

# 6강. 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.  
객체 지향 언어에서는 클래스(class)라고도 하며 객체의 설계도, 틀 역할을 한다.

# 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 - Math

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

## Related Topics

# 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 환경에서도 사용하고 있다. 스레드, 동적 언어로, 객체지향형, 명령형, 선언형은 JavaScript에 대하여를 참고하세요.

## ▼ Built-in objects

[AggregateError](#)

[Array](#)

[ArrayBuffer](#)

[AsyncFunction](#)

[Atomics](#)

[BigInt](#)

[BigInt64Array](#)

[BigUint64Array](#)

[Boolean](#)

[DataView](#)

[Date](#)

# 내장 객체 - Math

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 - Math

## ■ Console에서 실행하기

```
> 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)
```

math\_ex.html

```
//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 자연수중 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

전체 응모자 수 :

확인

취소

### 당첨자 발표

전체 응모자 수 : 20명

당첨자 : 18번

# 이벤트 당첨자 추첨 프로그램

picked\_one.html

```
<section>
  <h1>당첨자 발표</h1>
</section>

<script>
  let member = prompt("전체 응모자 수 : ");

  //추첨
  picked_one = Math.floor(Math.random() * member) + 1;

  //발표
  document.write("<p>전체 응모자 수 : " + member + "명</p>");
  document.write("<p>당첨자 : ")
  document.write(picked_one + "번</p>");
</script>
```

```
section{
  width: 800px;
  margin: 0 auto;
  text-align: center;
}
h1{
  background-color: #eee;
  border: 1px solid #ccc;
  display: inline-block;
  padding: 10px 20px;
}
p{text-align: center;}
```

# 이벤트 당첨자 추첨 프로그램

## ◆ 이벤트 당첨자 추첨 프로그램(여러명)

### 당첨자 발표

전체 응모자 수 : 100명

당첨자 : 82번, 31번, 11번, 96번, 60번

# 이벤트 당첨자 추첨 프로그램

## ◆ 이벤트 당첨자 추첨 프로그램(여러명)

```
let member = prompt("전체 응모자 수 : ");
let winner = prompt("당첨자 수 : ");

if(member == null || winner == null){
    document.write("<p>입력이 취소되었습니다.");
}
else{
    document.write("<p>전체 응모자 수 : " + member + "명</p>");
    document.write("<p>당첨자 : ");

    for(let i = 1; i <= winner; i++){
        let picked_one = Math.floor(Math.random()*member) + 1; //당첨 번호
        if(i < winner)
            document.write(picked_one + "번, ");
        else
            document.write(picked_one + "번 ");
    }
}
```

picked\_winners.html



# 이벤트 당첨자 추첨 프로그램

## ◆ 이벤트 당첨자 추첨 프로그램(배열로 구현)

```
let member = prompt("전체 응모자 수 : "); //응모자 수
let win = prompt("당첨자 수 : ");
//let winner = new Array(5); //5개의 배열 공간 생성
let winner = new Array(Number(win)); //입력된 문자 win을 숫자로 변경
```

```
if(member == null || win == null){
    document.write("<p>입력이 취소되었습니다</p>");
}
```

picked\_winners2.html

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

    //랜덤 번호 추첨 - 5번
    for(let i = 0; i < winner.length; i++){
        let num = Math.floor(Math.random() * member) + 1; //랜덤 번호
        winner[i] = num;
    }

    //로또 번호 출력
    for(let i = 0; i < winner.length; i++){
        document.write(winner[i] + "번, ");
    }
}
```

# 로또(lotto) 추첨 프로그램

## ◆ 로또(lotto) 추첨 프로그램(배열로 구현)

로또 추첨 번호

번호 : 5, 25, 26, 41, 25, 8, → 중복이 없어야 함.

