

# JavaScript

## 3. 객체

# OBJECT 기본

# 객체

---

- 객체는 하나 이상의 속성(property)을 가질 수 있는 변수이다.
- 각 객체 속성(property)은 name(key): value 형태를 가진다.
- 객체의 속성은 *object.name* 또는 *object["name"]* 으로 접근할 수 있다.
- for ... in 문으로 객체의 각 속성에 접근할 수 있다.
  - *for (변수 in 객체)*
- 객체의 속성에는 원시 값, 객체, 함수, 배열 등이 대입될 수 있다. 함수가 대입될 때 메소드라 한다.

# 객체

```
<html>
  <body>
    <script>
      function car(name, cost, born) {
        this.name = name;
        this.cost = cost;
        this.born = born;
      }

      var sonata = new car("sonata", 2000, 2018);
      var x;
      for(x in sonata)
        document.write(sonata.x + " ");

      document.write("<br> ");

      for(x in sonata)
        document.write(sonata[x] + " ");
    </script>
  </body>
</html>
```

# 객체 생성 – 객체 리터럴 방식

- 객체를 하나의 문구로 정의와 동시에 생성한다.

```
<html>
  <body>
    <script>
      function print_car(car)
      {
        document.write("--- " + car.name + " ---<br>");
        document.write("    color: " + car.bodyColor + "<br>");
        document.write("    cost: " + car.cost + "<br>");
        document.write("    born: " + car.born + "<br>");
        document.write("    age: " + car.age() + "<br><br>");
      }

      var sonata = {
        name: "sonata",
        bodyColor: "blue",
        cost: 2000,
        age: function(){return 2019-this.born;},
        born: 2009
      };

      print_car(sonata);
    </script>
  </body>
</html>
```

this 키워드는 객체 자신을 나타낸다.

# 객체 생성 – new Object()

- 키워드 new를 이용하여 변수를 객체로 지정하고 객체 속성은 이후 생성한다.

```
<html>
  <body>
    <script>
      function print_car(car)
      {
        document.write("--- " + car.name + " ---<br>");
        document.write("    color: " + car.bodyColor + "<br>");
        document.write("    cost: " + car.cost + "<br>");
        document.write("    born: " + car.born + "<br>");
        document.write("    age: " + car.age() + "<br><br>");
      }

      var sonata = new Object();
      sonata.name = "sonata";
      sonata.bodyColor = "blue";
      sonata.cost = 2000;
      sonata.born = 2009;
      sonata.age = function() {return 2019-this.born;};

      print_car(sonata);
    </script>
  </body>
</html>
```

# 객체 생성 – 객체 생성 함수

---

- 객체 생성 함수(Object constructor function)를 사용하면 여러 개의 동일한 객체를 선언할 수 있다.
- 객체 생성 함수는 new 키워드와 함께 호출된다.
- 객체 생성 함수를 통해 인자 값을 전달할 수도 있고, 인자 값을 전달하지 않을 수도 있다.
- 객체 생성 함수의 이름의 첫 단어는 가급적 대문자를 사용한다.
- 객체 생성 함수에서 this 키워드는 객체 자체를 나타낸다.

# 객체 생성 – 객체 생성 함수

```
<body>
  <script>
    function print_car(car)
    {
      document.write("--- " + car.name + " ---<br>");
      document.write("    color: " + car.bodyColor + "<br>");
      document.write("    cost: " + car.cost + "<br>");
      document.write("    born: " + car.born + "<br>");
      document.write("    age: " + car.age()+ "<br><br>");
    }
    function car(name, color, cost, born) {
      this.name = name;
      this.bodyColor = color;
      this.cost = cost;
      this.born = born;
      this.age = function(){return 2019-this.born;};
    }

    var sonata = new car;
    sonata.name = "sonata";
    sonata.bodyColor = "blue";
    sonata.cost = 2000;
    sonata.born = 2009;
    print_car(sonata);

    var audi = new car("audi","white",5000,2011);
    print_car(audi);
  </script>
</body>
```



