

7강. 객체(Object)



목 차

1

내장 객체

2

사용자 정의 객체

3

생성자 함수를 사용한 객체

객체의 정의

● 객체(Object)란?

- 프로그램에서 인식할 수 있는 모든 대상
- 데이터를 저장하고 처리하는 기본 단위.

◎ 자바 스크립트 내장 객체

문서 객체 모델(DOM) : 문서 뿐만 아니라 웹 문서 안에 포함된 이미지, 링크, 텍스트 필드 등을 모두 별도의 객체로 관리

브라우저 객체 모델(BOM) : 웹 브라우저 정보를 객체로 관리

◎ 사용자 정의 객체

여러 가지 자료형을 포함하는 '**복합**' **자료형**을 직접 만들어 사용하는 것을 말한다.

내장 객체 - Math

Math - 수학, 통계 관련 함수들을 내장하고 있음

함수	설명
abs(x)	숫자의 절대값을 반환합니다.
round(x)	숫자의 소수점 이하를 반환합니다.(정수로 반환)
floor(x)	숫자의 소수점 이하를 버립니다.
pow(x, y)	거듭제곱수 구하기 - x의 y제곱을 반환합니다.
random()	0과 1사이의 무작위 수(난수)를 반환합니다.

내장 객체 - Date

MDN > Technology > JavaScript > Built-in objects > Math

 MDN Web Docs
moz://a

▶ Technologies ▶ References & Guides

개발자를 위한 웹 기술 > JavaScript

이 페이지는 영어로부터 커뮤니티에 의하여 번역되었습니다. MDN Web Docs에서 한국 커뮤니티

Table of contents

- 자습서
- 참고서
- 도구 & 자원

JavaScript

JavaScript (JS)는 경량, 인터프리터 혹은 [just-in-time](#) 컴파일러를 가진 웹 페이지를 위한 스크립트 언어로 잘 알려져 있다. 웹 페이지를 위한 스크립트 언어로 잘 알려져 있으며 [많은 비 브라우저 환경](#)에서도 사용하고 있다. 스레드, 동적 언어로, 객체지향형, 명령형, 선언형, [JavaScript에 대하여](#)를 참고하세요.

▼ Built-in objects

[AggregateError](#)

[Array](#)

[ArrayBuffer](#)

[AsyncFunction](#)

[Atomics](#)

[BigInt](#)

[BigInt64Array](#)

[BigUint64Array](#)

[Boolean](#)

[DataView](#)

[Date](#)

내장 객체 - Date

MDN > Technology > JavaScript > Built-in objects > Math

Math.pow(x, y)

x의 y 제곱을 반환합니다.

Math.random()

0과 1 사이의 난수를 반환합니다.

Math.round(x)

숫자에서 가장 가까운 정수를 반환합니다.

```
Math.round( 20.49); // 20
Math.round( 20.5 ); // 21
Math.round( 42    ); // 42
Math.round(-20.5 ); // -20
Math.round(-20.51); // -21
```

```
Math.pow(7, 2);    // 49
Math.pow(7, 3);    // 343
Math.pow(2, 10);   // 1024
```

내장 객체 - Date

```
> Math.PI
< 3.141592653589793
> Math.abs(-3)
< 3
> Math.abs(-3.3)
< 3.3
> Math.floor(2.54)
< 2
> Math.floor(-6.3)
< -7
> Math.random()
< 0.7657452729807372
> Math.random()*6
< 5.708358771977116
> Math.floor(Math.random()*6)+1
< 5
```

내장 객체 - Math

절대값과 거듭 제곱함수 만들기

```
function myAbs(x){  
  if(x < 0)  
    return -x;  
  else  
    return x;  
}  
//x=-5, -(-5), 5  
//x=5, 5  
  
function myPow(x, y){ //x:밑, y:지수  
  var num = 1; //결과값  
  for(var i = 0; i < y; i++){  
    num *= x;  
  }  
  return num;  
}  
// i=0, 0<3, 1*2  
// i=1, 1<3, 2*2  
// i=2, 2<3, 4*2(2*2*2)
```

