



함수



- 함수를 기본 코드를 활용할 줄 안다.
    - **function** 선언 형식
  - 사용자 정의함수, 내장 함수 구분 및 활용
  - 함수에서 입력값 처리방식, 프로세스(핵심), 리턴값에 대한 이해.
  - 사용자 편의성에 의해서 만들어진 내장 자바스크립트 함수들의 종류들과 활용방법을 안다.
-



생각해봅시다! :

---



# 자바스크립트 현재 어디즈음:

- 개요
  - 기본문법
  - 조건반복문
  - 객체
  - **함수**
  - 생성자 함수
  - 기본 내장 객체
  - 브라우저 객체 모델
  - 문서 객체 모델
  - 이벤트
-

# 함수란? :

- 코드의 집합이며, 정의된 코드 내용을 간단하게 호출하여 쓸 수 있는 것을 말한다.

길동아!나무를베어오렴!!

```
function 나무베기( 입력값){  
    입력값에 의해서 처리할 내용..(process)  
    나무베는 기술과 방법  
    return 결과값;  
}
```

정의된 함수 호출 : 나무베기('길동아');

- 함수의 종류 호출
  - 내장 정의 함수 : **prompt()**, **parseInt()**
  - 사용자 정의 함수 : 호출 => **call()**;
    - **var call = function ()** { 정의된 내용 }
    - **function call()** {정의된 내용}



## 사용자 정의 함수 예제 :

- function 함수이름(입력값1, 입력값2){}
  - function plus(num01, num02){ : 입력값 정의
  - 처리할 프로세스 정의
  - var sum = num01 + num02;
  - return sum; : 결과물을 return
  - }
  - 호출: 함수명(입력할 데이터값1, 입력할 데이터값2);
    - plus( 25, 21); ➔ 46 결과값이 함수를 통해서 처리



## 확인 예제 :

- minus 함수를 정의하고, 이 함수에 입력값 num01, num02를 처리했을 때, 결과값을 console.log, alert에서 처리하게 하세요..
- multi 함수
  - 입력값 num1, num02
  - 출력값 "@@ \* @@ = @@@@@" 문자열 처리된 내용
  - console.log에서 출력 처리하세요..




## 확인예제 풀이 :

- `function multi(num01, num02){`
    - `var sum = num01*num02;`
    - 현재 데이터값은 num01, num02, sum
    - `var strRst=num01+" * "+num02+" = "+sum;`
    - `return strRst;// 문자열을 return...`
  - `}`
  - `multi(3,2);`
    - num01:3, num02:2, sum :6
    - strRst : "3 \* 2 = 6"; return
  - `var result =multi(3,2);`
  - `alert(result);`
-



- 코드의 집합을 나타내는 자료형  
**var 함수명 = function(){ 코드들 };**  
**function 함수명(){**  
    코드들.  
**}**
- 코드의 집합
  - 괄호 내부에 처리할 코드를 의미.

- 함수의 변수를 지정..
  - 변수 : `var 함수명;`
  - 함수를 할당 : `함수명 = function (param1){  
처리할 프로세스 };`
  - 함수호출 : `함수명(25);`
- `var show01 = function(showStr){`
- `alert("입력받은 출력내용"+showStr);`  
`};`
  - 함수변수=함수내용할당;
  - `show01("안녕하세요!!!");`

- 
- 함수 변수 선언..
  - 함수명 : introMyfamily
  - 입력값 : function  
(mem01,mem02,mem03.....)
  - 출력내용..
    - 우리가족 구성원은...
    - @@@, @@@, @@@, @@@ 입니다.
-



## 여러 가지 내장 함수 :

- 출력..
    - `alert("기본 경고 창 출력");`
    - `confirm("확인창");`
      - 입력값으로 "화면에 출력된 내용"
      - 프로세스 확인 → `return true;`
      - 프로세스 취소 → `return false;`
      - `return` 값을 `true/false`의 `boolean`
      - 일반적으로 프로세스 중단여부, 다음 `page`전환여부를 처리할 때 활용
    - `prompt("lebel내용", "입력할 기본데이터");`
-



