

자바스크립트 코어

Core - Javascript

클로저


Closure

함수는 일급객체(First Class Object)

함수 내부에 **함수를 인자로 전달**할 수 있습니다.

```
function() { ... }
```

함수 function ()



반환함수 (return function)

함수 내부에서 **함수를 결과 값으로 반환**할 수 있습니다.

함수 function ()



return function() { ... }

클로저 (Closure)

함수에서 반환된 함수를 실행하면 생성됩니다.

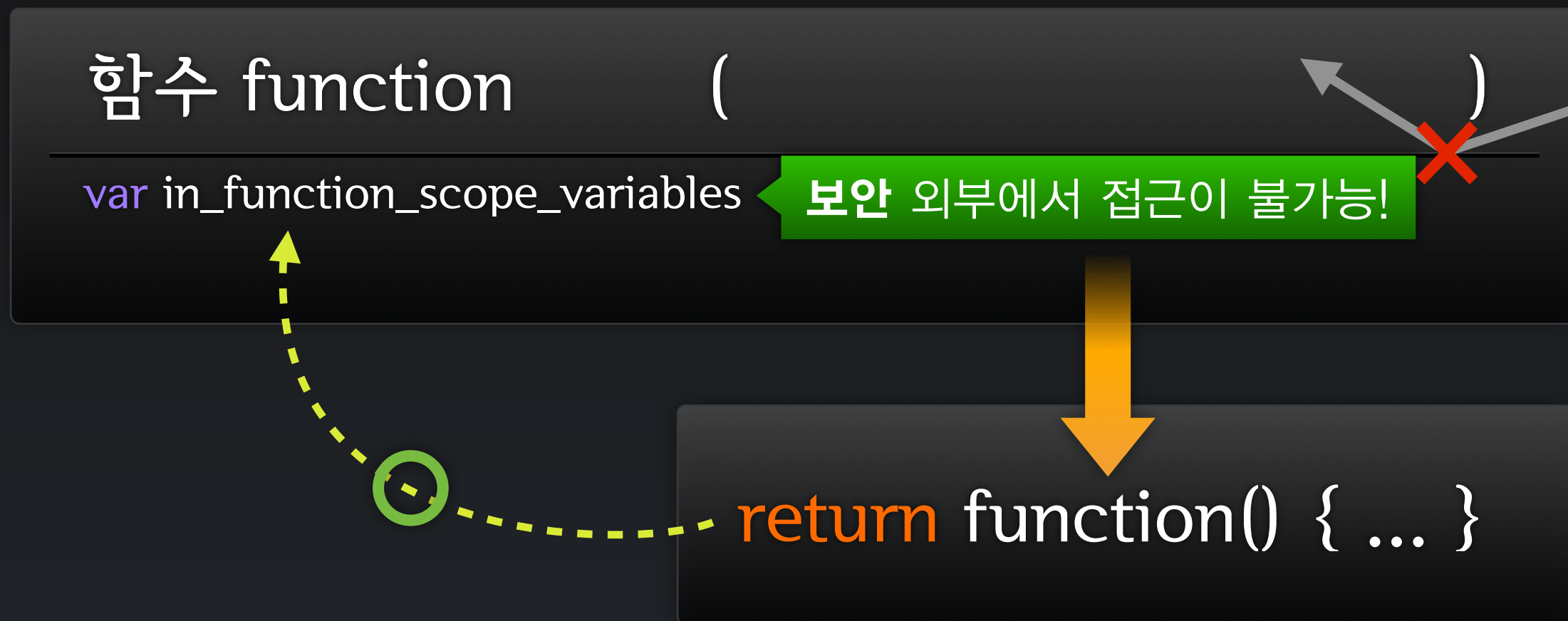
함수 function ()

return function() { ... }

클로저
생성

클로저 (Closure)

클로저는 자신을 탄생시킨 부모 함수의스코프에 접근이 가능합니다.



반환객체 (return Object)

함수 내부에서 객체를 결과 값으로 반환할 수 있습니다.

함수 function (

return { ... }

클로저
생성

클로저 (Closure)

함수에서 객체를 반환한 경우 생성됩니다.

