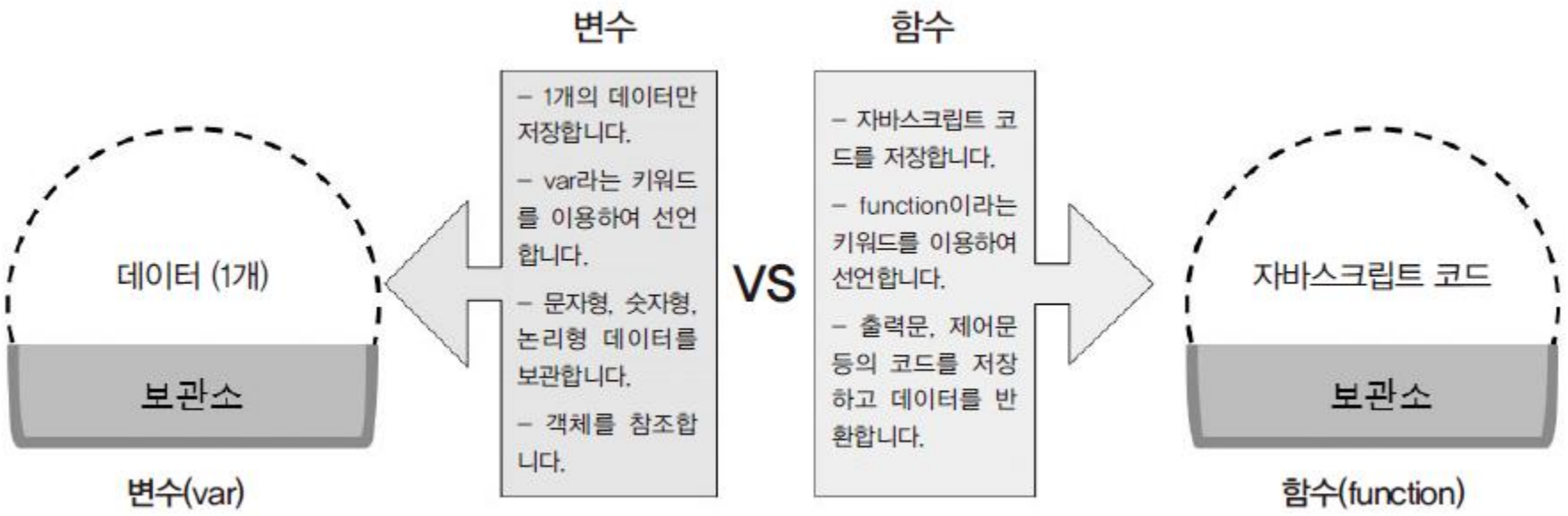
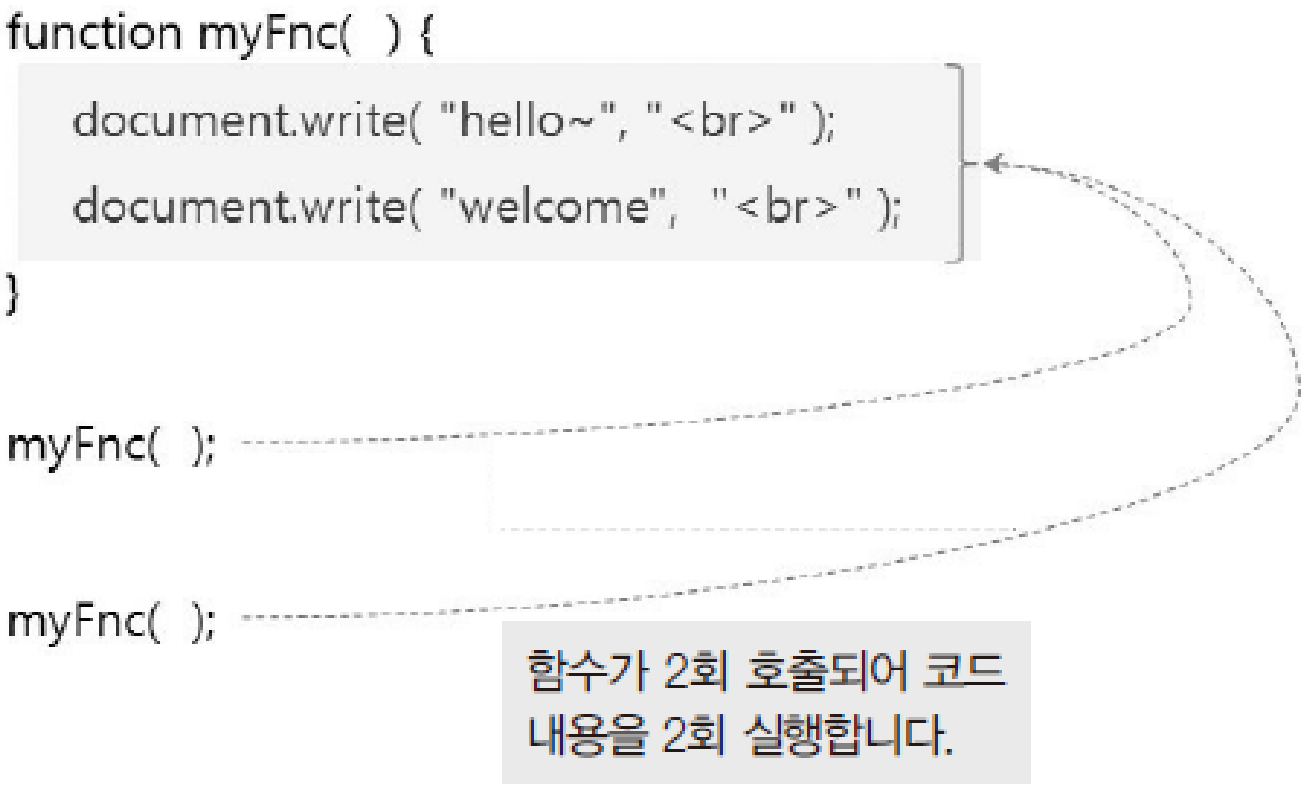
05-1 함수