//내장 객체와 비교하기

```
document.write(Math.abs(-5) + "<br>");  
document.write(myAbs(-5)+ "<br>");
```

```
document.write(Math.pow(2, 3)+ "<br>");  
document.write(myPow(2, 3)+ "<br>");
```


내장 객체 - Math

절대값 - Math.abs() 사용하기

```
<script>
  //두 수를 입력받아 절대값을 구하는 함수
  var difference = function(a, b){
    |   return Math.abs(a - b);
    |
    }

  res1 = difference(4, 7);
  console.log("res1 = " + res1);

  res2 = difference(7, 4);
  console.log("res2 = " + res2);

  res3 = difference(2.54, 1.23);
  console.log("res3 = " + res3);

</script>
```

res1 = 3
res2 = 3
res3 = 1.31


내장 객체 - Math

무작위수 만들기 - **Math.floor(Math.random())**

```
<script>
  //1~10 자연수 중 무작위수
  var rand = Math.floor(Math.random()*10)+1;
  console.log(rand);

  //주사위 10번 던지기
  var dice = function(){
    return Math.floor(Math.random()*6)+1;
  }

  for(var i=1; i<=10; i++){
    console.log(dice());
  }
</script>
```



5
6
1
5
1
2
5
2 4
2

내장 객체 - Math

주사위 던지기 게임

```
<script>
  //주사위 2개를 10번 던지기
  for(var i=1; i<=10; i++){
    var dice1 = Math.floor(Math.random()*6) + 1;
    var dice2 = Math.floor(Math.random()*6) + 1;
    var total = dice1 + dice2;
    console.log(total);
    if(total===8)
      console.log("Eight Thrown!");
    if(total===11)
      console.log("Eleven Thrown!");
    if(dice1===dice2)
      console.log("Double Thrown");
  }
</script>
```

9	JavaScript
4	
Double Thrown	
2 9	
8	
Eight Thrown!	
2 10	
2 5	

내장 객체 - Math

버튼 눌러 주사위 던지기

주사위

2

```
body{text-align: center;}
button{margin-top: 30px; padding: 10px 20px; font-size: 1.3rem;}
p{font-size: 1.5rem;}
```

```
<button onclick="dice()">주사위</button>
<p id="throw"></p>
<script>
  function dice(){
    var result = Math.floor(Math.random()*6) + 1;
    // document.getElementById("throw").innerHTML = result;

    //querySelector()로 대체 가능 -> class는 ".throw"
    document.querySelector("#throw").innerHTML = result;
  }
</script>
```

이벤트 당첨자 추첨 프로그램

이벤트 당첨자 추첨 프로그램

당첨자 발표

전체 응모자 수 : 100명

당첨자 : 18번

```
<h1>당첨자 발표</h1>
```

```
<script>
```

```
var member = prompt("전체 응모자 수 : ");
```

```
var picked = Math.floor(Math.random() * member) + 1;
```

```
document.write("<p>전체 응모자 수 : " + member + "명</p>");
```

```
document.write("<p>당첨자 : " + picked + "번</p>");
```

```
</script>
```

```
body{text-align: center;}  
h1{  
  border: 1px solid #ccc;  
  display: inline-block;  
  padding: 10px;  
  background-color: #eee;  
}  
p{font-size: 1.2em;}
```

이벤트 당첨자 추첨 프로그램

이벤트 당첨자 추첨 프로그램(여러명)

당첨자 발표

전체 응모자 수 : 100명

당첨자 : 82번, 31번, 11번, 96번, 60번

```
<h1>당첨자 발표</h1>
<script>
  var member = prompt("전체 응모자 수 : ");
  var winner = prompt("당첨자 수 : ");

  document.write("<p>전체 응모자 수 : " + member + "명</p>");
  document.write("<p>당첨자 : ");
  for(var i = 1; i <= winner; i++){
    var picked = Math.floor(Math.random() * member) + 1;
    if(i < winner){
      document.write(picked + "번, ");
    }else{
      document.write(picked + "번");
    }
  }
</script>
```