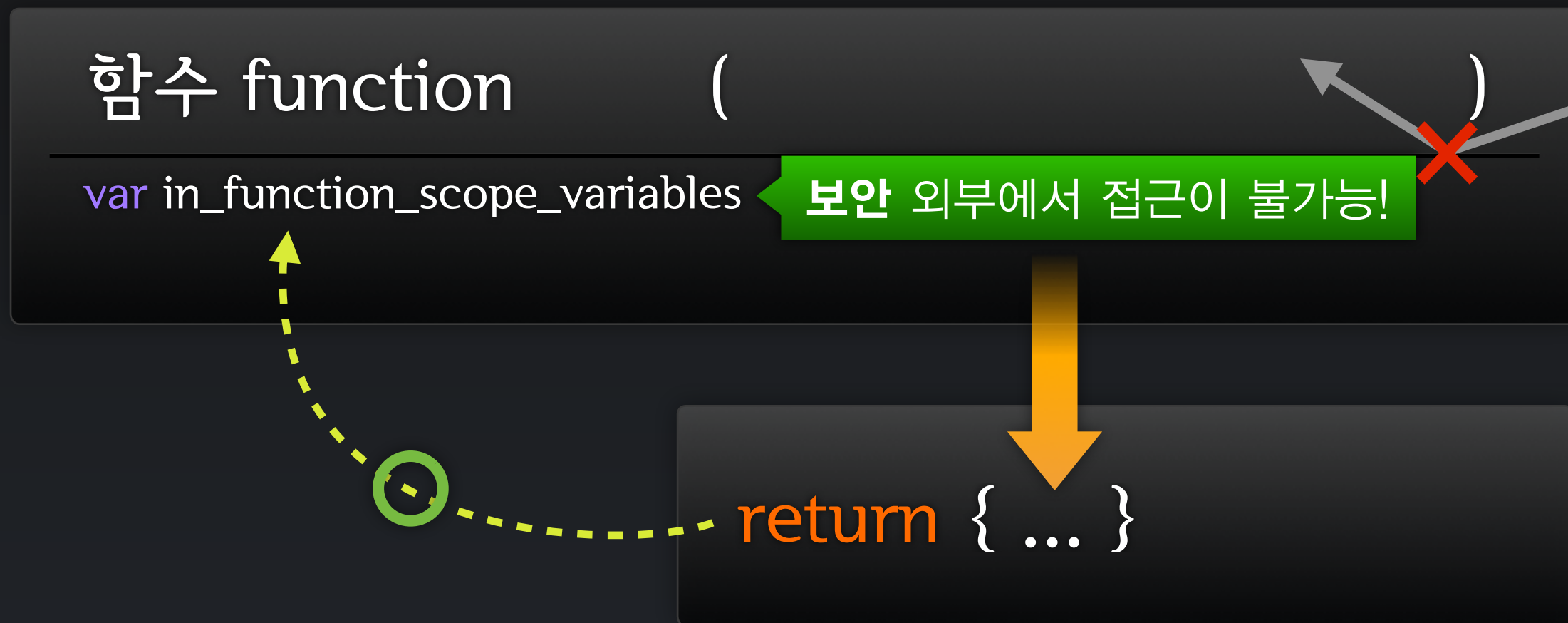
함수 function (

return { ... }

클로저
생성

클로저 (Closure)

클로저는 자신을 탄생시킨 부모 함수의스코프에 접근이 가능합니다.



클로저 (Closure)

클로저는 자신을 탄생시킨 부모 함수의 스코프에 접근이 가능합니다.



