


# HTML – 09

## 자바스크립트 객체

# 객체

- **객체(object)**란 하나의 변수에 여러 속성을 저장할 수 있도록 해주는 데이터 타입이다.
- key/value가 한 쌍으로 저장되는 구조이며 그 중 value가 함수로 정의된 것은 객체가 가지는 동작. 즉 메소드라 칭한다.
- JavaScript는 객체 기반의 스크립트 언어로서, 거의 모든 것이 객체로 이루어져 있다.

Object	Properties	Methods
	<code>car.name = Fiat</code>	<code>car.start()</code>
	<code>car.model = 500</code>	<code>car.drive()</code>
	<code>car.weight = 850kg</code>	<code>car.brake()</code>
	<code>car.color = white</code>	<code>car.stop()</code>

# 객체의 종류

- 내장 객체 (built-in Object)
  - 생성자가 미리 작성되어 있어 바로 사용 가능하다.
  - Date, Object, Array 등 많은 내장 객체들이 제공된다.
- 사용자 정의 객체 (custom Object)
  - 사용자가 생성자를 정의한다.

# 객체의 생성

- 리터럴 사용
  - 가장 간단하고 쉽고 빠른 방법이며, {이름 : 값 , ...} 형태로 정의
- new 연산자 사용
  - 사용자가 직접 생성자 함수 정의 후 객체 생성
  - 생성자 함수는 대문자로 정의
- 새로운 객체를 만들어 속성을 정의하지만 리터럴 방식을 더 권장함

객체의 경우 상수로 선언해도 메모리값만 상수일 뿐 객체 안의 내용은 변경이 가능하다.  
즉 객체가 저장된 공간을 가리키는 정보만 상수일뿐 그 객체의 정보 자체는 변경이 가능하다.  
이런 이유로 JavaScript에서 객체는 변수로 선언할 이유가 없으며 거의 모든 케이스에서 상수로 선언하는게 일반적이다

# 리터럴을 이용한 객체 생성

```
const instance = {  
  
  property1 : 초기값,  
  property2 : 초기값,  
  
  method1 : function() {  
  },  
  method2 : function() {  
  }  
  
}
```

```
const myCar = {  
  model : " bmw " ,  
  color : " white " ,  
  speed : 100,  
  brake : function() {  
    this.speed = 0;  
  },  
  accel : function() {  
    this.speed = 150;  
  }  
}
```

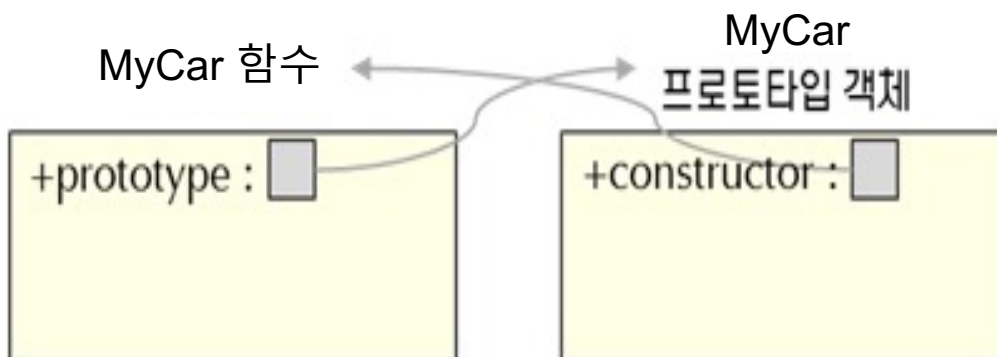
//ES6 축약방식(function 키워드 생략)

```
  brake() { this.speed = 0; },  
  accel() { this.speed = 150; }  
};
```

# 생성자를 이용한 객체 생성

```
//생성자 함수는 대문자로  
function 클래스 이름() {  
  this.property1 = 초기값;  
  this.property2 = 초기값;  
  
  this.method1 = function() {  
  }  
  this.method2 = function() {  
  }  
}
```

```
function MyCar(model, color, speed) {  
  
  this.model = model;  
  this.color = color;  
  this.speed = speed;  
  
  this.brake = function() {  
    this.speed = 0;  
  }  
  this.accel = function() {  
    this.speed = 150;  
  }  
}
```