# 객체 속성 추가

- prototype 키워드를 이용한 속성 추가
  - Prototype을 이용하여 속성을 추가하면 이전에 할당된 객체에도 새로운 속성이 추가된다.

```
--- audi ---  
color: white  
cost: 5000  
born: 2011  
age: 8  
--- sonata ---  
color: white  
cost: 2000  
born: 2009  
age: 10
```

```
<html>  
<body>  
  <script>  
    function print_car(car)  
    {  
      document.write("--- " + car.name + " ---<br>");  
      document.write("    color: " + car.bodyColor + "<br>");  
      document.write("    cost: " + car.cost + "<br>");  
      document.write("    born: " + car.born + "<br>");  
      document.write("    age: " + car.age() + "<br>");  
    }  
    function car(name, color, cost, born) {  
      this.name = name;  
      this.cost = cost;  
      this.born = born;  
    }  
  
    var sonata = new car;  
    sonata.name = "sonata";  
    sonata.cost = 2000;  
    sonata.born = 2009;  
  
    var audi = new car("audi", "white", 5000, 2011);  
    car.prototype.bodyColor = "white";  
    car.prototype.age = function(){return 2019-this.born;};  
  
    print_car(audi);  
    print_car(sonata);  
  </script>  
</body>  
</html>
```

# 객체 복사

- 객체 변수는 참조 값이므로 변수에 직접 복사가 안된다.

```
<html>
  <body>
    <script>
      function print_car(car)
      {
        document.write("--- " + car.name + " ---<br>");
        document.write("    color: " + car.bodyColor + "<br>");
        document.write("    cost: " + car.cost + "<br>");
        document.write("    born: " + car.born + "<br>");
        document.write("    age: " + car.age() + "<br><br>");
      }
      function car(name, color, cost, born) {
        this.name = name;
        this.bodyColor = color;
        this.cost = cost;
        this.born = born;
        this.age = function(){return 2019-this.born;};
      }

      var sonata = new car;
      sonata.name = "sonata";
      sonata.bodyColor = "blue";
      sonata.cost = 2000;
      sonata.born = 2009;

      var audi = sonata;
      audi.cost = 5000;
      print_car(sonata);
      print_car(audi);
    </script>
  </body>
</html>
```

--- sonata ---  
color: blue  
cost: 5000  
born: 2009  
age: 10

--- sonata ---  
color: blue  
cost: 5000  
born: 2009  
age: 10

# STRING 객체

# JavaScript String

---

- 문자열(string)은 두 개의 따옴표 또는 하나의 따옴표로 감싸서 나타낼 수 있다.

```
<p id="display"></p>  
<script>
```

```
var webc1 = "Digital Media";  
var webc2 = 'Digital Media';
```

```
var webc3 = 'Digital Media "Web contents"';  
var webc4 = "Digital Media 'Web contents'";
```

```
document.getElementById("display").innerHTML =  
webc1 + "<br>" + webc2 + "<br>" + webc3 + "<br>" + webc4;
```

```
</script>
```

# JavaScript String

---

- 문자열이 문장과 같이 긴 경우에는 코드 상의 가독성이 떨어지기 때문에 줄 바꿈을 해주는 것이 필요하다.
- ‘\n’을 이용해서 문자열의 연속성을 유지할 수 있다.
  - “Digital media in duksung” → “Digital media ( X ) → “Digital media \ ( △ )  
in duksung” in duksung”
- ‘\n’을 사용하는 것보다 확실한 방법은 문자열 연산을 이용하는 것이다. 하지만, 위의 방법보다 load가 있다.
  - “Digital media in duksung” → “Digital media” + “in duksung” ( O )

# JavaScript String

---

- 문자열 내에서 따옴표를 표시하기 위해서는 특수 문자 '₩'을 사용한다.
- ex) "Digital media in \"duksung\""