함수란?


변수
함수
데이터 (1개)
보관소
변수(var)
– 1개의 데이터만 저장합니다.
– var라는 키워드를 이용하여 선언합니다.
– 문자형, 숫자형, 논리형 데이터를 보관합니다.
– 객체를 참조합니다.
VS
– 자바스크립트 코드를 저장합니다.
– function이라는 키워드를 이용하여 선언합니다.
– 출력문, 제어문 등의 코드를 저장하고 데이터를 반환합니다.
자바스크립트 코드
보관소
함수(function)

기본 함수 정의문 — 정의, 사용



일반 함수 정의 vs 익명 함수 선언 참조

전문가의 조언

일반 함수 정의 방식과 익명 함수 선언 참조 방식의 차이점

일반 함수 정의는 함수 호출 시 호이스팅(hoisting) 기술을 지원합니다. 그러나 익명 함수 선언 참조 방식은 호이스팅을 지원하지 않습니다. 호이스팅을 적용하면 함수 정의문보다 호출문이 먼저 나와도 함수 정의문을 끌어올려 함수를 호출합니다.

A. 일반 함수 정의 방식 (정상 작동)	B. 익명 함수 선언 참조 방식 (오류 발생)

ⓐ testFnc( ); ← 호출문
 	
호이스팅(hoisting)
 	
ⓑ function testFnc( ) {
ⓒ 자바스크립트 코드;
} ← 함수 정의문
	ⓓ testFnc( );
	
	ⓔ var testFnc = function ( ) {
	ⓕ 자바스크립트 코드;
	}

매개변수가 있는 함수 정의문

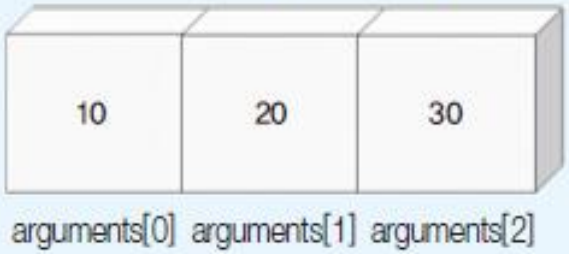
기본형

```
function 함수명{(매개변수 1, 매개변수 2,…매개변수 n)
   자바스크립트 코드;
}
함수명(데이터 1, 데이터 2,…데이터  n);
```