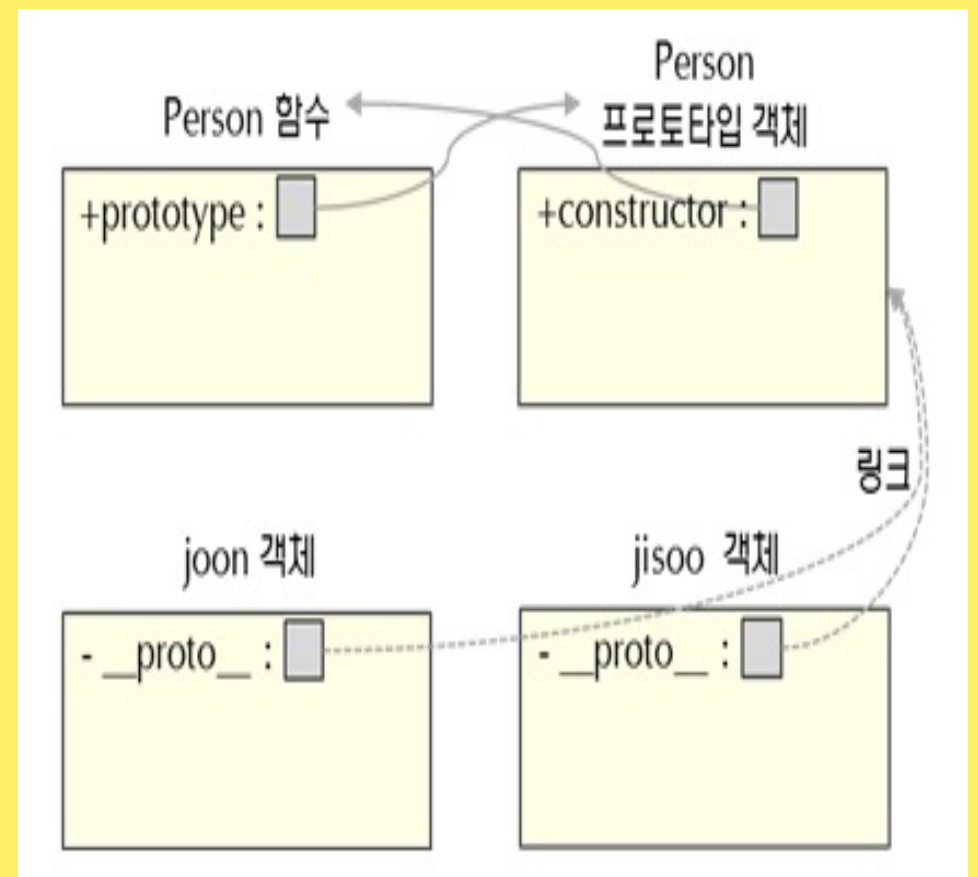
```
const myCar =  
  new MyCar("BMW", "gray", 100);
```

# 객체에 속성과 메소드 추가

- 기존에 존재하고 있던 객체에도 속성을 추가할 수 있다.
- 생성자 함수는 변경할 필요가 없다.

```
function Person(first, last, age, eyecolor) {  
  this.firstName = first;  
  this.lastName = last;  
  ...  
}  
Person.prototype.name = function() {  
  return this.firstName + " "  
    + this.lastName;  
};
```

```
let joon = new Person("lee","joon", ...);  
let jisoo = new Person("ji","soo",...);  
document.write(joon.name() + " , "  
  + jisoo.name());
```



lee joon , ji soo

# 자바 스크립트 내장 객체

- Object 객체 - `new Object()` 대신 `{}` 사용 가능
- Array 객체 - `new Array()` 대신 `[]` 사용 가능
- Date 객체
- Number 객체
- String 객체 - `new String()` 대신 `“”` 사용 가능
- Boolean 객체 - `new Boolean()` 대신 `true, false` 사용 가능
- Function 객체 - `new Function()` 대신 `function(){} 사용가능`
- Math 객체 - `pow()`, `floor()` `round()` `ceil()`, `max()`, `min()`...
- ...
  - `new` 키워드를 사용해서 생성 (예. `var date = new Date();`)
  - Math는 전역 객체이기 때문에 `new` 키워드를 사용하지 않음



# Array 객체

- 배열은 하나의 변수에 많은 값을 담을 수 있다.
- 인덱스 번호를 참조하여 값에 접근할 수 있다.
- 배열 요소의 타입이 고정되어 있지 않으므로 같은 배열에 서로 다른 타입의 요소가 존재할 수 있다.
- 현재 배열의 크기보다 큰 인덱스를 사용하여 크기를 자동으로 증가시킬 수 있다.
- 리터럴을 사용한 배열
  - `const cars = ["Saab", "Volvo", "BMW"];`
- `new` 키워드를 사용한 배열
  - `const cars = new Array("Saab", "Volvo", "BMW");`

# Array 객체의 메소드

- 속성
  - length
- 메소드

메소드	설명
indexOf(item, start)	배열에서 요소를 찾아 위치를 반환한다.
lastIndexOf(item, start)	역순으로 요소를 찾아 위치를 반환한다.
push(a,b,c,...)	배열 끝에 요소를 추가한다.
pop()	마지막 요소를 제거하고 반환한다.
shift()	배열 처음의 원소를 제거하고 반환한다.
unshift(a,b,c,...)	배열 처음에 요소를 추가한다.

