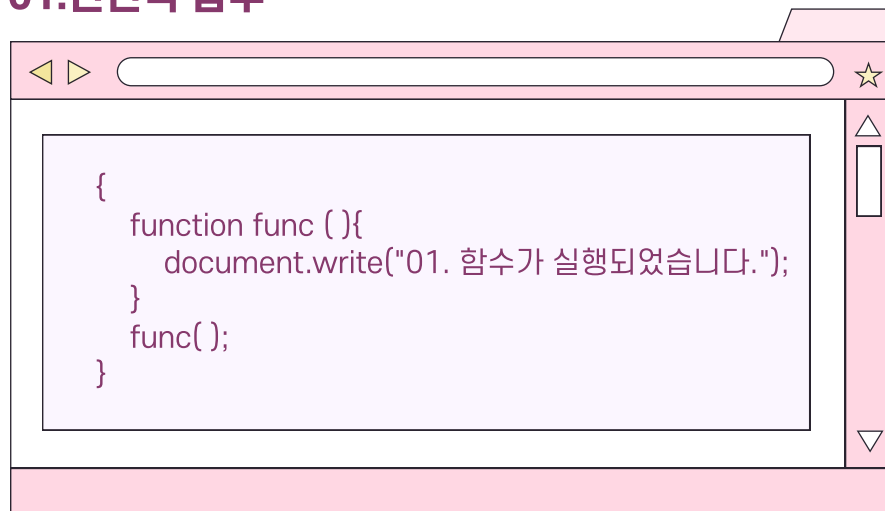


데이터 실행하기

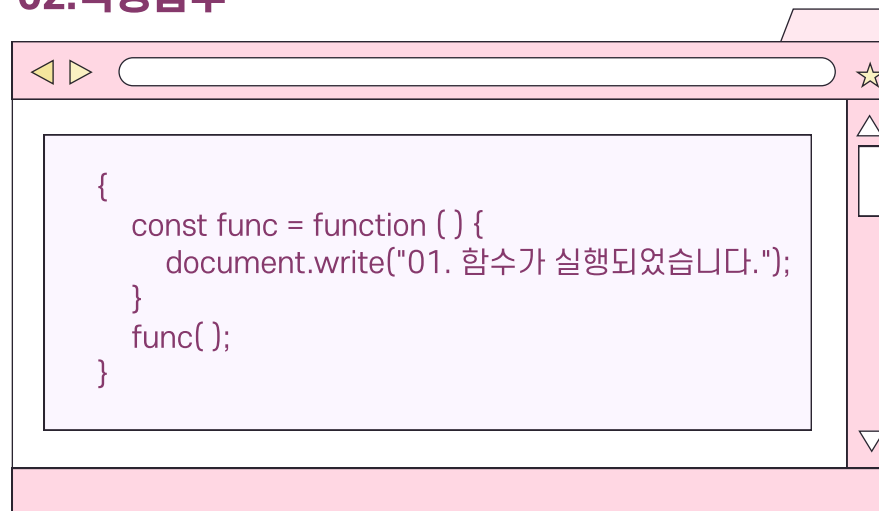
01.선언적 함수



함수가 실행되었습니다.

선언적 함수를 사용하여, func안에 “함수가 실행되었습니다”라는 문자가 도출될수 있도록 함수를 설정해 주었다.

02.익명함수

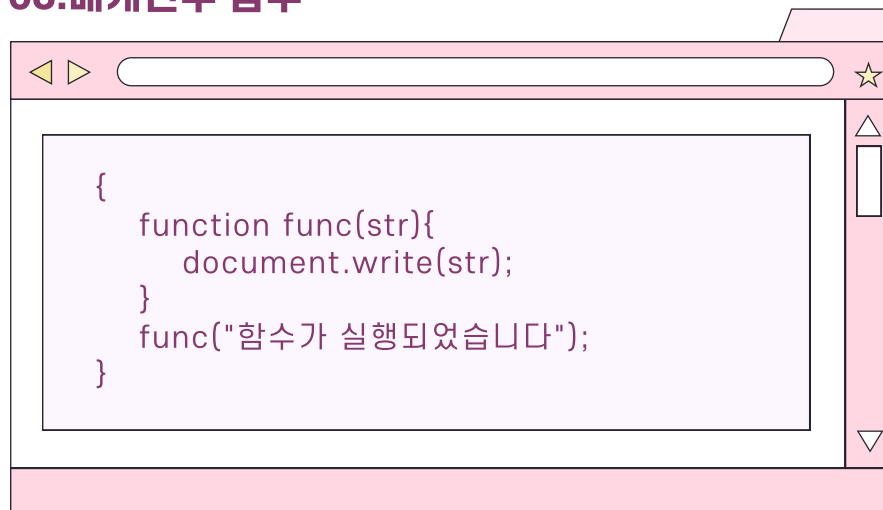


함수가 실행되었습니다.

