배열(Array)

배열(Array)

- 하나의 변수에 여러 개의 데이터를 담는 그릇 or 여러 개의 변수를 한꺼번에 다룰수 있는 자료
- 대괄호 [] 를 사용해 생성하고, 안에 쉼표로 구분해 데이터를 입력
- 여러 개의 데이터를 인덱스 번호로 관리
- [] 를 지정 안해도 여러 개의 값이 들어있으면 무조건 배열

```
<script>
var arrayNames = []; // 배열 생성
arrayNames[0] = 데이터1; // 데이터 요소 추가
arrayNames[1] = 데이터2;
arrayNames[2] = 데이터3;
</script>
```

배열(Array)

배열의 선언과 출력

- 배열은 변수 여러 개를 한꺼번에 선언하는 것
- 배열 요소 각각을 사용하려면 배열 바로 뒤에 대괄호를 사용하고 안에 숫자를 넣어야 함

```
<script>
var arrayNames = [273, 'String', true, function () {}, {}, [273, 103]];
console.log(arrayNames);
</script>
```

배열(Array)

배열 요소에 접근하는 방법

- 배열 기호 안에 들어간 숫자를 '**인덱스**'라고 부름

```
<script>
var arrayNames = [273, 32, 103, 57, 52];
console.log(arrayNames[0]);
console.log(arrayNames[2]);
console.log(arrayNames[4]);
</script>
```

배열(Array)

배열 길이 체크

```
<script>
var arrayNames = [];
console.log(arrayNames.length);
arrayNames = [10, 20];
console.log(arrayNames.length);
arrayNames[100] = 'A';
console.log(arrayNames.length);

arrayNames = []; // 다시 초기화
</script>
```

배열(Array)

배열 주요 메서드

- join - 문자열로 리턴 (원본 변경 X)
- reverse - 거꾸로 변경
- sort - 배열 정렬
- concat - 배열 이어 붙임 (원본 변경 X)
- slice - 부분 배열 반환 (원본 변경 X)
- splice - 삭제/추가
- push, pop - 맨뒤 추가,삭제
- shift, unshift - 맨앞 추가,삭제

```
<script>
var arrayNames = [273, 'String', true, function () {}, {}, [273, 103]];
console.log(arrayNames.reverse());
console.log(Array.prototype); // 메서드 종류 확인
</script>
```

객체(Object)

객체(Object)

- 데이터와 연산 작업을 함께 담고 있는 덩어리
- 배열과 매우 비슷하지만, 훨씬 구체적이고 상세함

```
<script>
var objNames = {}; // 객체 생성
objNames.a = 데이터1; // 데이터 요소 추가
objNames.b = 데이터2;
objNames.c = 데이터3;
</script>
```

객체(Object)

객체의 선언과 출력

- 객체는 데이터 및 연산 작업을 함께 담고 있는 덩어리

```
<script>
var objNames = {
        a : 10,
        name : '홍길동',
        fn : function () {
                alert(this.name)
        }
};
console.log(objNames);
</script>
```

객체(Object)

객체 요소에 접근하는 방법

- objNames Object 에 있는 fn 함수를 호출

```
<script>
var objNames = {
        a : 10,
        name : '홍길동',
        fn : function () {
                alert(this.name)
        }
};
objNames.fn();
</script>
```

객체(Object)

객체 반복문

- 반복문으로 키값을 꺼내오는 구문

```
<script>
var objNames = {
        a : 10,
        name : '홍길동',
        fn : function () {
                alert(this.name)
        }
};
for (var key in objNames) {
        console.log(key, objNames[key]);
}
</script>
```

객체(Object)

객체 값 삭제구문

- 객체 값만 삭제가 가능

```
<script>
var objNames = {
        a : 10,
        name : '홍길동',
        fn : function () {
                alert(this.name)
        }
};
console.log(objNames.name);
delete objNames.name;
console.log(objNames.name);
</script>
```

객체(Object)

객체 주요 메서드

- hasOwnProperty - 속성 유무 반환 (true / false)

```
<script>
var objNames = {
        a : 10,
        name : '홍길동',
        fn : function () {
                alert(this.name)
        }
};
console.log(objNames.hasOwnProperty('name'));
console.log(Object.prototype); // 메서드 종류 확인
</script>
```

Thank you.

HIVELAB

Question.