

9강_함수-심화

- 9-1 함수 반환(리턴)함수
- 9-2 클로저
- 9-3 내부함수
- 9-4 콜백함수
- 9-5 내장함수

9-1 : 함수 반환(리턴)함수

Ex : 9_01.html

함수는 또 다른 함수를 반환할 수 있다.

```
function funName() {  
    실행문  
  
    return function(){  
        실행문  
    };  
}
```

→ 함수 반환

```
/*  
 * 함수 반환  
 */  
  
function funName() {  
  
    return function(x) {  
        for(var i=1; i<10; i++){  
            console.log(x + " * " + i + " = " + (x*i));  
        }  
    };  
}  
  
var returnFun = funName();  
returnFun(4);
```

9-2 : 클로저

Ex : 9_02.html

반환된 함수의 life-cycle가 종료될 때까지 호출된 함수의 life-cycle이 유지되는 것

scop

free var.

```
function funName(x) {  
  var varName = ":: 요청한 " + x + "에 대한 구구단::";  
  return function() {  
    console.log(varName);  
    for(var i=1; i<10; i++){  
      console.log(x + " * " + i + " = " + (x*i));  
    }  
  };  
}  
  
var returnFun = funName(5);  
returnFun();
```



```
Elements Console Sources  
top Filter  
:: 요청한 5에 대한 구구단::  
5 * 1 = 5  
5 * 2 = 10  
5 * 3 = 15  
5 * 4 = 20  
5 * 5 = 25  
5 * 6 = 30  
5 * 7 = 35  
5 * 8 = 40  
5 * 9 = 45  
>
```

9-3 : 내부함수

Ex : 9_03.html

함수안에 또다른 함수

```
function outFun () {
```

```
  innerFun1() {...}  
  innerFun2() {...}  
  ...  
  innerFunn() {...}
```

```
}
```

내부 함수로 외부에서는 호출 불가

```
/*  
  내부함수  
  : 함수안에 또다른 함수  
*/  
  
function outFun(width, height){  
  
  console.log("triangle : " + triangle());  
  console.log("quadrangle : " + quadrangle());  
  
  //삼각형 면적 계산  
  function triangle() {  
  
    return (width * height) / 2;  
  
  }  
  
  //사각형 면적 계산  
  function quadrangle() {  
  
    return width * height;  
  
  }  
  
}  
  
outFun(3, 5);
```



Elements	
top	
triangle : 7.5	
quadrangle : 15	
>	

9-4 : 콜백함수

Ex : 9_04.html

매개변수로 함수를 전달하고, 전달된 매개변수가 특정 시점에 호출되는 것

```
function funName (fun) {
```

실행문;

```
fun();
```

매개변수로 전달된 함수 실행

```

/*
   콜백함수
   : 매개변수로 함수를 전달
*/

function funName (funCBF, funCBP, num){

    for(var i=1; i<=10; i++) {
        console.log(num + " * " + i + " = " + (num * i));
        if(i < 10) funCBP(i);
    }

    funCBF();
}

function funCallBackProgress (n){

    console.log((n * 10) + "% 진행 완료!");
};

function funCallBackFinish(){

    console.log("서버 작업 종료!");
};

funName (funCallBackFinish, funCallBackProgress, 7);

```



Elements	Cons
top	
	7 * 1 = 7
	10% 진행 완료!
	7 * 2 = 14
	20% 진행 완료!
	7 * 3 = 21
	30% 진행 완료!
	7 * 4 = 28
	40% 진행 완료!
	7 * 5 = 35
	50% 진행 완료!
	7 * 6 = 42
	60% 진행 완료!
	7 * 7 = 49
	70% 진행 완료!
	7 * 8 = 56
	80% 진행 완료!
	7 * 9 = 63
	90% 진행 완료!
	7 * 10 = 70
	서버 작업 종료!

9-5 : 내장함수

Ex : 9_05.html

개발자가 직접 만들지 않아도 javascript가 기본적으로 제공하는 함수

```
alert();
```

```
setTimeout();
```

```
clearTimeout();
```

```
console.log();
```

```
setInterval();
```

```
eval();
```

```
clearInterval();
```