

JavaScript & Jquery 09

by withsky

1 객체

객체(Object)란?

프로퍼티(Property) : 속성, 데이터 값

메서드(Method) : 동작, 기능, 절차

객체 예시

자동차

속성 : 엔진, 바퀴, 핸들...

동작 : 가속, 브레이크, 후진, 좌회전, 우회전

객체지향 프로그래밍 (Object Oriented Programming, OOP)

명령어(소스 코드) 단위가 아닌 독립단위 (객체) 단위로 프로그래밍 하는 방식

각 객체는 서로 메시지를 주고 받으며, 데이터를 처리 한다. (반대 : 절차지향 프로그래밍)

예) Java, C++, Python

장점 : 대규모 프로젝트인 경우 코드 재사용이 쉬우며, 유지보수 (업그레이드)가 쉽고, 디버깅이 쉽다.

2 객체

객체 선언방법

배열과 비슷한 개념
배열은 각 요소에 접근할 때 '인덱스'를
사용하지만, 객체는 '키'를 사용

속성(프로퍼티)는 변수형태로 선언
메서드는 함수형태로 선언

```
var car = {  
  company : "현대",  
  type : "소형",  
  name : "아반떼",  
  
  accell : function () {  
    document.write("가속합니다.");  
  },  
  
  backward : function () {  
    document.write("후진합니다.");  
  }  
};  
  
console.log(car);  
  
document.write(car.name);  
document.write(car['company']);  
  
car.accell();
```

3 객체 속성 추가/삭제

속성 추가

생성된 객체에 . 으로 직접 추가

속성 삭제

delete 키워드로 삭제

```
// 속성 추가
var student = {};
student.name = "홍길동";
student.gender = "남";

student.write = function() {
    document.write("제 이름은 "+this.name+"입니다.");
    document.write("저는 "+this.gender+"성입니다.");
}

student.write();

// 속성 삭제
delete(student.name);
document.write(student.name);
```

4 객체 - 생성자 함수 생성

instance 객체

new 키워드로 생성

생성자 함수는 instance 객체를 선언하지 않으면 접근 할 수 없음.

```
function Square(w, h) {  
    var width = w;  
    var height = h;  
  
    var r = 0;  
  
    this.getResult = function() {  
        r = w * h;  
        return r;  
    };  
}  
  
var s1 = new Square(5,10);  
  
document.write(s1.r);    // 접근 불가  
document.write(s1.getResult());    // 함수 실행  
  
document.write("<br/>");  
  
// new 키워드를 사용하지 않은 경우  
var s2 = Square(7,10);  
  
document.write(s2.getResult());
```

5 객체 클래스 생성

ECMAScript 6

ECMA에서 정의한 스크립트 프로그래밍 표준

클래스 (Class) 생성 방법

class 키워드로 생성
생성자는 constructor

instance는 생성자 함수와 같이
new 키워드로 생성

```
class Square {  
  
    constructor(w, h) {  
        this.width = w;  
        this.height = h;  
  
        //document.write(this.height);  
    }  
  
    getResult() {  
        //console.log(this.)  
        this.r = this.width * this.height;  
        console.log(this.r);  
        return this.r;  
    }  
  
}  
  
var s1 = new Square(5,10);  
document.write(s1.getResult());
```

6 내장 객체

내장 객체 종류

Number : 숫자를 조작하기 위한 객체

String : 문자를 조작하기 위한 객체

Boolean : 불리언값(true/false)를 조작하기 위한 객체

Array : 배열을 조작하기 위한 객체

Function : 함수를 조작하기 위한 객체

Math : 수치 연산을 하기 위한 객체

Date : 날짜 관련 조작하기 위한 객체

RegExp : 정규표현에 관한 기능을 제공하는 객체

Error : 에러정보를 관리하는 객체

6 내장 객체 - Number

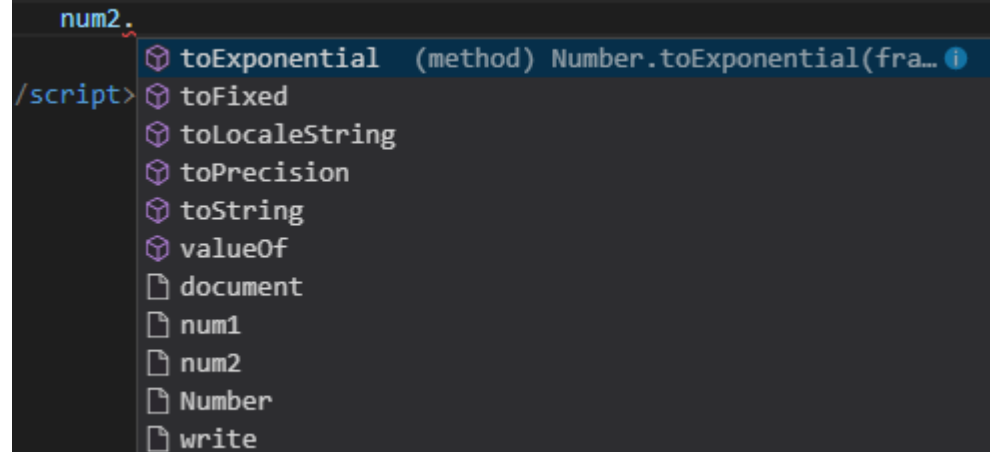
Number

toExponential : 지수표기

toFixed : 소수점 반올림

toPrecision : 전체 자리수 반올림

```
var num1 = 100;  
var num2 = new Number(100);  
  
document.write(typeof(num1)); // number  
document.write(typeof(num2)); // object
```



```
num2.  
/script> toExponential (method) Number.toExponential(fra...  
toFixed  
toLocaleString  
toPrecision  
toString  
valueOf  
document  
num1  
num2  
Number  
write
```