① ② ③ ⓐ ⓑ ⓒ

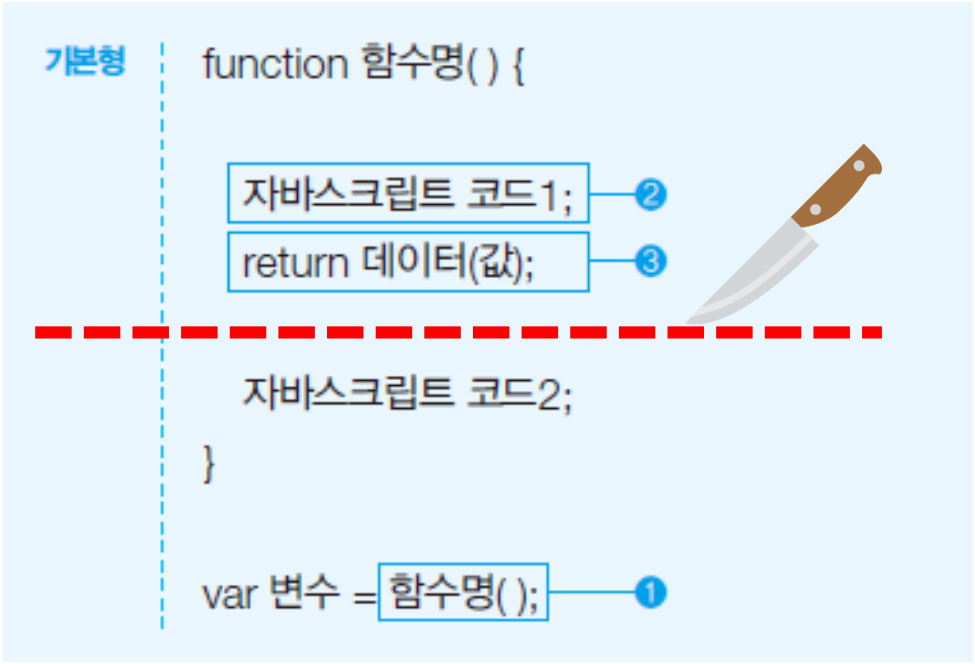
매개변수 없이 함수에 전달된 값 받아오기

```
function sum( ) {
   var sum = arguments[0] + arguments[1] + arguments[2];

   document.write(sum);
}

sum(10, 20, 30);
```