# Array 객체의 메소드

메소드	설명
reverse()	배열의 순서를 반대로 나타낸다.
sort(sortfunction)	배열을 정렬한다. 인수로 값을 비교하는 함수를 지정할 수 있으며 생략 시 사전 순으로 정렬된다.
slice(start, end)	start ~ end(포함하지 않음) 범위의 요소를 따로 떼어내어 새로운 배열을 만든다.
splice(index, n, a, b, c, ...)	배열 일부를 수정한다. 일정 범위를 삭제하고 새로운 요소를 삽입한다.
abc= a.concat(b,c...)	여러 개의 배열을 합친다.
join(deli)	배열 요소를 하나의 문자열로 합친다. 구분자를 지정할 수 있으며 생략 시 콤마로 구분한다.

# Array 객체 문제

1. prompt()를 이용하여 이름을 계속 입력 받아 배열에 저장하고 출력하는 프로그램을 작성하시오.
  - 입력의 마지막은 공백 문자를 입력하거나 “취소” 버튼을 눌렀을 때(null)로 한다.
2. 서로 중복되지 않은 정수 5개를 입력 받아 출력하는 프로그램을 작성하시오

# Date 객체

- Date 객체는 현재 날짜와 시간으로 새 날짜 객체를 생성한다.
- 새 날짜 객체를 만드는 4가지 방법
  - `new Date()` // 현재 날짜와 시간
  - `new Date(milliseconds)` // 1970/01/01 이후의 밀리 초
  - `new Date(dateString)`
  - `new Date(year, month, date[, hours[, minutes[, seconds[,ms]]]])`
- UTC
  - 1972년 1월 1일부터 시행된 국제 표준시이다.
  - UTC는 그리니치 평균시(GMT)로 불리기도 하는데, UTC와 GMT는 초의 소수점 단위에서만 차이가 나기 때문에 일상에서는 혼용되어 사용된다.
  - 기술적인 표기에서는 UTC가 사용된다.

# Date 객체의 메소드

get메소드	반환 값	set메소드
getDay()	0(일요일) ~ 6(토요일)	setDay(day)
getDate()	1 ~ 31	setDate(date)
getMonth()	0 ~ 11	setMonth(month-1)
getFullYear()	4개의 숫자로 된 연도	setYear(year)
getHours()	0 ~ 23	setHours(hours)
getMinutes()	0 ~ 59	setMinutes(minutes)
getSeconds()	0 ~ 59	setSeconds(seconds)
getMilliseconds()	0 ~ 999	setMilliseconds(millisec)
getTime()	경과시간(milliseconds 단위)	setTime(millisec)

- getTime()의 millisecond값을 1000으로 나누면 실제 초를 얻을 수 있다.  
예) millisecond / 1000 / 60 / 60 / 24 / 365 : 초 / 분 / 시 / 일 / 년도 별 계산

# Date 객체 문제

- 요일구하기
  - If나 switch를 이용
  - 배열을 이용
- 태어난 날로부터 현재까지 경과한 날 수
- 오늘로부터 500일 후 날짜를 계산하시오.
  - 현재 날짜의 millisecond 구함
  - 500일에 대한 ms값을 구함 (  $500 * 1000 * 60 * 60 * 24$  )
  - 일자로 변환 (  $ms / 1000 / 60 / 60 / 24$  )

# Date 객체 문제

1. getTime()를 이용하여 교환 기간을 계산하는 프로그램을 작성 하시오.