# JavaScript String

---

```
<p id="display"></p>
<script>
```

```
var webc1 = "Digital \
           Media";
```

```
var webc2 = 'Digital Media';
```

```
var webc3 = 'Digital Media \'Web contents\'';
```

```
var webc4 = "Digital Media \"Web contents\"";
```

```
document.getElementById("display").innerHTML =
webc1 + "<br>" + webc2 + "<br>" + webc3 + "<br>" + webc4;
```

```
</script>
```

# JavaScript String Method

---

- JavaScript에서는 문자열 변수도 객체로 인식하기 때문에 메소드(method)와 속성(properties)을 사용할 수 있다.
- 속성

Attribute	Description
length	문자열의 길이

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

```
<script>
```

```
var txt = "The Department of Digital Media in Duksung W.U";
```

```
document.getElementById("demo").innerHTML = txt.length;
```

```
</script>
```



# JavaScript String Method

- 메소드

Method	Description
<code>indexOf(string)</code> <code>search(string)</code>	문자열 안에서 특정 문자열이 처음으로 발견된 위치를 알려준다. 특정 문자가 발견되지 않는다면 -1을 반환한다.
<code>lastIndexOf(string)</code>	문자열 안에서 특정 문자열이 마지막으로 발견된 위치를 알려준다. 특정 문자가 발견되지 않는다면 -1을 반환한다.
<code>slice(from index, to index)</code> <code>slice(from index)</code>	인자로 받은 From index 위치부터 to index 위치까지 문자열을 잘라 넘겨준다. To index가 생략되면 from index 위치부터 끝까지의 문자열을 잘라 넘겨 준다. Index 값이 음수이면 마지막 index 부터의 순서로 index를 따진다.
<code>substring(from index, to index)</code>	<code>slice()</code> 메소드와 동일하다 index 값으로 음수 값을 사용하지 못한다.
<code>substr(from index, length)</code>	From index 부터 length 만큼의 문자열을 잘라 넘겨준다. Length가 생략되면 끝까지 잘라 넘겨준다.

---

```

<body>
<p id="len"></p>
<p id="indexOf"></p>
<p id="lastindexOf"></p>
<p id="lastindexOf2"></p>

<p id="slice1"></p>
<p id="slice2"></p>
<p id="slice3"></p>

<p id="substr"></p>

```

```

61
4
35
-1

Department
Departments in Duksung W.U
Media of Departments in Duksung W.U
Department

```

```

<script>
    var txt = "The Department of Digital Media of Departments in Duksung W.U";

    document.getElementById("len").innerHTML = txt.length;
    document.getElementById("indexOf").innerHTML = txt.indexOf("Department");
    document.getElementById("lastindexOf").innerHTML = txt.lastIndexOf("Department");
    document.getElementById("lastindexOf2").innerHTML = txt.lastIndexOf("Deparment");

    document.getElementById("slice1").innerHTML = txt.slice(4, 14);
    document.getElementById("slice2").innerHTML = txt.slice(35);
    document.getElementById("slice3").innerHTML = txt.slice(-35);

    document.getElementById("substr").innerHTML = txt.substr(4, 10);
</script>

</body>

```

# JavaScript String Method

---

- 메소드

Method	Description
<code>replace(from string, to string)</code>	문자열에서 특정 문자열을 찾아 원하는 문자열로 바꿔준다.
<code>toUpperCase()</code>	문자열을 대문자로 변환한다.
<code>toLowerCase()</code>	문자열을 소문자로 변환한다.
<code>concat(string, string, string, ...)</code>	String에 다른 String 들을 서로 합친다.
<code>charAt(index)</code>	Index 위치의 단어를 알려준다.
<code>charCodeAt(index)</code>	Index 위치의 단어를 ASCII 코드 값으로 알려준다.
<code>split(separator, max)</code>	<i>separator</i> 를 기준으로 문자열을 잘라 <i>max</i> 개수의 array로 저장한다. Max가 0인 경우 개수의 제한이 없다. 구분자가 “”일 때는 하나의 단어로 array가 만들어 진다. 구분자가 없을 때는 [0] 에 모든 문자열이 들어간다.

```

<body>
<p id="replace"></p>
<p id="toupper"></p>
<p id="tolower"></p>
<p id="concat"></p>
<p id="chat"></p>
<p id="chodat"></p>
<p id="split"></p>