실습 문제

좋은 글귀를 배열에 저장해 두고 무작위로 출력하는 프로그램

성공하는 사람은 실패하는데 익숙한 사람이다.

```
// 좋은 글귀
var words = []

words[0] = "당신은 지금도 최고고, 이전에도 최고였으며 앞으로도 최고일 것이다";
words[1] = "성공하는 사람은 실패하는데 익숙한 사람이다.";
words[2] = "후회를 최대한 이용하라. 깊이 후회한다는 것은 새로운 삶을 산다는 것이다.";
words[3] = "가짜 친구는 소문을 믿고, 진짜 친구는 나를 믿는다.";
words[4] = "성공이라는 뜻을 박으려면 끈질김이라는 망치가 필요하다.";

var rand = Math.floor(Math.random()*words.length);

document.write(words[rand]);
```

실습 문제

words.html

```
<link href="https://fonts.googleapis.com/css2?family=Nanum+Pen+Script&display=swap"
      rel="stylesheet">
<style>
  body{
    margin-top: 50px;
    font-size: 2.5em;
    text-align: center;
    font-family: 'Nanum Pen Script', 바탕;
    background: ■rgb(1, 1, 151);
    color: □white;
  }
</style>
</head>
<body>
  <script src="../js/word.js"></script>
</body>
```


내장 객체 - Date

인스턴스 메서드

Date.prototype.getDate()

Date 에서 현지 시간 기준 일(1-31)을 반환합니다.

Date.prototype.getDay()

Date 에서 현지 시간 기준 요일(0-6)을 반환합니다.

Date.prototype.getFullYear()

Date 에서 현지 시간 기준 연도(네 자리 연도면 네 자리로)를 반환합니다.

Date.prototype.getHours()

Date 에서 현지 시간 기준 시(0-23)를 반환합니다.

Date.prototype.getMilliseconds()

Date 에서 현지 시간 기준 밀리초(0-999)를 반환합니다.

Date.prototype.getMinutes()

Date 에서 현지 시간 기준 분(0-59)을 반환합니다.

Date.prototype.getMonth()

Date 에서 현지 시간 기준 월(0-11)을 반환합니다.

Date.prototype.getSeconds()

Date 에서 현지 시간 기준 초(0-59)를 반환합니다.

Date.prototype.getTime()

1970년 1월 1일 00:00:00 UTC로부터의 경과시간을 밀리초 단위로 반환합니다. Date 가 기준 시간 이전을 나타낼 경우 음수 값을 반환합니다.

내장 객체 - Date

Date 객체

- 날짜와 시간 정보를 다루는 객체
- **new Date()**로 인스턴스를 만들 수 있다.

```
> var now = new Date()
< undefined
> now
< Sun Jun 13 2021 18:47:13 GMT+0900 (대한민국 표준시)
> now.toLocaleString()
< "2021. 6. 13. 오후 6:47:13"
> now.getFullYear()
< 121
> now.getFullYear()
< 2021
> now.getMonth()
< 5
> now.getDate()
< 13
> now.getDay()
< 0
```

요일
(0-일,1-월,2-화,3-수,4-목,5-금,6-토)

내장 객체 - Date

날짜/시간을 알려주는 Date 객체

===== 날짜 =====

현재 시각은 Fri Aug 27 2021 04:31:42 GMT+0900 (한국 표준시)

현재 시각은 2021. 8. 27. 오전 4:31:42

현재 년도 : 2021

현재 월 : 8

현재 일 : 27

현재 요일 : 5

===== 시간 =====

현재 시각은 오전 4:31:42

현재까지 시간 : 1630006302.181초

현재 시: 4

현재 분: 31

현재 시: 42

```
var now = new Date();
document.write("===== 날짜 =====<br>");
document.write("현재 시각은 " + now + '<br>');
document.write("현재 시각은 " + now.toLocaleString() + '<br>');

document.write("현재 년도 : " + now.getFullYear() + '<br>');
document.write("현재 월 : " + (now.getMonth() + 1) + '<br>');
document.write("현재 일 : " + now.getDate() + '<br>');
document.write("현재 요일 : " + now.getDay() + '<br>');

document.write("<br>===== 시간 =====<br>");
document.write("현재 시각은 " + now.toLocaleTimeString() + '<br>');
//1970년 1월 1일 자정 이후 ~ 지금
document.write("현재까지 시간 : " + now.getTime()/1000 + '초<br>');
document.write("현재 시: " + now.getHours() + '<br>');
document.write("현재 분: " + now.getMinutes() + '<br>');
document.write("현재 시: " + now.getSeconds() + '<br>');
```

