

내장객체 기본

2022년 10월 8일 토요일 오후 6:03

[기본자료형과 객체 자료형의 차이]

자바스크립트 자료형 : 숫자, 문자열, 불, 함수, 객체, undefined

기본자료형 : 숫자, 문자열, 불

```
> //기본자료형
< undefined
> let number = 10;
< undefined
> let String = '안녕';
< undefined
> let boolean = true;
< undefined
> console.log(typeof number);
number
< undefined
> console.log(typeof String);
string
< undefined
> console.log(typeof boolean);
boolean
< undefined
> // 자료형 출력
```

화면 캡처: 2022-10-08 오후 6:18

즉 일반적인 숫자, 문자열, 불이 아니라 객체라는 의미

```
> console.log(typeof number);
object
```

화면 캡처: 2022-10-08 오후 6:22

객체

2022년 10월 8일 토요일 오후 6:23

(Number 객체)

```
> let numberFromLiteral=234;
< undefined
> let numberFromLiteral= new Number(233);
< undefined
```

화면 캡처: 2022-10-08 오후 6:24

위의 두가지 방법으로 생성 가능

1. 메소드

메소드	설명
toExponential()	숫자를 지수 표시로 나타낸 문자열 리턴
toFixed()	숫자를 고정소수점 표시로 나타낸 문자열 리턴
toPrecision()	숫자 길이에 따라 지수 표시 또는 고정 소수점 표시로 나타낸 문자열 리턴

```
> let number = 273.32323;
< undefined
> console.log(number.toLocaleString());
273.3
< undefined
> console.log(number.toLocaleString());
273.3232
```

2. 클래스 속성

속성	설명
MAX_VALUE	자바스크립트의 숫자가 나타낼 수 있는 최대 숫자
MIN_VALUE	자바스크립트의 숫자가 나타낼 수 있는 최소 숫자
NaN	자바스크립트의 숫자로 나타낼 수 없는 숫자
POSITIVE_INFINITY	양의 무한대 숫자
NEGATIVE_INFINITY	음의 무한대 숫자

```
> let numberA = Number.MAX_VALUE;
< undefined
> let numberB = Number.MAX_VALUE + 1;
< undefined
> console.log(numberA);
1.7976931348623157e+308
< undefined
> console.log(numberB);
1.7976931348623157e+308
```

화면 캡처: 2022-10-08 오후 6:30

실행은 되지만 제대로된 결과 출력 ❌
자바스크립트는 너무 큰 수를 다룰 때는 부동소수점 형식으로 숫자를 다룰기때문에 작은 수 무시

(String 객체)

```
> let stringFromLiteral = "End";
< 문자열 리터럴 표시
> let stringFromConstructor = new String("안녕");
```

화면 캡처: 2022-10-08 오후 6:32

1. 속성

속성	설명
Length	문자열의 길이 나타냄

2. 메소드

메소드	설명
charAt(position)	Position에 위치한 문자를 리턴
charCodeAt(position)	Position에 위치하는 문자의유니코드 번호 리턴
concat(args)	매개 변수로 입력한 문자열을 이어 리턴
indexOf(searchString, position)	앞에서부터 일치하는 문자열의 위치 리턴
lastIndexOf(searchString, position)	뒤에서 부터 일치하는 문자열의 위치를 리턴
match(regExp)	문자열 안에 regExp가있는지 확인
replace(regExp, replacement)	regExp를 replacement로 바꾼 후 리턴
search(regExp)	regExp와 일치하는 문자열의 위치를 리턴
Slice(start, end)	특정 위치의 문자열을 추출해 리턴한다
Split(separator, limit)	Separator로 문자열을 잘라 배열 리턴
Start(start, count)	Start 부터 count만큼 문자열을 잘라서 리턴
Substring(start, end)	Start 부터 end 까지 문자열을 잘라서 리턴
toLowerCase()	문자열을 소문자로 바꿔 리턴
toUpperCase()	문자열을 대문자로 바꿔 리턴

자기가자신을 변경하는 것이 아니라 변경된 값을 리턴

```
> let string = 'abcde' ;
< undefined
> string.toLowerCase();
Uncaught TypeError: string.toLowerCase is not a function
> console.log(string);
abcde
```

화면 캡처: 2022-10-08 오후 6:59

```
> let string = 'abcde';
Uncaught SyntaxError: Identifier 'string' has already been declared
> string = string.toLowerCase();
> console.log(string);
abcde
```

화면 캡처: 2022-10-08 오후 7:01

3. 메소드 활용

- indexOf():
: 매개변수로 받는 문자열이 포함된 부분 리턴
: 문자열이 포함되어 있지 않으면 -1 리턴 따라서 리턴값이 0보다 큰지 확인하면 문자열 내부에 특정 문자열 있는지 확인 할 수 있다.

```
> let string = "안녕하세요, 좋은 아침입니다.";
< undefined
> if (string.indexOf("안녕"))>=&#x2D;0){
Uncaught SyntaxError: Unexpected token '{'
> if (string.indexOf("안녕"))>=&#x2D;0){
console.log('좋은 아침이네요!');
}
좋은 아침이네요!
< undefined
```

- Split():
: 특정 기호를 기반으로 문자열을 분해할 수있다

```
> let string = '김자,고구마,바나나,사과';
< undefined
> let array=string.split(',');
< undefined
> console.log(array);
(4) ["김자", "고구마", "바나나", "사과"]
```

화면 캡처: 2022-10-09 오후 4:19

(date 객체)

클래스	설명
New date()	현재시간으로 date 객체 생성
New date(유니크스 타임)	유니크스타임(1970년1월1일00시00분00초 부터 경과한 밀리초)으로 date 객체 생성
New date(시간, 문자열)	문자열로 date 객체 생성
New date(년, 월-1, 일, 시간, 분, 초, 밀리초)	시간 요소 (연, 월-1, 일, 시간, 분, 초, 밀리초) 기반으로 date 객체 생성

주의 : month 나타내는 월은 0부터 시작
0이 1월 11이 12월

- 1. 메소드 활용
 - Get00() 형태의 메소드 가짐

(array 객체)

1. 기본 메소드

메소드	설명
Concat()	매개 변수로 입력한 배열의 요소를 모두 합쳐 배열 만들어 리턴
Join90	배열 안의 모든 요소를 문자열로 만들어 리턴
Pop()	배열 마지막 요소 제거하고 리턴
Push()	배열의 마지막 부분에 새로운 요소 추가
Reverse()	배열의 요소 순서를 뒤집는
slice()	배열 요소의 지정한 부분을 리턴한다
Sort()	배열의 요소를 정렬한다
Splice()	배열 요소의 지정한 부분을 삭제하고 삭제한 요소 리턴

*표시 메소드는 자기자신을 변화 : 파괴적 메소드

2. 콜백 함수와 함께 사용하는 메소드

메소드	설명
ForEach()	배열의 요소를 하나씩 뽑아 반복 돌린다
Map()	콜백 함수에서 리턴하는 것으로 기반으로 새로운 배열 만든다
Filter()	콜백 함수에서 true를 리턴하는 것으로만 새로운 배열을 만들어리턴