05-2 함수에서 return 문의 역할

데이터를 반환하고 강제 종료하는 return 문



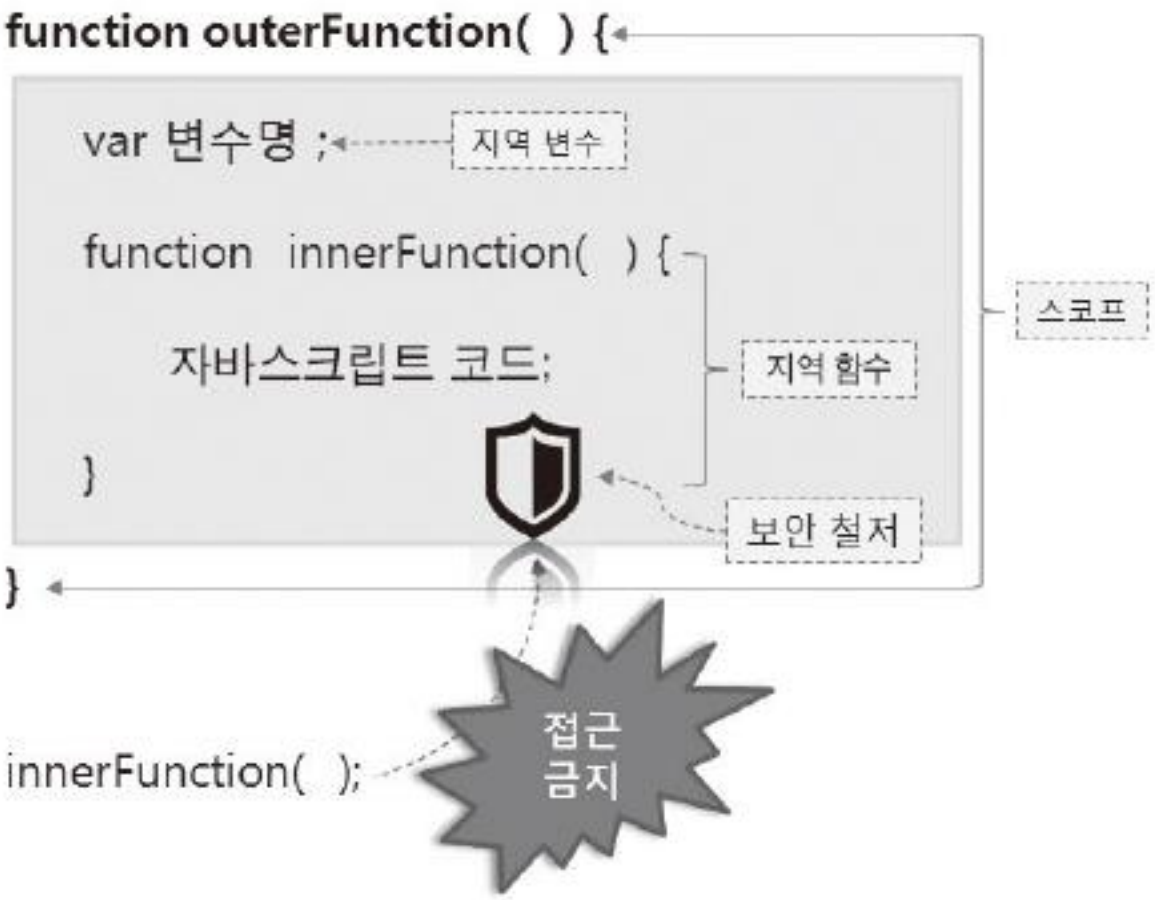
재귀 함수 호출

기본형

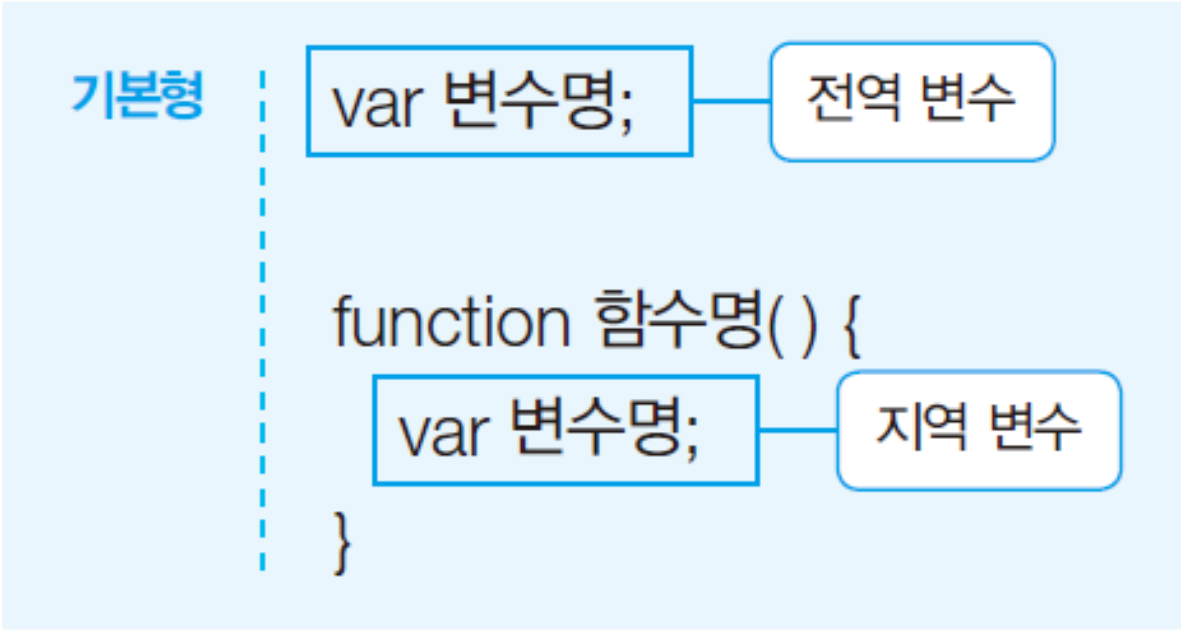
```
function myFnc( ){
    자바스크립트 코드;
    myFnc( );
}
myFnc( );
```