내장 객체 - Date

특정한 날 이후부터 현재까지의 날 수 계산하기.

```
var startDate = new Date("2021-10-26")
undefined

startDate.getTime()
1635206400000

var now = new Date()
undefined

now
Mon Nov 15 2021 23:33:55 GMT+0900 (한국 표준시)

now - startDate
1780435713

(now - startDate) / (24*60*60*1000)
20.60689482638889

var passedTime = (now - startDate) / (24*60*60*1000)
undefined

Math.round(passedTime)
21
```

날짜 계산 프로그램

날짜 계산 프로그램 만들기

지금까지 몇 일?

입사 이후 **17**일 지났습니다.

수고하셨습니다.

날짜 계산 프로그램

날짜 계산 프로그램 만들기

```
<div id = "container">
  <h2>지금까지 몇 일?</h2>
  <p>입사 이후 <span id="day" class="accent"></span>일 지났습니다. <br>
  | 수고하셨습니다. </p>
</div>
<script>
  var now = new Date();
  var day1 = new Date("2021-8-10");

  var passedTime = now.getTime() - day1.getTime();
  passedTime = Math.round(passedTime/(24*60*60*1000));
  //document.write(passedTime);

  document.getElementById('day').innerHTML = passedTime;
</script>
```

```
#container{
  width: 300px;
  height: 300px;
  border: 2px solid ■ #333;
  border-radius: 50%;
  margin: 50px auto;
  text-align: center;
  background-color: □ rgb(238, 250, 220);
}
h2{padding-top: 60px;}
p{line-height: 2;}
.accent{
  color: ■ red;
  font-size: 1.5em;
  font-weight: 700;
}
```

내장 객체 - Date

indexOf() 함수 예제

127.0.0.1:5500 내용:

입력해주세요

확인취소

127.0.0.1:5500 내용:

안녕하세요

확인

단어를 검색하면 대답하는 프로그램

- '안녕'이 있으면 '안녕하세요' 출력
- 시간을 검색하면 '현재 시간' 출력
- '잘있어' 또는 '잘가'를 검색하면
"안녕히 가세요"를 출력하고 프로그램
이 종료됨
- 찾는 단어가 없으면 "모르는 단어입니다" 출력

내장 객체 - Date

indexOf()함수 예제

```
while(true){
    var input = prompt("입력해 주세요");
    if(input.indexOf("안녕") >= 0){
        alert("안녕하세요");
    }
    else if(input.indexOf("시") >= 0 || input.indexOf("분") >= 0){
        var now = new Date();
        var hours = now.getHours();
        var minute = now.getMinutes();
        alert("현재 시각은 " + hours + "시 " + minute + "분 입니다.");
    }
    else if(input.indexOf("잘 가") >= 0 || (input.indexOf("잘 있어") >= 0)){
        alert("안녕히 가세요.");
        break;
    }
    else{
        alert("모르는 단어입니다.");
    }
}
```


시간 설정하기

window 객체 – setInterval(), setTimeout() 함수

- setInterval(in milliseconds) : 초가 설정되고 계속 반복함.

```
<h3>3초 후에 알림창이 뜨고, 1초에 한번 계속 창이 뜹니다.</h3>
<script>
  setInterval( //실행 함수
    function(){
      alert("안녕하세요~")
    }, 3000
  );

  /*
  setInterval(winHello, 3000); //일반함수 정의

  function winHello(){
    alert("Hello~ ");
  }*/
</script>
```

시간 설정하기

window 객체 – setInterval(), setTimeout() 함수

- setTimeout() : 초가 1번 설정되고 종료함

```
<h3>3초 후에 알림창이 뜨고, 확인 하면 종료합니다.</h3>
<script>
  setTimeout(
    function(){
      alert("환영합니다.")
    }, 3000
  );
</script>
```

