 삭제 불가 설정 가능
- 인덱스로 배열의 엘리먼트 삭제 [코드 5](#)

배열 엘리먼트 삭제 메커니즘

- 삭제된 인덱스에 `undefined` 설정 [코드 1](#)
 - 배열을 읽을 때 제외시켜야 합니다.

엘리먼트 삽입, 첨부

unshift()

구분	데이터(값)
data	기준
파라미터	[item1 [, item2 [, ...]]]opt
반환	추가 후의 length

- 0번 인덱스에 파라미터 값 삽입
- 배열에 있던 엘리먼트는 뒤로 이동 [코드 1](#)

push()

구분	데이터(값)
data	첨부 기준
파라미터	[item1 [, item2 [, ...]]]opt
반환	첨부 후의 length

- 배열 끝에 파라미터 값을 첨부 [코드 1](#)

concat()

구분	데이터(값)
data	연결 기준
파라미터	[item1 [, item2 [, ...]]]opt
반환	연결 결과

- 배열에 파라미터 값을 연결하여 반환 [코드 1](#)
- 파라미터가 1차원 배열이면 값만 반영 [코드 2](#)

엘리먼트 복사

slice()

구분	데이터(값)
data	대상
파라미터	시작 인덱스, 디폴트: 0
	끝 인덱스opt, 디폴트: length
반환	[결과]

- 배열의 일부를 복사하여 배열로 반환
 - 첫 번째 파라미터의 인덱스부터
두 번째 인덱스 직전까지 [코드 1](#)
- true, false를 숫자로 변환 [코드 2](#)
- 첫 번째 파라미터만 작성 [코드 3](#)
- 첫 번째 파라미터 값이 클 때 [코드 4](#)
- 파라미터에 음수 작성 [코드 5](#)

엘리먼트 값을 문자열로 변환

join()

구분	데이터(값)
data	기준
파라미터	분리자opt, 디폴트: 콤마(,)
반환	연결한 문자열

- 엘리먼트와 분리자를
하나씩 결합하여 문자열로 연결
 - [0] 인덱스, 분리자, [1] 인덱스, 분리자
- 마지막 엘리먼트는
분리자를 연결하지 않음 [코드 1](#)
- 파라미터를 작성하지 않으면 콤마로 분리 [코드 2](#)
- 파라미터에 빈 문자열 작성 [코드 3](#)

toString()

구분	데이터(값)
data	변환 대상
파라미터	사용하지 않음
반환	변환한 값

- 배열의 엘리먼트 값을 문자열로 연결
 - 콤마로 엘리먼트를 구분 [코드 1](#)

toLocaleString()

구분	데이터(값)
data	변환 대상
파라미터	사용하지 않음
반환	변환한 값

- 엘리먼트 값을 지역화 문자로 변환
 - 문자열을 콤마로 연결하여 반환 [코드 1](#)

[코딩 시간]

- 요구사항

```
<ul>  
  <li id=id1>id1</li>  
  <li id=id2>id2</li>  
</ul>
```

- JS로 를 10개 만들어 웹 페이지에 표시합니다.
- #id1에서 1은 일련번호로 1부터 10까지 사용합니다. [조건, 힌트](#)

엘리먼트 삭제

shift()

구분	데이터(값)
data	대상
파라미터	사용하지 않음
반환	삭제한 엘리먼트

- 배열의 첫 번째 엘리먼트 삭제
- 삭제한 엘리먼트 값이 `undefined`로 남지 않고 완전히 삭제됨
 - `length` 값이 하나 줄어 듦 [코드 1](#)
- 빈 배열이면 `undefined`가 반환 됨 [코드 2](#)

pop()

구분	데이터(값)
data	대상
파라미터	사용하지 않음
반환	삭제한 엘리먼트

- 배열의 마지막 엘리먼트 삭제
- 삭제한 엘리먼트 값이 **undefined**로 남지 않고 완전히 삭제됨
 - **length** 값이 하나 줄어 듦 [코드 1](#)
- 빈 배열이면 **undefined** 반환 [코드 2](#)

splice()

구분	데이터(값)
data	대상
파라미터	시작 인덱스, 디폴트: 0
	삭제할 엘리먼트 수
	추가할 엘리먼트: [item1 [, ...]]]opt
반환	[결과]

- 엘리먼트를 삭제하고
삭제한 엘리먼트 반환 [코드 1](#)
- 삭제한 위치에 세 번째 파라미터 삽입 [코드 2](#)
- 파라미터를 작성하지 않으면? [코드 3](#)

sort(분류)

sort()

구분	데이터(값)
data	대상
파라미터	함수opt
반환	sort 결과

- 엘리먼트 값을 승순으로 정렬
- 정렬 기준은 엘리먼트 값의 Unicode
 - 코드 포인트가 작으면 앞에 오고
크면 뒤에 옵니다. [코드 1](#) [코드 2](#)
- 주의: sort 대상 배열도 정렬됨 [코드 3](#)
- 값이 undefined이면 끝으로 이동 [코드 4](#)

sort()와 **Unicode**

- 숫자에 해당하는 Unicode의 code point로 정렬 [코드 1](#)

sort 알고리즘

sort()

- sort 알고리즘 [코드 1](#)

【코딩 시간】

- 요구사항

```
var value = [101, 26, 7, 1234];
```

배열의 엘리먼트 값을 역순으로 정렬하세요.

- 힌트

- `sort()` 메소드는 역순 정렬을 제공하지 않습니다.
- 하지만, `sort()` 메소드의 콜백 함수를 사용하여 역순으로 정렬할 수 있습니다.
- 앞 페이지의 `sort` 알고리즘을 참조하세요.

reverse()

구분	데이터(값)
data	대상
파라미터	사용하지 않음
반환	결과

- 배열의 엘리먼트 위치를 역순으로 바꿈
 - 엘리먼트 값이 아닌 인덱스 기준 [코드 1](#)
 - 대상 배열도 바뀜