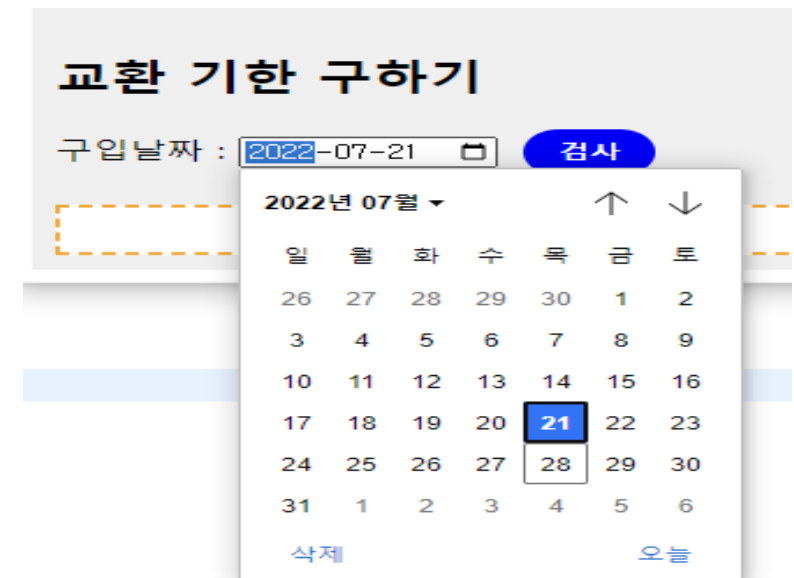
- 구입 날짜와 현재 날짜의 Date객체를 생성하고 getTime()으로 milliseconds 값을 반환 받음

- 현재 날짜에서 구입 날짜를 뺀 milliseconds 값을 일자로 변경  
( milliseconds / 1000 / 60 / 60 / 24)

- 결과값 비교

- 7 보다 크면 "교환 불가 ",

- 작으면 "교환 가능" 출력





# String 객체의 메소드

- 속성
  - length : 문자열의 길이

메소드	설명
charAt(index)	index 위치의 문자를 구한다. index가 문자열의 범위를 벗어나면 빈 문자열이 반환된다.
charCodeAt(index)	index 위치의 문자에 대한 유니코드를 구한다.
indexOf (searchvalue, start)	부분 문자열의 위치를 검색한다. start는 검색 시작 위치이며 생략 시 0이 적용되어 처음부터 검색한다. 없을 경우 -1을 반환한다.
lastIndexOf (searchvalue, start)	부분 문자열의 위치를 역방향에서 검색한다. start는 검색 시작 위치이며 생략 시 문자열의 제일 끝이 적용된다. 없을 경우 -1을 반환한다.
concat(s1, s2, ...)	여러 개의 문자열을 연결한다. + 연산자와 동일하다.
trim()	앞 뒤의 공백을 제거한다.

# String 객체의 메소드

메소드	설명
toLowerCase()	소문자로 변환한다
toUpperCase()	대문자로 변환한다
replace(search,value)	문자열을 대체한다. 정규식도 사용 가능
search(searchvalue)	부분 문자열 또는 정규식을 검색하여 그 위치를 반환한다.
match(regex)	정규식으로 검색하여 일치하는 결과를 배열로 반환한다. 발견되지 않으면 null을 반환한다.
substring(from, to)	두 위치 사이의 부분 문자열을 추출한다. to를 생략하면 뒤쪽 모든 문자열을 추출한다.
slice(start, end)	start 위치에서 end 위치까지 부분 문자열을 추출한다. 음수로 끝에서부터 위치를 지정할 수 있다.
substr(start, length)	start에서 시작하여 length 길이만큼 부분 문자열을 추출한다. 길이를 생략하면 뒤쪽 모든 문자열을 추출한다.
split(separator, limit)	구분자로 구분된 문자열을 분리하여 배열로 반환한다. Limit는 최대 몇 개까지 반환할 것인가를 지정한다.

# 유니코드란

- 유니코드란 전 세계적으로 사용하는 모든 문자 집합을 하나로 모은 것이다. 유니코드 1.0.0은 1991년 8월 제정되었으며, 그 후 약 5년이 지나서야 유니코드 2.0.0에 한글 11,172자가 모두 포함되었다. 현재 버전은 2010년 10월 11일 제정된 6.0이다.
- 유니코드 값을 나타내기 위해서는 코드 포인트(code point)를 사용하는데, 보통 U+를 붙여 표시한다. 예를 들어, 'A'의 유니코드 값은 U+0041로 표현한다(Wu0041로 표기하기도 함). 유니코드는 공식적으로 31비트 문자 집합이지만 현재까지는 21비트 이내로 모두 표현이 가능하다. 유니코드는 논리적으로 평면(plane)이라는 개념을 이용하여 구획을 나누며, 평면 개수는 0번 평면인 기본 다국어 평면(BMP; Basic Multilingual Plane)에서 16번 평면까지 모두 17개이다. 대부분의 문자는 U+0000~U+FFFF 범위에 있는 기본 다국어 평면에 속하며, 일부 한자는 보조 다국어 평면(SMP, Supplementary Multilingual Plane)인 U+10000~U+1FFFF 범위에 속한다. 이 중 한글은 U+1100~U+11FF 사이에 한글 자모 영역, U+AC00~U+D7AF 사이의 한글 소리 마디 영역에 포함된다.
- **유니코드의 인코딩 방식**
- 유니코드의 인코딩 방식으로는 코드 포인트를 코드화한 UCS-2와 UCS-4, 변환 인코딩 형식(UTF, UCS Transformation Format)인 UTF-7, UTF-8, UTF-16, UTF-32 인코딩 등이 있다. 이 중 ASCII와 호환이 가능하면서 유니코드를 표현할 수 있는 UTF-8 인코딩이 가장 많이 사용된다. UTF-8은 코드 포인트 범위에 따라 다음 표에서 보는 바와 같이 인코딩 방식이 다르다. 다음 표는 코드 포인트 범위에 따른 UTF-8 인코딩 방식을 보여준다.