가
비지
컬렉

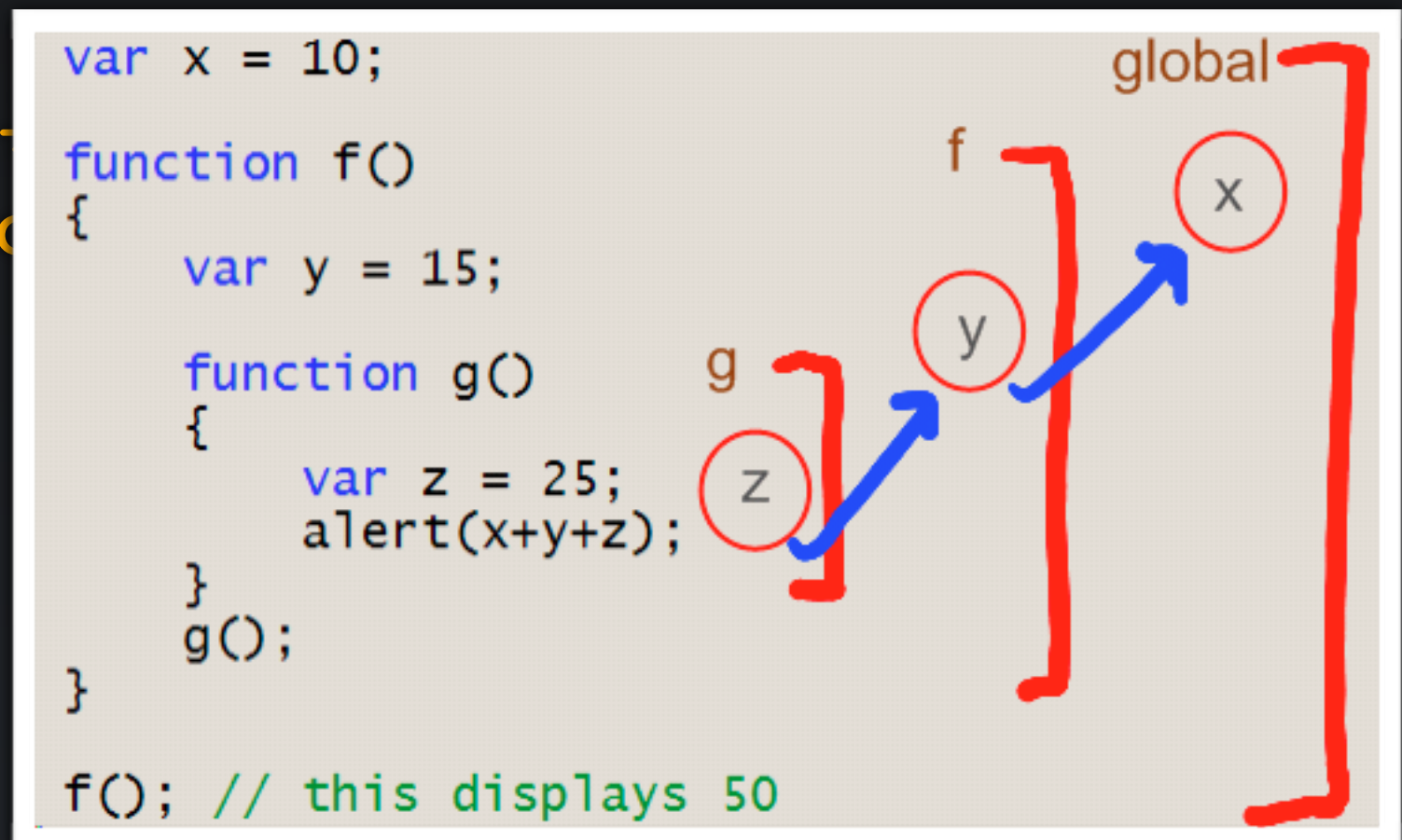
일반적으로 함수가 종료되면 함수의 스코프는 사라진다..
함수가 종료되면서 탄생시킨 클로저에서는 부모 함수의 스코프 영역을 저장합니다.

이 점은 클로저의 장점이 되기도 하지만, 단점도 될 수 있기 때문에 사용에 주의하여야 합니다.

가비지 컬렉션 (Garbage Collection)

자바스크립트는 자동 가비지 컬렉션을 사용하기 때문에 객체가 어떠한 것에도 참조되어 있지 않을 경우, 자동으로 메모리에서 소멸됩니다.

스코프 체인에 더이상 접
이는 가비지 컬렉션 대상



벤 나델의 그림으로 이해해보는 jQuery 안의 Javascript Closure

THE BLOG OF **BEN NADEL**

ON OBSESSIVELY THOROUGH WEB APPLICATION DEVELOPMENT



```
// When the DOM loads, initialize the user interface.
$(
  function(){
    // Gather all the links in the document.
    var jlinks = $("a");

    // Loop over each link.
    jlinks.each(
      function( intLinkIndex, objLink ){
        // Get a jQuery reference to this link.
        var jThis = $( this );

        // Bind the click handler.
        jThis.click(
          function( objEvent ){
            // Alert the link index in the context of the
            // greater document.
            alert(
              "I am link " +
              (intLinkIndex + 1) +
              " of " +
              jlinks.length
            );

            // Prevent default link behavior.
            return( false );
          }
        );
      }
    );
  }
);
```