

## Ch05 함수

### 5.1 익명 함수

함수는 코드의 집합을 나타내는 자료형. 익명 함수는 다음 형태로 만든다.

```
var functionVar = function() { }; // 함수 표현식
```

```
var functionVar = () => {}; // 화살표 함수
```

```
// 매개변수 지정 방법
```

```
() => { ... } // 매개변수가 없을 경우
```

```
x => { ... } // 매개변수가 한 개인 경우, 소괄호를 생략할 수 있다
```

```
(x, y) => { ... } // 매개변수가 여러 개인 경우, 소괄호를 생략할 수 없다
```

```
// 함수 몸체 지정 방법
```

```
x => { return x*x } // single line block
```

```
x => x * x // 함수 몸체가 한줄의 구문이라면 중괄호를 생략
```

```
() => { return { a: 1 }; }
```

```
() => ({ a: 1 }) // 위 표현과 동일하다. 객체 반환시 소괄호를 사용한다.
```

```
() => { // multi line block.
```

```
    const x = 10;
```

```
    return x * x;
```

```
};
```

※ 익명 함수 생성과 출력 ; 이름이 없으므로 변수에 넣어 사용해야 한다.

### 5.2 선언적 함수

※ `function 함수명 () { // 함수 선언식`

```
}
```

※ 익명함수와 선언적 함수의 차이

```
<script>
```

```
    fun();
```

```
    function fun() { alert('함수 A'); } //웹브라우저는 script 태그 내부의 내용을 읽기전에
```

```
선언적 함수부터 읽는다
```

```
    function fun() { alert('함수 B'); } // 함수를 재정의 하면 기존의 함수는 없어지고, 재정의한
```

```
함수를 따른다
```

```
</script>
```

### 5.3 매개변수와 리턴값

```
function 함수이름(매개변수, 매개변수, 매개변수) {  
    //함수 코드  
    return 리턴값;  
}
```

※ 리턴값이 없는 경우

```
<script>  
    var fun = function (i) {  
        alert('리턴값이 없는 함수 '+i);  
    };  
    var result = fun(20);  
    alert('함수결과 : '+result);  
</script>
```

### 5.4 매개변수

자바 스크립트는 함수를 생성할 때 지정한 매개변수보다 많거나 적은 매개변수를 사용하는 것을 허용함. 원래 함수에서 선언된 매개변수보다 많이 사용하면 일반적으로 추가된 매개변수는 무시하고, 원래 함수에서 선언한 매개변수보다 적게 사용하면 지정하지 않는 매개변수는 undefined로 입력된다.

※ Array()함수 ; 빈 배열을 만든다.

Array(number) ; 매개변수만큼의 크기를 가지는 배열을 만든다.

Array(any, ..., any); 매개변수를 배열로 만든다.

### 5.5 가변 인자 함수 ;

매개변수를 선언된 형태와 다르게 사용했을 때, 매개변수를 모두 활용할 수 있는 함수. 매개변수의 개수가 변할 수 있는 함수 ex. Array()함수

☞ arguments : 매개변수 배열

※ 자기호출함수

(function(){ ... })(); //다른 개발자에게 영향을 주지 않게 함수를 생성하자마자 호출하는 경우

## 5.6 내부 함수 ; 함수 내부에 선언하는 함수

```
<script>↵
  /* A씨가 만든 함수 */↵
  function square(x) { return x * x; }↵
  function pythagoras(width, height) {↵
    return Math.sqrt(square(width) + square(height));↵
  }↵
</script>↵
<script>↵
  /* B씨가 만든 함수 */↵
  function square(width, height, hypotenuse) {↵
    if (width * width + height * height == hypotenuse * hypotenuse) {↵
      return true;↵
    } else {↵
      return false;↵
    }↵
  }↵
</script>↵
```

※ 위의 충돌을 막기 위해 내부함수를 사용하면 내부함수가 우선시 사용 됨.

## 타이머 함수

- ※ `setTimeout(function, millisecond)` ; 일정시간 후 함수를 한번 실행
- ※ `setInterval(function, millisecond)` ; 일정시간마다 함수를 반복해서 실행
- ※ `clearTimeout(id)` ; `setTimeout` 중지
- ※ `clearInterval(id)` ; `setInterval` 중지