```

```

<script>
    var txt = "The Department of Digital Media of Departments in Duksung W.U";
    var txt2 = "in Seoul, Korea";

    document.getElementById("replace").innerHTML = txt.replace("Department", "department");
    document.getElementById("toupper").innerHTML = txt.toUpperCase();
    document.getElementById("tolower").innerHTML = txt.toLowerCase();
    document.getElementById("concat").innerHTML = txt.concat(" + ", txt2);
    document.getElementById("chat").innerHTML = txt.charAt(4);
    document.getElementById("chodat").innerHTML = txt.charCodeAt(4);

    var ta = txt.split("of");
    for (var i=0; i < ta.length;i++)
    {
        document.getElementById("split").innerHTML += i+"th is "+ta[i]+" , ";
    }
</script>

</body>

```

The department of Digital Media of Departments in Duksung W.U

THE DEPARTMENT OF DIGITAL MEDIA OF DEPARTMENTS IN DUKSUNG W.U

the department of digital media of departments in duksung w.u

The Department of Digital Media of Departments in Duksung W.U + in Seoul, Korea

D

68

0th is The Department , 1th is Digital Media , 2th is Departments in Duksung W.U,

# NUMBER 객체

# JavaScript Number

---

- JavaScript에서 숫자는 모두 64비트 float 타입이다.
  - IEEE 754 표준



# JavaScript Number Method

---

- 속성

Properties	Description
Number.MAX_VALUE	JavaScript에서 사용 가능한 가장 큰 값 (1.7976931348623157e+308)
Number.MIN_VALUE	JavaScript에서 사용 가능한 가장 작은 값 (5e-324)
Number.NEGATIVE_INFINITY	
NaN	숫자가 아님.
Number.POSITIVE_INFINITY	

# JavaScript Number Method

---

- 메소드

Method	Description
<code>number.toString( )</code>	<code>number</code> 를 string으로 변환하여 넘겨준다. <code>number</code> 에는 연산자를 포함한 계산 식이 들어갈 수도 있다.
<code>number.toExponential(num)</code>	Exponential 형식으로 표현한다. 인자로 소수점 뒷자리 수를 지정할 수 있다.
<code>number.toFixed(num)</code>	<code>num</code> 만큼 소수점 이하 자리 수를 지정한다. 자리 수 이하의 값은 반올림한다.
<code>number.toPrecision( )</code>	전체 자리 수를 지정한다. 자리 수 이하의 값은 반올림한다.
<code>number.valueOf( )</code>	원시 값을 반환한다. 실제 코드에서는 사용할 일이 없음.



---

	123.456
<code>&lt;body&gt;</code>	1.2345600e+2
<code>&lt;p id="tostr"&gt;&lt;/p&gt;</code>	
<code>&lt;p id="toexp"&gt;&lt;/p&gt;</code>	123.46
<code>&lt;p id="toFixed"&gt;&lt;/p&gt;</code>	
<code>&lt;p id="toprec"&gt;&lt;/p&gt;</code>	123.5
<code>&lt;p id="valueOf"&gt;&lt;/p&gt;</code>	123.456
<code>&lt;script&gt;</code>	
<code>var x = 123.456;</code>	
<code>document.getElementById("tostr").innerHTML =</code>	<code>x.toString();</code>
<code>document.getElementById("toexp").innerHTML =</code>	<code>x.toExponential(7);</code>
<code>document.getElementById("toFixed").innerHTML =</code>	<code>x.toFixed(2);</code>
<code>document.getElementById("toprec").innerHTML =</code>	<code>x.toPrecision(4);</code>
<code>document.getElementById("valueOf").innerHTML =</code>	<code>x.valueOf();</code>
<code>&lt;/script&gt;</code>	
<code>&lt;/body&gt;</code>	