# 예제

- test\_stringMethod

```
str = 우리 나라 좋은 나라
```

```
name= 홍길동/개나리/진달래/무궁화/삼천리
```

## 찾기

charAt

charCodeAt

indexOf

lastIndexOf

## 추출

substring

substr

slice

## 기타

toLowerCase

toUpperCase

split

trim

# String 객체 문제

- prompt로 주민등록번호를 입력 받아 생년월일과 성별, 나이를 출력하는 프로그램을 작성하시오.
  - 예) 주민등록번호를 110326-4432618로 입력 받은 경우
  - 생일 : 2011년 3월 26일
  - 성별 : 여자
  - 나이 : 12
- 주민등록번호를 입력 받아 주민등록번호의 유효성을 검사하는 프로그램을 작성하시오.(ABCDEFGH-IJKLMNOP)
  1.  $A*2 + B*3 + \dots + H*9 + I*2 + \dots + L*5$ 의 총합을 구한다.
  2. 1번의 합을 11로 나눈 나머지를 구한다.
  3. 11에서 2번의 결과를 뺀다.
  4. 3번의 결과가 0~9이면 값 그대로, 10이면 0, 11이면 1로 변환
  5. 4번의 결과와 M자리의 값이 같으면 맞는 번호이다.

# Math 객체 속성

속성	설명
E	오일러의 상수(약 2.718)
PI	파이 상수(약 3.14)
SQRT2	2의 제곱근(약 1.414)
SQRT1_2	$\frac{1}{2}$ 의 제곱근(약 0.707)
LN2	자연 로그(밑수:2)(약 0.693)
LN10	자연 로그(밑수:10)(약 2.302)
Math.LOG2E	e의 로그 2
Math.LOG10E	e의 로그 10

# Math 객체 메소드

메소드	설명
abs(x)	절대값
ceil(x), floor(x)	실수를 정수로 올림, 내림 함수
cos(x), sin(x), tan(x)	삼각함수
exp(x)	지수함수
log(x)	로그함수
max(x,y,z,...,n)	최대값
min(x,y,z,...,n)	최소값
pow(x,y)	지수함수 $x^y$
random()	0과 1사이의 난수 값 반환
round(x)	반올림
sqrt(x)	제곱근

# 원하는 범위의 랜덤 값 생성

- $0.0 \leq \text{Math.random()} < 1.0$
- $\text{Math.floor}(\text{Math.random()} * (\text{최대값} - \text{최소값} + 1) + \text{최소값});$
- $\text{Math.round}(\text{Math.random()} * (\text{최대값} - \text{최소값}) + \text{최소값});$
- 예) 1부터 10까지의 랜덤 수 만들기

```
var ranNum = Math.floor(Math.random() * (10 - 1 + 1) + 1);  
document.write(ranNum);
```

- $\text{Math.random()} * 10 \Rightarrow 0 \sim 9$
- $\text{Math.random()} * 10 + 1 \Rightarrow 1 \sim 10$
- $\text{Math.random()} * 20 + 11 \Rightarrow 11 \sim 30$



# 예제

```
<h1>Number Guess</h1>
```

```
<p>1부터 100 사이 숫자 입력</p>
```

```
<input id="number" type="text">
```

```
<button type="button" onclick="test()">추측</button>
```

```
<p id="hint"></p>
```

### Number Guess

1부터 100 사이 숫자 입력

힌트 : 더 큰수를 입력하세요

# Math 객체 문제

1. 가위 바위 보 게임을 할 수 있는 프로그램을 작성하시오.  
(컴퓨터는 랜덤, 사용자는 prompt로 입력 받아서 처리)
2. 로또 번호를 생성하는 프로그램을 작성하시오.  
(1번 ~ 45번 중 6개의 번호를 चु침)