## 여러 가지 내장 함수 :

- 출력..
    - `prompt("입력값1", "입력값2");`
      - 입력값1: 화면에 **title**내용 출력..
      - 입력값2: **return**할 초기 **default**값
      - 프로세스 : 확인 ➔ 입력문자열 **return**
      - 취소 ➔ **null return;**
      - **return** : 입력한 내용을 문자열로 **return**
-



## 여러 가지 내장 함수 :

### ■ 기본예제

- `var isMale=confirm("남자입니까?");`
    - 확인을 눌렀을 때는 `true`값을 `isMale`에 할당
    - 취소를 눌렀을 때는 `false`값을 `isMale`에 할당
  - `if( isMale ){`
    - `isMale`이 `true`일 때, 처리할 내용...
    - `alert("gentleman 이리로 오십시오!!");`
  - `}else{`
    - `isMale`이 `false`일 때, 처리할 내용...
    - `alert("pretty lady 이쪽입니다");`
  - `}`
-



# Math.random() :

- 범위( $R = \text{Math.random}()$ )
  - 실수로...  $0.0 \leq R < 1.0$
  - 0부터 9까지의 정수를 임의로 발생하게 하자..
    - $0.0 * 10 \leq R * 10 < 1.0 * 10$
    - $0.0 \leq R * 10 < 10.0$
    - 실수를 정수로 변환처리하는 함수.. `parseInt()`
    - $0 \leq \text{parseInt}(R * 10) < 10$
    - 0,1,2,3,4,5,7,8,9까지의 임의 수가 발생..
  - 1부터 6까지의 주사위 숫자가 임의로 발생..
    - $R * N : N$ 은 가지수를 처리한다. 한계치..
    - $1 \leq \text{parseInt}(R * 6) + 1 < 7$  (1,2,3,4,5,6)



## 여러 가지 내장 함수 :

- 확인예제
    - confirm을 활용해서 지금 오전입니까?
      - 확인일 때는 "Good morning!! 좋은 하루시작하세요!"
      - 취소일 때는 "Good job!!! 활기찬 오후 되시길!!"
    - 숫자값을 임의로 두 개 입력 받아서..
      - 합산된 값이 10이상일 때, Win!!!
      - 합산된 값이 10이하일 때, Lost!!!
-





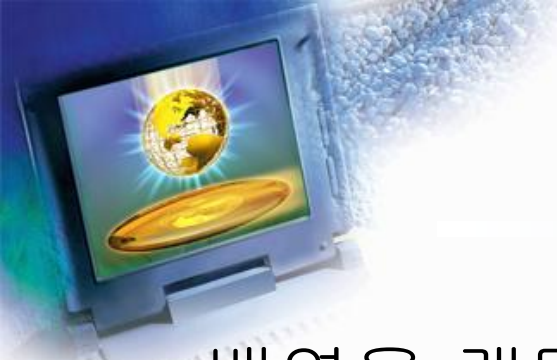
## confirm 예제 :

- 임의 두개의 값 Math.Random() 처리하게 한다.
    - var ran01=parseInt(Math.random()\*6)+1;
    - var ran02=parseInt(Math.random()\*6)+1;
  - 합산 값을 변수에 할당한다.
    - var sum = ran01+ran02;
  - confirm 합산값 @@이 10이상?
  - 확인 : Win if(sum>=10) ...
  - 취소 : Lost else ...
-



## 가위 바위 보 :

- 한 사람이 가위 바위 보를 임의로 내고,
  - 다음 사람이 가위 바위 보를 임의로 냈을 때..
  - 승리는 prompt처리..
- 팀명 입력하기..
  - `var teamName=prompt("팀명을 입력하세요!","홍팀");`
  - `prompt`창으로 문자열을 입력받는데, 입력하지 않고, 확인을 누르면.. `teamName` 이 홍팀으로 입력됨..
- 가위바위보 임의 처리..
  - `var games=["가위","바위","보"]; index(0,1,2)`
  - `games[ 0 ], games[2]`
  - `random`범위가 0~2 `games[ 랜덤범위..];`
  - `games[ parseInt( Math.random()*3 ) ] → 0~2`