# 로또(lotto) 추첨 프로그램

```
let lotto = new Array(6); //로또 배열 생성

//랜덤 번호 추첨
document.write("<p>번호 : ");
for(let i = 0; i < lotto.length; i++){
    let num = Math.floor(Math.random() * 45) + 1; //랜덤 번호
    lotto[i] = num; //로또 배열에 저장
    console.log(lotto);

    //중복 제거
    for(let j = 0; j < i; j++){
        if(lotto[j] == lotto[i]){
            console.log("중복 번호 : " + lotto[i])
            i--; //중복된 인덱스 삭제함
        }
    }
}

//출력
for(let i = 0; i < lotto.length; i++){
    document.write(lotto[i] + ", ");
}
```

lotto.html

# Math 객체 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]);
```

# Math 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

## Date 객체

- 날짜와 시간 정보를 다루는 객체
- **const date = new Date()**로 인스턴스를 만듦.

함수	설명
getFullYear()	연도를 4자리 숫자로 표시함
getMonth()	0부터 11사이의 숫자로 월을 표시함(0- 1월)
getDate()	1~31 사이의 숫자로 일을 표시함
getDay()	0~6 사이의 숫자로 요일을 표시함(0-일요일)
getHours()	0~23 사이의 숫자로 시를 표시함
getMinutes()	0~59 사이의 숫자로 분을 표시함
getSeconds()	0~59 사이의 숫자로 초를 표시함
getTime()	1970년 1월 1일 이후의 시간을 밀리 초로 표시함

# 내장 객체 - Date

## Date 객체

현재 날짜와 시간 표시

2022. 1. 19. 오전 5:58:58

```
<h2>Date 객체</h2>
```

```
<p>현재 날짜와 시간 표시</p>
```

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

```
<script>
```

```
    const date = new Date();
```

```
    //document.getElementById("demo").innerHTML = date; //날짜, 시간
```

```
    document.getElementById("demo").innerHTML = date.toLocaleString() //시간
```

```
    //document.getElementById("demo").innerHTML = date.toLocaleDateString(); //날짜
```

```
    //document.getElementById("demo").innerHTML = date.toLocaleTimeString(); //시간
```

```
</script>
```

# 내장 객체 - Date

## Date 객체

특정한 날짜와 시간 표시

2021. 12. 26. 오전 9:00:00

```
<h2>Date 객체</h2>
```

```
<p>특정한 날짜와 시간 표시</p>
```

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

```
<script>
```

```
    const date = new Date("2021-12-26"); //특정한 날짜
```

```
    //document.getElementById("demo").innerHTML = date; //날짜, 시간
```

```
    document.getElementById("demo").innerHTML = date.toLocaleString() //시간
```

```
    //document.getElementById("demo").innerHTML = date.toLocaleDateString(); //날짜
```

```
    //document.getElementById("demo").innerHTML = date.toLocaleTimeString(); //시간
```

```
</script>
```



# 내장 객체 - Date

## Date 객체의 날짜/시간 함수

```
> 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
```

# 내장 객체 - Date

## 날짜/시간을 알려주는 Date 객체

현재 년도 : 2022

현재 월 : 1

현재 일 : 19

현재 요일 : 3

===== 시간 =====

현재 시: 6

현재 분: 50

현재 초: 0

현재까지 시간 : 1642542600.136초

<h2>날짜/시간을 알려주는 Date 객체</h2>

<script>

```
const now = new Date();
```

```
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 시: " + now.getHours() + '<br>');
```

```
document.write("현재 분: " + now.getMinutes() + '<br>');
```

```
document.write("현재 초: " + now.getSeconds() + '<br>');
```

```
document.write("현재까지 시간 : " + now.getTime()/1000 + '초<br>');
```

</script>

# 날짜 계산 프로그램

## 날짜 계산 프로그램 만들기

지금까지 몇 일?

개강 이후 **22**일 지났습니다.

축하합니다.

# 날짜 계산 프로그램

## 날짜 계산 프로그램 만들기

```
<div id="container">
  <h2>지금까지 몇 일?</h2>
  <p>개강 이후 <span id="day" class="accent"></span>일 지났습니다.</p>
  <p>축하합니다.</p>
</div>

<script>
  let now = new Date(); //날짜 객체 생성
  let firstDay = new Date("2021-12-26");

  let passedTime = now.getTime() - firstDay.getTime(); //지난 시간 계산(밀리초)
  console.log(passedTime + "ms");

  passedTime = Math.round(passedTime/(24*60*60*1000));
  //밀리초(ms) -> 초(s)로 환산후 일(day)로 환산.. 24시간 60분 60초 1000밀리초

  document.querySelector("#day").innerText = passedTime;
  //document.getElementById("day").innerHTML = passedTime;
</script>
```

passedtime.html

# 날짜 계산 프로그램

## 날짜 계산 프로그램 만들기

