

자바스크립트 (Javascript)

✓ 원리를 알면 IT가 맛있다

JavaScript for Beginners

chapter 05.

함수 (function) 기본

- 함수의 역할 3가지.
- 함수를 정의하는 방법 3가지
- 자바스크립트 함수 모델 개요
- arguments 내장변수 사용
- 중첩 함수
- 콜백 함수
- 유틸리티 함수

○ 함수의 역할 3가지.

가. 일반적인 호출 가능한 형식으로서의 함수
: 가장 일반적으로 인식하는 함수역할

나. 값으로서의 함수 (함수객체)
: 변수에 할당 가능
: 다른 함수의 인자로 전달 가능
: 다른 함수의 반환값으로 사용 가능.

다. 다른 객체(인스턴스)를 생성할 수 있는 함수
: '생성자 함수' 라고 한다.
: new 이용.

○ 함수를 정의하는 방법 3가지

가. 선언적 정의 방법 (이름있는 함수)

나. 함수 리터럴 (익명함수)

다. Function 생성자 이용

□ 2) 함수 (function) 정의 방법 3가지

가. 선언적 정의 방법 (이름있는 함수)

문법:

```
function 함수명( [매개변수1,매개변수2,...] ){  
  
    문장;  
  
    [return 리턴값;]  
}
```

- 예 1 : 매개 변수가 없는 경우

```
function func1() {  
  
    alert("JavaScript");  
  
}
```

- 예 2 : 매개 변수가 있는 경우

```
function func2(name) {  
  
    document.write(name + "씨 환영한다.<br>");  
  
}
```

□ 2) 함수 (function) 정의 방법 3가지

- 예 3 : 복귀 값이 있는 경우

```
function func3(value) {  
  
    var ans;  
  
    ans = value + 200;  
  
    return ans;  
  
}
```

- 함수가 생성되는 시점은 프로그램의 파싱(parsing,컴파일)단계에서 생성된다.
따라서 함수 정의 전에 호출이 가능하다.

```
예> func4();  
function func4(){  
    ...  
}
```

나. 함수 리터럴 (익명 함수)

문법:

```
var 변수명 = function ( [매개변수1,매개변수2,..] ){  
  
    문장;  
    [return 리턴값;]  
}
```

- 변수명이 함수명이 된다.
- 함수가 생성되는 시점은 프로그램의 실행단계에서 생성된다.
따라서 함수 정의 전에 호출이 불가능하다.

```
예> func4(); // 에러  
var func4 = function (){  
    ...  
}
```


다. Function 생성자 이용
문법:

```
var 변수명 = new Function( [인자1,인자2,...] 함수본체코드);
```

- 여러 개의 인자설정이 가능하다. 마지막 인자위치에는 함수에서 실행될 실행코드를 명시한다.
- 함수가 생성되는 시점은 프로그램의 실행단계에서 생성된다.
따라서 함수 정의 전에 호출이 불가능하다.

```
예> var result = func5(10,20); // 에러  
var func5 = new Function ( "x ", "y " , "return x + y" );
```

○ 자바스크립트 함수 모델

- 함수 생성시 자바스크립트가 함수를 관리하는 메커니즘이다.
- 다음과 같이 복잡한 이유는 함수의 3가지 기능을 모두 만족시켜야 되기 때문이다.

예> `function add(x , y){
 return x+y;
}`

변수 스코프 객체

+prototype
- x
- y

실행코드 영역

return x+y;

‘함수명()’ 으로 접근

‘함수명.prototype’ 으로 접근

prototype 객체

+constructor

공개변수 영역

‘함수명.멤버’ 으로 접근

○ 매개변수와 인자

- 함수에서 정의된 매개변수(parameter)와 호출하는 인자(argument) 갯수가 달라도 가능하다.
- 자바스크립트는 메소드 호출시에 파라미터의 갯수를 체크하지 않는다.

```
function printWebTech(value) {  
    document.writeln(value);  
}  
  
printWebTech ();    // undefined  
  
printWebTech ('CSS3 ');    // CSS3  
  
printWebTech ('CSS3', 'JSP');    // CSS3
```

- 이때 전달되는 파라미터의 개수를 모두 출력하고자 하는 경우에는 arguments 라는 함수내에서 자동으로 생성되는 내장변수를 사용할 수 있다. 내부적으로 arguments 내장변수는 넘어오는 파라미터를 배열로 관리한다.
- 파라미터 개수는 arguments.length 형식으로 얻는다.
- 요소접근은 arguments[index] 형식으로 접근 가능하다.

예>

```
function one( x , y , z){  
    for( var i = 0 ; i < arguments.length ; i++){  
        console.log( arguments[i]);  
    }  
}
```

```
one(10,20);  
one(10,20,30);  
one(10,20,30,40);
```

- `arguments.callee` 속성은 현재 실행되고 있는 함수를 나타낸다.
- 재귀 호출시 유용하게 사용 가능하다.

예>

```
function makeFactorialFunc(){  
    return function(x){  
        if( x <= 1) return 1;  
        return x * arguments.callee( x - 1 );  
    }  
}  
  
var result = makeFactorialFunc()(5);  
console.log( result ); // 120 ( 5 * 4 * 3 * 2 * 1 )
```

○ 고급 함수 형태 2가지

가. 중첩 함수

특정 함수에서만 사용하는 기능이라면 외부에 노출시키지 않고 내부에 정의해서 사용 가능하다.

a. 익명함수 형태

// b 함수는 외부에서 접근이 불가

```
var a = function(arg){  
    var b = function(n){  
        return n*100;  
    }  
    return b(arg);  
};
```

```
//외부에서 접근 방법  
var result = a(2);  
console.log(result);
```

b. 선언적 함수 형태

// y 함수는 외부에서 접근 불가

```
function x(arg){  
    function y(n){  
        return n*10;  
    }  
    return y(arg);  
}
```

```
//외부에서 접근 방법  
var result = x(3);  
console.log(result);
```

- 중첩함수 특징은 내부함수에서는 내부에 정의된 변수를 접근할 수 있다는 것이다.
- 외부에서는 접근할 수 없는 내부의 변수에 내부함수가 접근할 수 있다는 것을 이용하면 객체지향의 '은닉화(encapsulation)' 기법을 적용할 수 있다.

```
//은닉화
var k = function(){
  var d = 10;
  var k2 = function(){
    return d * 2;
  }
  return k2();
}

console.log("k() 호출 : " + k() ); //20
```

- 16 -

가. alert 함수

경고창을 화면에 보여줄 때 사용한다.

예> alert(값);

나. setTimeout 함수

일정시간에 함수를 단 한번 실행한다.

예> setTimeout(function(){
 console.log("3초후에 출력");
},3000);

다. setInterval 함수

일정시간마다 반복해서 함수를 실행한다.

예> var interval = setInterval(function(){
 console.log("3초후에 출력");
},3000);
clearInterval(interval);

라. 문자열을 숫자로 , 실수로

예> var num = parseInt("123"); var f = parseFloat("3.14");

- 함수 정의 방법 3가지 및 특징
- 함수 모델 개요
- arguments 내장변수
- 중첩함수와 콜백함수
- 유틸리티 함수



Thank you