# JavaScript Number Method

---

- 숫자와 관련된 전역 메소드

Method	Description
Number( "string" )	String을 숫자로 변경한다. string에 숫자를 나타내는 문자 이외의 문자가 있으면 숫자로 변경하지 못한다.
parseInt( "string" )	String을 정수로 변경한다. string에 숫자를 나타내는 문자 이외의 포함되어 있다면 그 전까지의 숫자 문자까지만 변경한다.
parseFloat( "string" )	String을 소수점으로 변경한다. string에 숫자를 나타내는 문자 이외의 포함되어 있다면 그 전까지의 숫자 문자까지만 변경한다.
isNaN( "string" )	숫자가 아니라면 true, 숫자라면 false 이다.

```

<body>
  <p id="num"></p>
  <p id="numx"></p>
  <p id="prsint"></p>
  <p id="prsintx"></p>
  <p id="prsflt"></p>
  <p id="prsfltx"></p>
  <p id="isnan"></p>
  <p id="isnax"></p>

```

```

<script>

```

```

  document.getElementById("num").innerHTML = Number("123.456");
  document.getElementById("numx").innerHTML = Number("123.456 is number");
  document.getElementById("prsint").innerHTML = parseInt("123.456 is number");
  document.getElementById("prsintx").innerHTML = parseInt("number is 123.456");
  document.getElementById("prsflt").innerHTML = parseFloat("123.456 is number");
  document.getElementById("prsfltx").innerHTML = parseFloat("number is 123.456");
  if(!isNaN("123.456"))
    document.getElementById("isnan").innerHTML = "Number";
  if(isNaN("number is 123.456"))
    document.getElementById("isnax").innerHTML = "Not a number";

```

```

</script>

```

```

</body>

```

123.456

NaN

123

NaN

123.456

NaN

Number

Not a number

# ARRAY 객체

- 
- JavaScript의 Array에서는 서로 다른 type의 데이터를 하나의 array에 저장할 수 있다.

# JavaScript Array Method

- 메소드

Method	Description
<code>array.toString( )</code>	배열에 있는 내용 들을 하나의 string으로 생성한다.
<code>array.join( "separator")</code>	배열에 있는 내용들을 하나의 string으로 생성한다. separator 인자가 있다면 array index 값 간에 separator를 추가하여 생성한다. - toString은 separator를 추가할 수 없다.
<code>array.pop( )</code>	Array의 마지막 번째 항목을 제거하고, 제거된 항목의 값을 반환한다.
<code>array.push( "new")</code>	Array에 마지막 번째에 새로운 항목을 추가하고, array의 길이를 반환한다.
<code>array.shift( )</code>	모든 array 값들의 index를 하나씩 감소시킨다. 따라서, 이전의 array 첫 번째 값은 제거된다. 제거된 첫 번째 항목 값이 반환된다.
<code>array.unshift("new" )</code>	Array의 첫 번째에 새로운 항목을 추가하고 다른 항목들의 index는 하나씩 증가한다. array의 길이를 반환한다.

```

<body>
<script>
function prt_array (a)
{
    for (var i=0; i < a.length; i++)
        document.write(a[i] + ", ");

    document.write ("length: " + a.length + "<br>");
}

var hnum = ["one", "two", "three", "four", "five"];
var tnum = ["six", "seven", "eight", "nine", "ten"];

var ret = hnum.join(" and ");
document.write (ret + "<br>");
document.write ("<br><br>");

ret = hnum.pop();
document.write (ret + "<br>");
prt_array(hnum);
document.write ("<br><br>");

ret = hnum.push("six");
document.write (ret + "<br>");
prt_array(hnum);
document.write ("<br><br>");

ret = hnum.shift();
document.write (ret + "<br>");
prt_array(hnum);
document.write ("<br><br>");

ret = hnum.unshift("zero");
document.write (ret + "<br>");
prt_array(hnum);
document.write ("<br><br>");
</script>
</body>