## 가위바위보게임 승패처리..

- 배열을 랜덤으로 가위, 바위, 보 나타내기..
  - `var games=["가위","바위","보"];` 0,1,2
  - `var ranIdx1 = parseInt(Math.random()*3);`
  - `var ranIdx2 = parseInt(Math.random()*3);`
  - A 팀: `games[ranIdx1]`, B팀: `games[ranIdx2]`
- 승부.. 가위 > 보, 바위 > 가위, 보 > 바위
- (0 2) (1 0) (2 1)
- index번호를 기준으로 +2가 같은 index 승

%3	가위	바위	보
	0	1	2



```
var teamName1=prompt("팀이름을 정하세요","홍팀");  
//alert("입력한 팀이름:"+teamName);  
var games=["가위","바위","보"];  
var ranIdx1 = parseInt(Math.random()*3);  
alert(teamName1+":"+games[ranIdx1] );  
var teamName2=prompt("팀이름을 정하세요","청팀");  
var ranIdx2 = parseInt(Math.random()*3);  
alert(teamName2+":"+games[ranIdx2] );  
if(ranIdx1==ranIdx2){  
alert("무승부");  
// 첫번째 팀의 index번호에서 2를 더한 값과 두번째 팀의 index  
// 비교해서 같으면 첫번째 팀이 승....  
}else if( ( ranIdx1+2 )%3 == ranIdx2 ){  
alert(teamName1+"승");  
}else{  
alert(teamName2+"승");  
}
```



## 시간관련 내장 함수 :

- `setTimeout(함수, 1/1000초);`
    - 지정된 함수를 시간 이후(1/1000)에 처리..
    - ex) 3초 후에 함수를 실행하기..
    - `setTimeout( function(){ alert("3초 지남!!");}, 3000);`
  - `setInterval(함수, 1/1000초);`
    - 지정된 함수를 시간(1/1000)초마다 반복처리..
  - `clearInterval(반복함수 변수);`
    - 반복정의 된 함수를 중지시켜줌.
    - ex) `var inter01 = setInterval();`
    - `clearInterval( inter01); // 반복함수 중지.`
-



## setTimeout확인예제 :

- 초기화면에 시험 시작!!!! - 화면 출력.
- 10초 후!!
- 시험 종료!!! 화면에 출력
- setTimeout,
- document.getElementsByTagName("tag")  
;
- 풀이
  - a07\_innerTimmingExp.html



## setInterval 기본예제 :

- 1초부터 초당 카운트 업 하는 예제..
  - 1초, 2초, 3초.....
- 전역변수 선언 : `var cnt=1;`
- `setInterval(function(){`
  - `document.getElementsByTagName("h1")[0].innerHTML=cnt+"초";`
  - `cnt++;`
- `}, 1000);` : 1초마다반복



## 나머지 연산자와 **count**처리.:

- **img**가

- **img01.png**, **img02.png**, **img03.png**
- 반복적으로 이미지를 나타낼려면 **1,2,3,1,2,3.....**
- 배열

- **var**

- imgs=["fish.png","global.gif","school.png"];**

- **imgs[index] : 0,1,2,0,1,2**

- **count**가 증가하더라도 반복적인 **index**나 번호가 나타날 수 있게 처리..- 나머지(**%**)**imgs[count%3]**

<b>count</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
<b>count%3</b>	<b>1</b>	<b>2</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>0</b>	<b>1</b>





## 정리 및 과제 :

- 함수값에 입력값을 전달하고 호출하는 형식을 알고 있는가?
- **javascript**의 **prompt** 입력값 **2**개와 **return**값이 무엇인지 알고 있는가?
- 임의의 수 **15~20**까지 출력하려고 한다. 이를 위한 어떤 함수를 어떻게 호출해야 할지 알고 있고, 실제 코드를 쓸 수 있는가?
- 시간 처리함수의 **3**가지 형태를 알고, 각각 입력값이 어떤 처리를 하는지 알고 있고 쓸 수 있는가?



**quiz:**

---



**다음 장은 입니다 !**

---