

# 함수





함수



# 함수

## 함수

- 소스코드의 집합으로 메소드, 모듈, 기능, 프로시저 등을 말함
- 자바스크립트에서는 함수 또한 하나의 자료형
- 함수는 인자, 매개변수, 리턴 값을 가질 수 있음



# 함수

## 함수선언

- 반환 값 선언 없이 function 키워드만 이용하여 사용
- function 키워드에 함수 명을 작성하여 사용하는 방법(선언적 함수)
- function 키워드에 함수 명을 작성하지 않고 변수에 대입하는 방법(익명 함수)
- 스스로 동작하는 함수(익명 함수)



# 함수

## 선언적 함수

```
function 함수명([매개변수]){  
    처리로직;  
    [return 되돌려줄 값;]  
}
```

함수가 필요한 곳에서 함수명([매개변수]);를 통해서 함수 호출



# 함수

## 변수에 함수를 넣어 사용

```
const 변수명 = function ([매개변수]){  
    처리로직;  
    [return 되돌려줄 값;]  
}
```

함수가 필요한 곳에서 변수명([매개변수]);를 통해서 함수 호출

## 스스로 동작하는 함수(호출 없이 바로 실행)

```
(function (){  
    처리로직;  
})();
```



# 함수

## 함수 호출

- 선언적 함수는 만든 함수를 메모리에 바로 올려 놓기 때문에 코드의 순서와 상관없이 호출 가능
- 변수에 담기는 익명함수는 함수 선언 이후에만 사용 가능
- `return`이 있는 함수는 결과를 변수에 담아서 사용
- 함수 호출 시 매개변수를 전달하며 호출
- 지정한 매개변수보다 매개변수가 많은 경우 초과되는 매개변수는 무시
- 지정한 매개변수보다 매개변수가 적은 경우 선언되지 않은 매개변수는 `undefined`로 자동설정

※ 자바스크립트에서는 함수도 하나의 자료형으로 매개변수로 함수도 전달 가능



# 함수

## 가변인자 함수

- 매개변수의 개수가 변하는 함수
- 모든 함수의 내부에 arguments라는 배열이 자동으로 생성되어 매개변수를 저장
  - 매개변수가 지정되지 않은 값이 넘어오면 arguments 배열에 순서대로 저장
- 매개변수의 개수에 따라 함수처리를 달리 하려면 조건문을 사용하여 다르게 적용 가능





# 함수

## 함수 리턴(클로저)

- 함수 내부에서 사용했던 지역변수를 외부에서 사용하기 위해 사용하며 클로저라고 불림
- 클로저
  - 지역변수만 남기는 현상
  - 리턴 함수로 인해 생겨난 공간
  - 리턴 되는 함수 자체
  - 남겨진 지역변수



# 함수

## 내장함수(인코딩, 디코딩)

- 웹상에서 통신 시 유니코드문자는 오작동을 일으킬 수 있어 인코딩이 필요

메소드	설명	비고
encodeURIComponent()	최소한의 문자만 인코딩	인터넷 주소에 사용되는(;;,/,=,?,&) 변환 제외
decodeURIComponent()	encodeURIComponent()로 인코딩 된 값을 디코딩	
encodeURIComponent()	대부분의 문자 인코딩	알파벳과 숫자를 제외한 모든 문자 인코딩
decodeURIComponent()	encodeURIComponent()로 인코딩 된 값을 디코딩	



# 함수

## 내장함수

메소드	설명
eval(String)	String을 자바스크립트 코드로 실행
isFinite(number)	매개변수가 숫자인지 확인(숫자면 true)
isNaN(number)	number가 NaN인지 확인(숫자가 아닌지 확인)

※ NaN : Not a Number라는 뜻으로 숫자가 아님을 의미