```

one and two and three and four and five

five

one, two, three, four, length: 4

5

one, two, three, four, six, length: 5

one

two, three, four, six, length: 4

5

zero, two, three, four, six, length: 5

# JavaScript Array Method

- 메소드

Method	Description
<i>array.splice(position, remove count, new element, new element, new element, ... )</i>	새로운 항목을 array에 추가한다. Position 번째 위치부터 remove count만큼 array를 제거하고, 새로운 element를 그 위치에 추가 한다.
<i>array.splice(position, remove count)</i>	Position 번째 위치부터 remove count만큼 array 항목을 제거한다.
<i>array.concat(array1, array2, ... array n)</i>	Array 들을 서로 연결하여 새로운 array를 만든다.
<i>array.slice(from, to)</i>	from 이상 번째부터 to 미만 번째 element로 새로운 array를 생성한다.



```

<body>
  <script>
    function prt_array (a)
    {
      for (var i=0; i < a.length; i++)
        document.write(a[i] + ", ");

      document.write ("length: " + a.length + "<br>");
    }

    var hnum = ["one", "two", "three", "four", "five"];
    var tnum = ["six", "seven", "eight", "nine", "ten"];

    prt_array(hnum);           one, two, three, four, five, length: 5
                                one, four, five, length: 3
    hnum.splice(1, 2);         six,seven,eight,nine,ten, four, five, length: 3
    prt_array(hnum);           six,seven,eight,nine,ten, four, five, eleven, length: 4
                                six,seven,eight,nine,ten, four, five, eleven, twelve, length: 5
    hnum.splice(0, 1, tnum);   six,seven,eight,nine,ten, four, five, eleven, twelve, six, seven, eight, nine, ten, length: 10
    prt_array(hnum);

    hnum.splice(3, 1, "eleven");
    prt_array(hnum);

    hnum.splice(hnum.length, 0, "twelve");
    prt_array(hnum);

    var nnum = hnum.concat(tnum);
    prt_array(nnum);

  </script>
</body>

```

# JavaScript Array Method

---

- 메소드

Method	Description
<code>array.sort()</code>	Array의 항목들을 알파벳 순으로 정렬한다.
<code>array.reverse()</code>	Array의 항목들을 알파벳 역순으로 정렬한다.

- sort() 메소드는 array 항목들이 숫자로 되어 있어도 문자열로 변환한 후 알파벳 순으로 정렬한다. 따라서, 숫자의 올림차수나 내림차수 정렬은 어렵다.
  - 이 경우 sort에 compare 함수를 인자로 주면 숫자의 올림차수 정렬이 가능하다.

# JavaScript Array Method

---

- 메소드

Method	Description
<code>array.sort(compare function(a, b))</code>	<p>Compare 함수를 적용하고 compare 함수로 부터 return되는 값에 따라 정렬한다.</p> <p>Compare function의 반환 값이 양수이면 a가 b보다 큰 것으로 인식하고, 음수이면 b가 a보다 큰 것으로 인식한다. 만약, 0이면 두 수는 같은 것으로 인식한다.</p>

# Array sort function

```
<body>
  <script>
    var target = new Array(4, 11, 2, 10, 3, 1, 43);
    var sorted;

    sorted = target.sort(CompareForSort);
    document.write("<br/>");
    document.write(sorted);
    document.write("<br/>");

    function CompareForSort(a, b)
    {
      var ret;
      if (a == b)
      {
        document.write (" is same" + "<br/>");
        return 0;
      } if (a > b) {
        document.write (b + " is bigger than " + a + "<br/>");
        return 1;
      } else {
        document.write (a + " is bigger than " + b + "<br/>");
        return -1;
      }
    }
  </script>
</body>
```