```
#container{  
  width: 300px;  
  height: 300px;  
  margin: 50px auto;  
  border: 2px solid ■#222;  
  border-radius: 50%;  
  background-color: □aliceblue;  
  text-align: center;  
}  
h2{padding-top: 70px;}  
.accent{color: ■red; font-size: 1.5rem; font-weight: bold;}
```

# 내장 객체 – Array

## ● indexOf() 함수 예제

127.0.0.1:5500 내용:

입력해주세요

안녕

확인

취소

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 - Array

## ● indexOf() 함수 예제

```
while(true){
    let input = prompt("단어를 입력해 주세요");

    if(input.indexOf("안녕") >= 0){
        alert("안녕하세요");
    }
    else if(input.indexOf("시") >= 0 || input.indexOf("분") >= 0){
        const now = new Date();
        let hours = now.getHours();
        let minutes = now.getMinutes();
        alert("현재 시각은 " + hours + "시 " + minutes + "분 입니다.");
    }
    else if(input.indexOf("잘 가") >= 0 || input.indexOf("잘 있어") >= 0){
        alert("안녕히 가세요.");
        break;
    }
    else{
        alert("모르는 단어입니다.");
    }
}
```

indexOf2.html

# 내장 객체 - window

## window 객체 - setInterval() 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>
```



# 내장 객체 - window

## window 객체 - setTimeout() 함수

- setTimeout() : 초가 1번 설정되고 종료함

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

# 내장 객체 - window

## window 객체 - clearTimeout(시간) 함수

### - 시간 설정 해제 함수

```
<h1>Window 객체</h1>
<h2>The clearTimeout() Method</h2>
<button onclick="stop()">stop</button>
<h2 id="demo"></h2>
```

```
<script>
  const timeout = setTimeout(myGreeting, 3000);

  function myGreeting() {
    document.getElementById("demo").innerHTML = "Happy Birthday!!"
  }

  function stop(){
    clearTimeout(timeout);
  }
</script>
```

## Window 객체

### The clearTimeout() Method

stop

Happy Birthday!!

# 디지털 시계

## ● 시계 만들기

오후 10:59:31

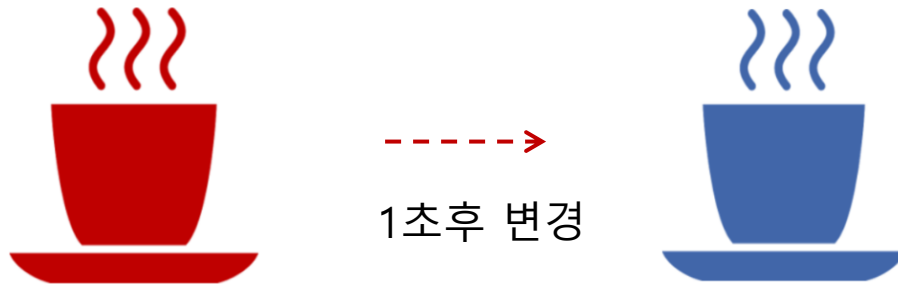
```
<h2>디지털 시계</h2>
<p id="demo"></p>
<script>
  // myWatch를 1초에 1번씩 호출
  const watch = setInterval(myWatch, 1000);

  function myWatch(){
    const date = new Date(); //날짜와 시간 생성
    var now = date.toLocaleTimeString(); //시간을 한글로 표시
    document.querySelector("#demo").innerHTML = now;
  }

  //실행 함수로 구현
  /*const watch = setInterval(function(){
    const date = new Date();
    let now = date.toLocaleTimeString();
    document.querySelector("#demo").innerHTML = now;
  }, 1000)*/
</script>
```

# 사진 바꾸기

- 클릭하면 1초 후에 이미지 변경



```
<div id="container">  
    
</div>
```

# 사진 바꾸기

- 클릭하면 1초 후에 이미지 변경

```
//실행 함수
function changePic(){
    let pic = document.getElementById("pic");

    setTimeout(function(){
        pic.src = "images/cup-2.png";
    }, 1000);
}
```

# 인사말 반복하기

## ● 인사말 반복하기

인사하기

생일 축하해!!