7 내장 객체 - String

String

length : 문자열의 길이 (속성)

replace : 문자열 교체

indexOf : 앞에서부터 일치하는
문자열 인덱스 리턴

split : 문자열로 잘라 배열로 리턴

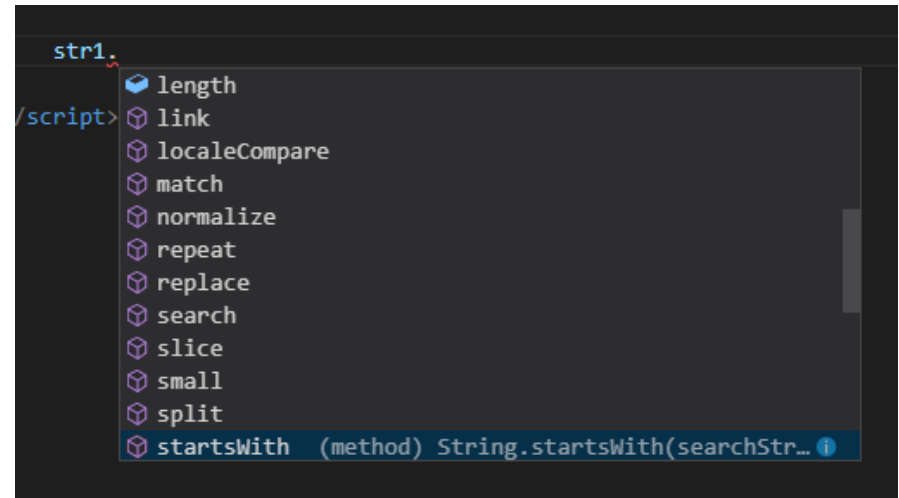
substr : 문자열 자르기

toLowerCase : 문자열을 소문자로 바꿈.

toUpperCase : 문자열을 대문자로 바꿈

```
var str1 = "hello";  
var str2 = new String("hello");  
  
document.write(typeof(str1)); // string  
document.write(typeof(str2)); // string
```

```
var str = "hello";  
document.write(str.length); // 5  
document.write(str.replace("e", "a")); // hallo  
document.write(str.indexOf("l")); // 2  
var strArr = str.split("e");  
document.write(strArr[0]); // h  
document.write(str.substr(0,3)); // hel  
document.write("HELLO".toLowerCase()); // hello  
document.write("hello".toUpperCase()); // HELLO
```



8 내장 객체 - Array

Array

생성자 함수

Array() : 빈 배열 생성

Array(number) : number 크기의 배열 생성

Array(num1, num2...) : 매개변수를 배열로 생성

속성

length : 배열 요소의 길이

메서드

join() : 모든 요소를 문자열로 바꿔 리턴

pop() : 배열의 마지막 요소를 제거 후 리턴

push() : 배열의 마지막에 새로운 요소 추가

reverse() : 배열의 요소 순서를 뒤집는다.

slice() : 배열 요소의 지정한 부분을 리턴

sort() : 배열의 요소를 정렬 후 리턴

splice() : 배열 요소의 지정한 개수까지

삭제하고 삭제한 요소를 리턴

```
var arr = new Array(1,2,3,4,5);
console.log(arr);

console.log(arr.concat());
console.log(arr.join());
arr.push(6)
console.log(arr);
arr.reverse();
console.log(arr);
console.log(arr.slice(2,4));
var arr2 = arr.sort();
console.log(arr2);
var arr3 = arr2.splice(3);
console.log(arr3);

// 내림차순 정렬
var arr4 = new Array(1,2,3,4,5);
var arr5 = arr4.sort(function (left, right) {
    return right - left;
});
console.log(arr5);
```

9 내장 객체 - Math

Math

내장 객체 중 유일하게 생성자 함수가 아닌 객체
new 로 instance 화 해서 사용할 수 없음.

메서드

abs(x) : x의 절대값

max(x,y,z,...) : 매개변수 중 최대값

min(x,y,z,...) : 매개변수 중 최소값

pow(x,y) : x의 y제곱

round(x) : x의 반올림

floor(x) : x보다 작거나 가장 큰 정수

random() : 0부터 1까지의 임의의 수

```
var a = Math.random();  
  
// 두자리 랜덤 수  
document.write(Math.floor(a*100));
```

9 내장 객체 - Date

Date

날짜, 시간관련 객체

메서드

getDate() : 날짜

getMonth() : 월

getFullYear() : 년도

getTime() : 1970년 1월 1일 0시 부터 현재시간
까지의 밀리초

...

```
<script>
    var date = new Date();
    alert(date);
</script>
```

이 페이지 내용:

Sun Sep 02 2018 22:40:15 GMT+0900 (한국 표준시)

확인

```
// 두 날짜 사이 간격 구하기
var now = new Date();
var before = new Date(2017,8,2,);

var interval = now.getTime() - before.getTime();

interval = Math.floor(interval / (1000 * 60 * 60 * 24));

document.write(interval);
```