05-3 함수 스코프 개념 이해

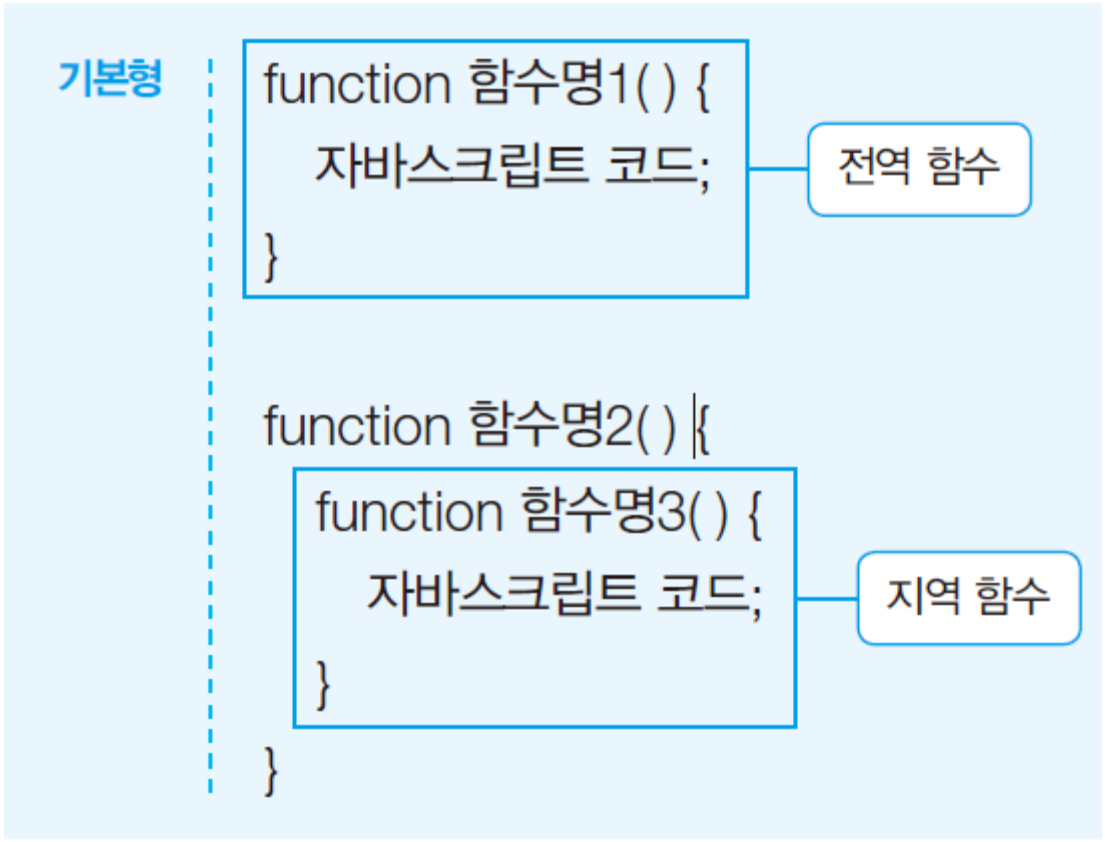
함수 스코프란?


function outerFunction( ) {
var 변수명 ;
지역 변수
function innerFunction( ) {
자바스크립트 코드;
지역 함수
스코프
}
보안 철저
}
innerFunction( );
접근
금지

전역 변수와 지역 변수의 개념과 차이



전역 함수와 지역 함수의 차이



05-4 자바스크립트 내장 함수

내장 함수

종류	설명	사용 예
encodeURI( )	문자를 유니 코드값으로 인코딩합니다. (영문, 숫자, 일부 기호(; , / ? : @ & = + $)는 제외)	encodeURI("?query=값"); → "?query=%EA%B0%91"
encodeURIComponent( )	문자를 유니 코드값으로 인코딩합니다(영문, 숫자 제외).	encodeURIComponent("?query=값") → "%3Fquery%3D%EA%B0%91"
decodeURI( )	유니 코드값을 디코딩해 다시 문자화합니다.	decodeURI("?query=%EA%B0%91") → "?query=값"
decodeURIComponent( )	유니 코드값을 디코딩해 다시 문자화합니다.	decodeURIComponent("%3Fquery%3D%EA%B0%91") → "?query=값"
parseInt( )	문자열 데이터를 정수형 데이터로 반환합니다.	parseInt("5.12") → 5 parseInt("15px") → 15
parseFloat( )	문자열 데이터를 실수형 데이터로 반환합니다.	parseInt("5.12") → 5.12 parseInt("65.5%") → 65.5
String( )	문자형 데이터로 반환합니다.	String(5) → "5"
Number( )	숫자형 데이터로 반환합니다.	Number("5") → 5
Boolean( )	논리형 데이터로 반환합니다.	Boolean(5) → true Boolean(null) → false
isNaN( )	is Not a Number의 약자이며 숫자가 아닌 문자가 포함되어 있으면 true를 반환합니다.	isNaN("5–3") → true isNaN("53") → false
eval( )	문자형 데이터를 따옴표가 없는 자바스크립트 코드로 처리합니다	eval("15 + 5") → 20