디지털 시계 만들기

- 시계 만들기

오전 5:04:24

```
<p id="display" class="access"></p>
<script>
  setInterval(watch, 1000); //1초마다 watch() 작동

  function watch(){
    var date = new Date(); //Date 객체의 인스턴스 생성
    var now = date.toLocaleTimeString();
    document.getElementById('display').innerHTML = now;
  }
</script>
```

객체의 정의

◎ 사용자 정의 객체

- 여러가지 자료형을 포함하는 '복합' 자료형을 말한다.
- 객체는 속성과 메서드로 구성되어 있다.

속성(property)

나이 : 39
이름 : 추신수
결혼유무 : true
자녀수 : 3

메서드(function)

야구를 한다.



```
var age = 38;  
var name = "추신수";  
var isMerried = true;  
var numberOfChildren = 3;
```

```
function play(){  
    "야구를 한다 "  
}
```

사람

var person = { }

객체의 정의 및 사용

사용자 정의 객체(object) 정의

```
var 객체이름 = {  
  속성 이름 : 값,  
  속성 이름 : 값,  
  함수 이름 : function(){...}  
}
```

콜론(:)사용

콤마(,)사용

키(key) : 값(value) 형태

//person 객체 만들기

```
var person = {  
  name : "추신수",  
  age : 38,  
  isMerried : true,  
  numberOfChildren : 3,  
  
  play : function(){  
    alert("홈런을 친다!!")  
  }  
}
```

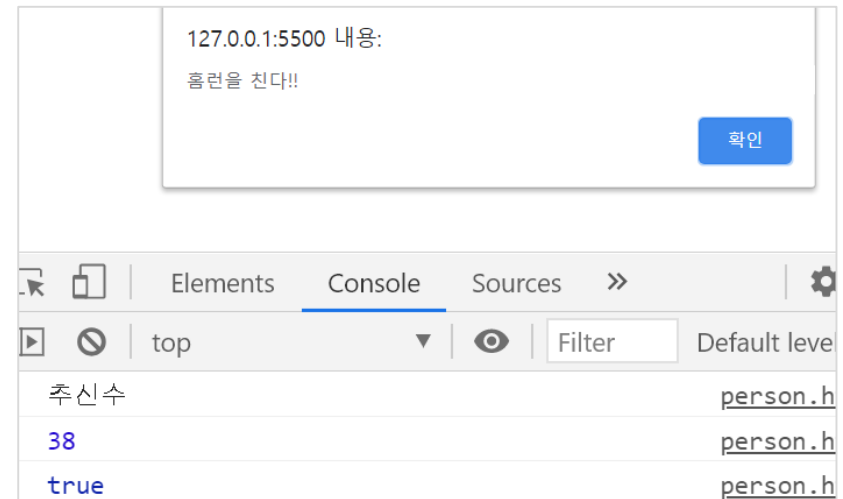
객체의 정의 및 사용

person 객체 사용

객체 이름.속성(함수)

```
//person 객체 사용하기  
console.log(person.name);  
console.log(person.age);  
console.log(person.isMerried);  
console.log(person.play());
```

점(.) 연산자로 접근



객체의 정의 및 사용

객체(object) 정의 및 사용

```
> var personSon = {  
  name : '손흥민',  
  age : 28,  
  play : function(){  
    console.log("Play Soccer!!");  
  },  
}  
⏏ undefined  
  
> personSon  
⏏ ▶ {name: "손흥민", age: 28, play: f}  
  
> personSon.name  
⏏ "손흥민"  
  
> personSon.play  
⏏ f () {  
    console.log("Play Soccer!!");  
  }  
  
> personSon.play()  
Play Soccer!!
```