4 is bigger than 11  
2 is bigger than 11  
2 is bigger than 4  
10 is bigger than 11  
4 is bigger than 10  
3 is bigger than 11  
3 is bigger than 10  
3 is bigger than 4  
2 is bigger than 3  
1 is bigger than 11  
1 is bigger than 10  
1 is bigger than 4  
1 is bigger than 3  
1 is bigger than 2  
11 is bigger than 43

1,2,3,4,10,11,43

DATE

- 
- UNIX에서는 1970년 1월 1일 0시 0분 0초 UTC를 기준으로 경과한 시간을 초로 환산하여 사용한다.
  - GMT (그리니치 표준 시)
    - 영국 그리니치 천문대를 기준으로 경도에 따라 시간대를 나눈 시간.
    - 강압적이지 않고 그 나라의 사정에 따라 변동하여 사용한다.
  - UTC (세계 협정 시 - Universal Time, Coordinated)
    - 1972년 1월 1일에 제정한 세계 표준 시.
  - 우리나라 시간은 UTC +9 시간이다.
    - 일본의 기준시간과 동일함.
    - 실제 우리나라의 시간은 UTC +8 이 보다 정확함.
    - <http://sydaby.eget.net/swe/tzones.htm>

---

- 날짜 보여주기

Date objects	Description
Date()	현재 날짜와 시간을 알려준다.
new Date()	날짜 객체를 생성한다.
new Date(milliseconds)	초기 시간에 milliseconds를 더한 시간이 저장된 날짜 객체를 생성한다.
new Date(date String)	지정한 특정 시간이 저장된 날짜 객체를 생성한다.
new Date(year, month, day, hour, minutes, seconds, milliseconds)	각각의 시간 영역을 지정하여 날짜 객체를 생성할 수 있다.

---

Method	Description
getDate()	현재 날짜를 알려준다. 1~ 31 일 범위.
getDay()	일요일~토요일(0~7)까지 중 현재 요일을 알려준다. 일요일: 0, 월요일: 1, ... 토요일: 7
getFullYear()	현재의 연도를 4자리 수로 알려준다.
getHour()	현재 시간을 알려준다. 0 ~ 23 시 범위.
getMilliseconds()	현재 시간 중 millisecond 을 알려준다. 0 ~ 999 범위.
getMinutes()	현재 시간 중 분 영역을 알려준다. 0 ~ 59 범위.
getMonth()	현재 시간 중 월 영역을 알려준다. 0 ~ 11 범위.
getSeconds()	현재 시간 중 초 영역을 알려준다. 0 ~ 59 범위.
getTime()	초기 시간 이후의 지난 시간을 milliseconds단위로 넘겨준다.



---

Method	Description
setDate()	현재 날짜에 지정한 일로 바꾸어 보여 준다. (실제 날짜가 바뀌는 것은 아니다.)
setDay()	현재 날짜에 지정한 요일의 숫자로 바꾸어 준다.
setFullYear()	현재 날짜에 지정한 년, 월, 일로 바꾸어 준다.
setHour()	현재 날짜에 지정한 시간으로 바꾸어 준다. (시간의 범위는 0~23이다.)
setMilliseconds()	현재 날짜에 지정한 milliseconds만큼 더한 시간으로 바꾸어 준다.
setMinutes()	현재 날짜에 지정한 분 만큼을 더한 시간으로 바꾸어 준다.
setMonth()	현재 날짜에 지정한 월로 바꾸어 준다. (월의 범위는 0 ~ 11이다.)
SetSeconds()	현재 날짜에 지정한 초로 바꾸어 준다.(초의 범위는 0 ~ 59이다.)
setTime()	초기 시간에 milliseconds 시간 만큼 더한 시간으로 바꾸어 준다.

END