

웹 프로그래밍

- javascript -

자바스크립트 특징

1. 인터프리터 언어
2. 동적 타이핑(dynamic typing)
3. 구조적 프로그래밍 지원
4. 함수형 프로그래밍 지원
5. 프로토타입 기반(prototype-based)

변수 정의시 유의사항

1. 숫자로 시작하면 X : var 1st = 11;
2. 대소문자 구분 : var name 과 var Name 은 다른 변수
3. 소문자로 시작 : 암묵적 규칙
4. 변하지 않는 환경 변수의 값을 담는 상수 변수는 모두 대문자로 : var ADMIN_ID = "admin";
5. 여러 단어가 조합되는 경우 낙타 표기법 : var userName = "web";
6. 이미 정의된 예약어를 사용하면 X : break, case, default, delete, this, var, for, while etc□

변수형 예제

```
<script>
var a = 10;
var b = a;
a = '30';
document.write("a = " + (a+b) + ", b = " + (b+b) );
</script>
```

출력값은? a = 3010, b = 20

객체란?

- 실제 세상에 존재하는 사물을 모델링한 것
- 데이터와 동작을 가지고 있다.
- 데이터 객체가 가지고 있는 특성값
- 동작 객체가 수행할 수 있는 행동
- 데이터 변수(속성) , 동작 함수(메소드)
- 속성(property)+메소드(method) = 객체(object)

객체의 생성과 사용

```
function Car(model, speed, color)
{
    this.model = model;
    this.speed = speed;
    this.color = color;

    this.brake = function() { this.speed -= 10; }
    this.accel = function() { this.speed += 10; }
}

var myCar = new Car("avante", 60, "black");
myCar.model;
myCar.speed;
myCar.brake();
myCar.accel();
```

프로토타입

- 객체들 사이에서 메서드를 공유하는 방법
- 객체 지향 언어들은 class 라는 개념을 사용
- JS는 프로토타입(prototype)

```
function Car(model, speed, color)
{
    this.model = model;
    this.speed = speed;
    this.color = color;

    this.brake = function() { this.speed -= 10; }
    this.accel = function() { this.speed += 10; }
}
Car.prototype.getSpeed = function()
{
    alert("속도는" + this.speed + "km/s 입니다");
}

var p1 = new Car( abc, 70, red);
var p2 = new Car( bcd, 80, blue);

var s1 = p1.getSpeed();
var s2 = p2.getSpeed();
```

내장객체-Date객체

- 날짜와 시간에 대한 객체생성
- today = new Date();
- 현재 날짜 시간 : getYear(), getMonth(), getDate()
- 날짜 시간 세팅 : setYear(), setMonth(), setDate()
- getFullYear()
- 이외에도 많은 메서드 들이 존재합니다. 관련 사이트 참고

내장객체-String객체

- length 속성
- concat(), indexOf(), lastIndexOf(), toUpperCase(), toLowerCase()
- big(), small(), bold(), italics(), strike(), fontcolor(), fontsize(), sub(), sup() 등
- match(), replace(), split() 등
- 이외에도 많은 메서드 들이 존재합니다. 관련 사이트 참고

내장객체-String객체

```
<script>
    var s = "aBcDeF";
    s.toLowerCase();
    document.write(s);

    var result = s.toLowerCase();
    document.write(result);
</script>
```

출력 결과는? aBcDeF ABCDEF

내장객체-String객체

```
<script>
  var sLiteral = "문자열 리터럴";
  sObject = new String("문자열 객체");

  function change(strlit, strobj) {
    strlit = "Hello World";
    strobj = "Hello World";
  }
  change(sLiteral, sObject);
  document.writeln(sLiteral + "<br>");
  document.writeln(sObject + "<br>");
</script>
```

출력결과는?