인덱싱
키를 주어서 값알기

```
> personSon["name"]  
⏏ "손흥민"  
  
> personSon["age"]  
⏏ 28  
  
> personSon["play"]  
⏏ f () {  
    console.log("Play Soccer!!");  
  }
```

this 키워드 사용

this 키워드 사용

- 객체 내부의 속성을 사용할 때 **this** 키워드를 붙여야 함

```
<script>
  var book = {
    title : "HTML5+CSS3",
    author : "고경희",
    pages : 500,
    price : 30000,

    info : function(){ //멤버 속성 접근은 this로 한다.
      alert(this.title + " 책의 분량은 " + this.pages + "페이지 입니다.");
    }
  }

  //document.write(book.title + "의 가격은 " + book.price + "원입니다.");
  book.info(); //book 객체로 접근
</script>
```


생성자 함수를 사용한 객체

생성자 (Constructor) 함수를 사용해 객체 만들기

- 필요한 속성과 함수를 틀(클래스)처럼 미리 만들어 놓은 객체

```
<script>
function Person(name, age, isMerried){ //생성자 함수(클래스)
    this.name = name;
    this.age = age;
    this.isMerried = isMerried;

    this.info = function(){
        if(this.isMerried==true){
            document.write(this.name + "는(은) 기혼입니다<br>");
        }
        else{
            document.write(this.name + "는(은) 미혼입니다<br>");
        }
    }
}
}
```

생성자 함수이름의 첫글자는 대문자

생성자 함수를 사용한 객체

- 객체의 인스턴스 만들기

new 객체명

인스턴스라 함 : 메모리 heap 영역에 위치

var **person** = new Person()

// Person 객체 생성하기

```
var chu = new Person("추신수", 39, true);  
document.write("이름 : " + chu.name + "<br>");  
document.write("나이 : " + chu.age + "<br>");  
chu.info();
```

```
var son = new Person('손흥민', 29, false);  
document.write("이름 : " + son.name + "<br>");  
document.write("나이 : " + son.age + "<br>");  
son.info();
```

</script>

이름 : 추신수
나이 : 39
추신수는(은) 기혼입니다
이름 : 손흥민
나이 : 29
손흥민는(은) 미혼입니다

원의 둘레와 면적 구하기

circle 생성자 함수를 정의하고 수를 입력 받아 원의 둘레와 면적을 계산하는 프로그램을 작성하세요

```
function Circle(radius){ //생성자 함수
    this.radius = radius;

    this.length = function(){ //원의 둘레 = 2 x PI x 반지름
        var length = Math.floor(2 * Math.PI * this.radius);
        return length;
    }

    this.area = function(){ //원의 면적 = PI x 반지름 x 반지름
        var area = Math.floor(Math.PI * this.radius * this.radius);
        return area;
    }
}

var radius = prompt("원의 반지름은? (cm)");

var circle = new Circle(radius);

document.write("원의 둘레 : " + circle.length() + "cm<br>");
document.write("원의 면적 : " + circle.area() + "cm<sup>2</sup>")
```

도서 목록 만들기

생성자 함수로 도서 목록 객체 만들기

책 제목으로 살펴보기

HTML5+CSS3+JavaScript

점프 투 파이썬

천개의 파랑

도서 목록 만들기

생성자 함수로 도서 목록 객체 만들기

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Object</title>
  <link rel="stylesheet" href="css/book.css">
  <script src="js/book.js"></script>
</head>
<body>
  </body>
```

```
body{
  padding: 20px;
}
h1{margin-top: 50px; padding-left: 10px; color: blue}
p{
  width: 300px;
  border: 1px solid #ccc;
  padding: 10px 20px;
  margin-bottom: 20px;
}
```

도서 목록 만들기

```
// 생성자 함수 - Book 만들기
function Book(title, author, price){
    this.title = title;
    this.author = author;
    this.price = price;
}
//Book 객체 생성
var web = new Book("HTML5+CSS3+JavaScript", "고경희", 30000);
var python = new Book("점프 투 파이썬", "박응용", 20000);
var robot = new Book("천개의 파랑", "천선란", 14000);

//생성된 객체를 배열에 저장
var bookList = [web, python, robot];

//bookList[0] = web
//bookList[0].title = "HTML5+CSS3+JavaScript"

document.write("<h1>책 제목으로 살펴보기</h1>");
for(var i=0; i<bookList.length; i++){
    document.write("<p>" + bookList[i].title + "</p>");
}
```

book.js