

# JavaScript & Jquery 08

---

by withsky

# 1 함수

## 함수란?

특정 기능을 하는 변수와 구문의 집합  
중복되는 부분을 묶어 함수로 포장 (재사용 용도)

## 함수의 구조

```
function 함수명 (파라미터명1, 파라미터명2....) {  
    실행코드  
    return 반환값;  
}
```

파라미터가 없을 수도 있고, return(반환값 / 결과값)도 없을 수도 있음

# 2 함수

## 함수 선언방법

1. 가장 일반적인 방식

```
function 함수명 (파라미터...) {  
    return 반환값;  
}
```

2. 이름없는 함수

```
var 함수명 = function(파라미터...) {  
    return 반환값;  
}
```

3. 즉시 실행 함수

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

```
<script>
```

```
// 즉시 실행
```

```
(function abc() {  
    aaa();  
    //bbb();    // 에러  
})();
```

```
// 일반적인 선언
```

```
function aaa() {  
    console.log("aaa");  
}
```

```
// 이름없는 함수
```

```
var bbb = function () {  
    console.log("bbb");  
}
```

```
bbb();
```

```
</script>
```

# 3 파라미터

## 파라미터 (Parameter, 매개변수)

외부데이터를 함수내부로  
전달해주기 위한 용도

함수 내부의 지역 변수  
(함수 내부에서만 사용 가능)

```
// 문자열
function aaa(name) {
    console.log("안녕하세요 저는 "+name+" 입니다.");
}

aaa("서민구");

// 숫자
function bbb (no) {
    var no2 = no * no;
    console.log(no + "의 제곱은 "+no2+" 입니다.");
}

bbb(9);

// 함수
function ccc() {
    console.log("ccc함수 호출");
}

function ddd(func) {
    func();
}

ddd(ccc);
```

# 4 아규먼트

## 아규먼트 (Arguments, 가변인자)

파라미터의 정보가 담긴 객체  
배열형태로 저장  
모든 함수에 무조건 존재

```
function aaa() {  
    console.log(typeof(arguments));  
    console.log(arguments);  
  
    console.log(arguments[2]);  
}  
  
aaa(1,2,3,4,5);
```

\* 함수 선언 시 파라미터를 명시하지 않아도 arguments를 이용해 코딩 가능하지만  
동일하게 맞춰서 코딩 할것.

# 5 return

## return

- 결과값을 호출한 곳으로 넘기는 역할
- 함수를 호출한 곳으로 돌아가라는 의미

```
<script>

function aaa() {
    console.log(1);

    return;

    console.log(2);
}

// 함수 호출
aaa();

</script>
```

# 6 내부 함수

## 함수 안에 존재하는 함수

소규모 웹사이트인 경우는 크게 문제되지 않지만,  
대규모 웹사이트인 경우 여러 개발자가 작업 시,  
동일한 이름을 가진 함수가 존재할 가능성 있음

동일한 이름의 함수는 나중에 선언된 함수로 정의

내부 함수를 이용하면 이름 충돌 방지  
jquery 플러그인 대부분은 내부함수로 작성

```
<script>

    function run() {
        console.log(" 출근하기 위해 뛰다");
    }

    function run() {
        console.log(" 학교가기 위해 뛰다");
    }


    function company() {
        run();
    }

    function schoole() {
        run();
    }

    company();
    run();

</script>
```

내부 함수로 변경 실습

# 7 내장 함수

## JavaScript 자체적으로 제공하는 함수

별도로 정의하지 않았지만 기본적으로 내장된 함수

예) alert(), parseInt(), toString()...

예) 인코딩 / 디코딩 내장 함수

escape() / unescape()

encodeURIComponent() / decodeURIComponent()

encodeURIComponent() / decodeURIComponent()

escape() : 영문, 숫자, 특수문자 일부 제외 인코딩

encodeURIComponent() : 영문, 숫자, 특수문자 일부(인터넷 주소에 사용되는 특수문자 :, / = ? &) 제외

encodeURIComponent() : 영문, 숫자 제외 모든 문자 인코딩 ( == **UTF-8 인코딩** )

## 왜 인코딩을 해야 하나?

- url에는 한글, 특수문자, 공백이 포함될 수 없음
- 서버나 타 시스템과의 데이터 송/수신