인사하기

오늘도 좋은 하루!!

```
<section>
  <h2>인사하기</h2>
  <h3 id="demo"></h3>
</section>
<script>
  let message = ["생일 축하해!!", "고맙습니다.", "오늘도 좋은 하루!!"];
  let m_idx = 0; //message 인덱스

  myGreeting(); //함수 호출

  function myGreeting() {
    document.getElementById("demo").innerHTML = message[m_idx];
    m_idx++; //1 증가시킴
    if(m_idx === message.length)
      m_idx = 0;
    setTimeout(myGreeting, 3000); //콜백 함수
  }
</script>
```

# 이미지 반복하기

- 이미지 반복하기

커피의 세계



커피의 세계



# 이미지 반복하기

- 이미지 반복하기

```
<section>
  <h2>커피의 세계</h2>
  
</section>
<script>
  let picture = ["images/coffee-blue.jpg", "images/coffee-gray.jpg",
                "images/coffee-pink.jpg"]
  let p_idx = 0;

  showPicture();

  function showPicture() {
    document.getElementById("pic").src = picture[p_idx];
    p_idx++;
    if(p_idx === picture.length)
      p_idx = 0;
    setTimeout(showPicture, 3000); //콜백 함수
  }
</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!!")  
  }  
}
```

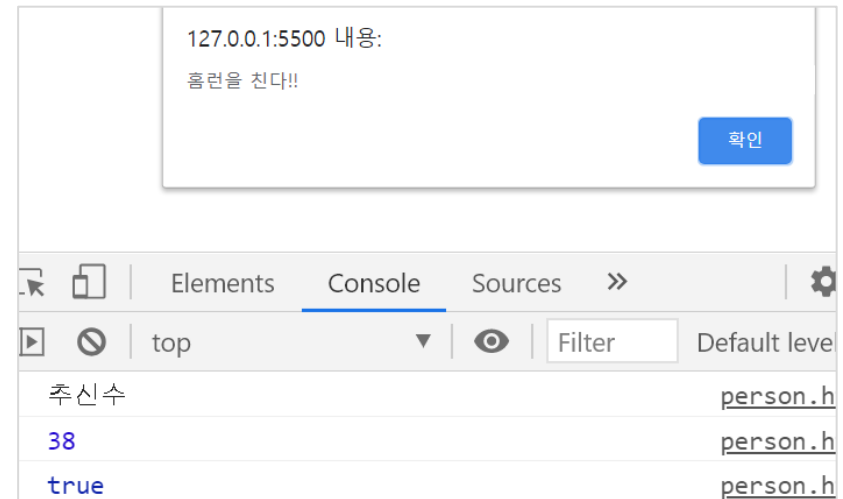
# 객체의 정의 및 사용

## person 객체 사용

### 객체 이름.속성(함수)

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

```
//book 객체(자료형) 생성
let book = {
  title : "웹 표준의 정석",
  author : "고경희",
  pages : 650,
  price : 30000,

  info : function(){
    alert(this.title + " 책의 분량은 " + this.pages + "페이지 입니다.");
  }
}

console.log("책 제목 : " + book.title);
console.log("저자 : " + book.author);
book.info();
```

# 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

작은 원의 둘레 : 31.42cm  
작은 원의 면적 : 78.54cm<sup>2</sup>  
큰 원의 둘레 : 62.83cm  
큰 원의 면적 : 314.16cm<sup>2</sup>

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

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

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

let c1 = new Circle(5);
let c2 = new Circle(10);

document.write("작은 원의 둘레 : " + c1.length().toFixed(2) + "cm<br>");
document.write("작은 원의 면적 : " + c1.area().toFixed(2) + "cm<sup>2</sup><br>");
document.write("큰 원의 둘레 : " + c2.length().toFixed(2) + "cm<br>");
document.write("큰 원의 면적 : " + c2.area().toFixed(2) + "cm<sup>2</sup><br>")
```



# 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