내장객체-Number객체

- var num = new Number("10");
- 1.2345.toString(2);
- toExponential([digits]), toFixed([digits]), toString([radix]), toPrecision([precision])
- 이외에도 많은 메서드 들이 존재합니다. 관련 사이트 참고

내장객체-Math객체

- 수학적인 작업을 하기 위한 객체
- 생성시 new를 이용하지 않음
- Math 객체의 속성
 - E : 오일러의 상수, LN2, LN10 : 자연로그2, 10
 - SQRT2 : 2의 제곱근, PI : 파이 값
- Math 객체의 메서드
 - abs(x) : 절대값
 - acos(x), asin(x), atan(x), cos(x), sin(x), tan(x)
 - ceil(x), floor(x) : 올림, 내림
 - max(x,y,z,□), min(x,y,z,□)
 - pow(x,y) : x^y
 - random() : 0과 1사이의 난수값
 - round(x) : 반올림
 - sqrt(x) : 제곱근
- 이외에도 많은 메서드 들이 존재합니다. 관련 사이트 참고

내장객체-Array객체

- 속성 : length
- 메서드
 - concat(value1[value2[value□]]);
 - push(value); , pop();
 - sort();
 - join(delimiter);
 - 2차원 배열
 - indexOf(searchStr);
 - shift(); , unshift();
 - slice([begin[,end]]);
 - filter();
- 이외에도 많은 메서드 들이 존재합니다. 관련 사이트 참고

내장객체-Array객체

```
<script>
  var myArray = [10, 7, 23, 99, 169, 19, 11, 1];
  myArray.sort()
  document.write(myArray);
</script>
```

```
<script>
  var myArray = [10, 7, 23, 99, 169, 19, 11, 1];
  myArray.sort(function (a, b) { return a - b });
  document.write(myArray);
</script>
```

- compareFunction (a, b)가 0보다 작은 경우 a를 b보다 낮은 색인으로 정렬합니다. 즉, a가 먼저옵니다.
- compareFunction (a, b)가 0을 반환하면 a와 b를 서로에 대해 변경하지 않고 모든 다른 요소에 대해 정렬합니다.
- compareFunction (a, b)가 0보다 큰 경우, b를 a보다 낮은 인덱스로 정렬합니다.

내장객체-Window객체

- 메서드 : alert(), confirm(), prompt(), setInterval(), setTimeout(), moveTo(), moveBy(), open() 등
- 이외에도 많은 메서드 들이 존재합니다. 관련 사이트 참고

HTML 변경하기 - 요소의 내용

```
<script>
  function get() {
    var val = document.getElementById("ex").innerHTML;
    alert(val);
  }
  function set(v) {
    document.getElementById("ex").innerHTML = v;
  }
</script>
<body>
  <div id="ex">여기가 div로 선언되었습니다.</div>
  <a href="#" onclick="get()">내용 출력하기</a><br>
  <a href="#" onclick="set('변경되었습니다.')">내용 변경하기</a>
</body>
```

HTML 변경하기 - 요소의 속성

```
<body>
  

  <script>
    function changeImage() { document.getElementById("image").src = "b.jpg"; }
  </script>
  <input type="button" onclick="changeImage()" value="눌러보세요" />
</body>
```

HTML 변경하기 - 요소의 스타일

```
<body>
<p id="p1">This is a paragraph.</p>
<script>
  function changeStyle() {
    document.getElementById("p1").style.color = "red";
    document.getElementById("p1").style.fontStyle = "italic";
  }
</script>
  <input type="button" onclick="changeStyle()" value="눌러보세요" />
</body>
```

HTML 변경하기 - DOM 추가

```
<ul id="myList">
  <li>Coffee</li>  <li>Tea</li>
</ul>

<button onclick="myFunction()">Add</button>
<script>
  function myFunction() {
    var node = document.createElement("li");
    var textnode = document.createTextNode("Milk");
    node.appendChild(textnode);
    document.getElementById("myList").appendChild(node);
  }
</script>
```

HTML 변경하기 - DOM 삭제

```
<script>
  function removeNode() {
    var parent = document.getElementById("target");
    var child = document.getElementById("p1");
    parent.removeChild(child);
  }
</script>

<div id="target">
  <p id="p1">첫번째 단락</p>
  <p id="p2">두번째 단락</p>
</div>
<button onclick="removeNode()">누르세요!</button>
```