변수를 사용하여 func를 익명으로 만들어주어 함수의 이름이 없는것을 익명함수라고 한다.

데이터 실행하기

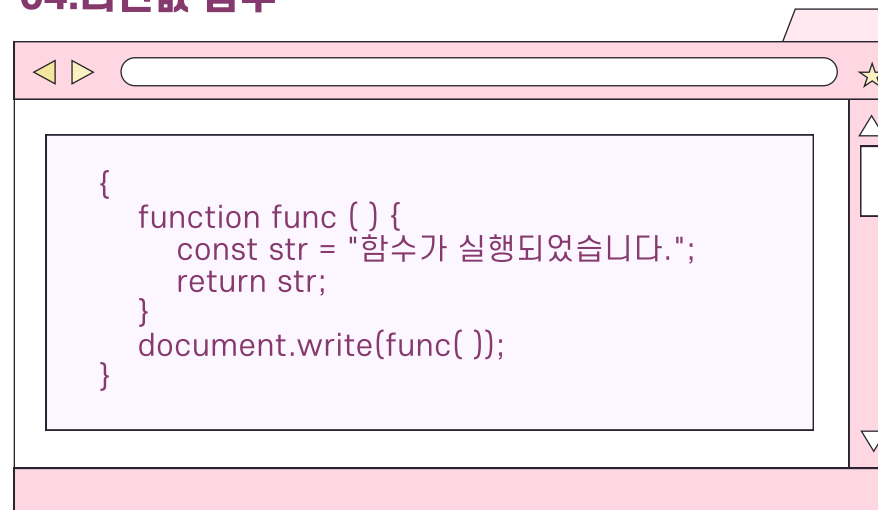
03. 매개변수 함수



함수가 실행되었습니다.

매개변수란 함수를 호출할 때 인수로 전달된 값을 함수 내부에서 사용할 수 있게 해주는 변수이다. 또한, 함수의 인수(argument)란 함수가 호출될 때 함수로 값을 전달해주는 변수를 가리킨다.

04. 리턴값 함수

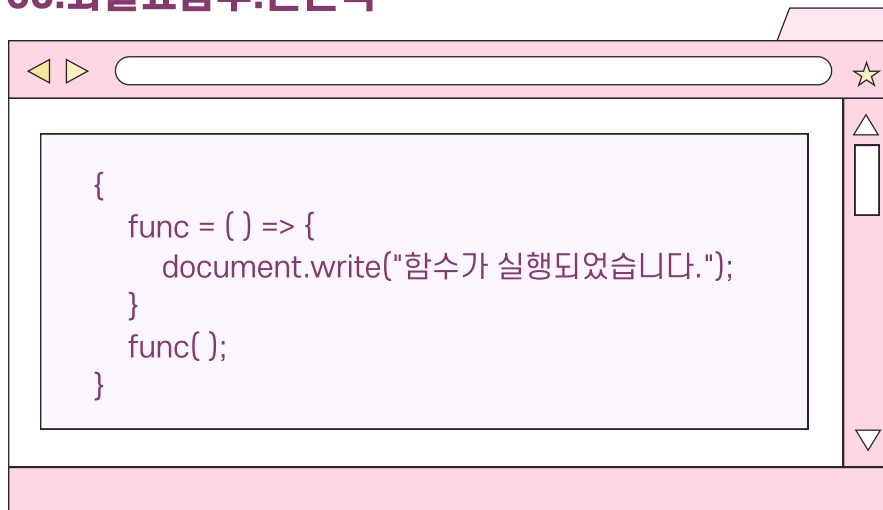


함수가 실행되었습니다.

리턴값이란 함수가 어떠한 기능을 수행하고 그 결과를 호출한 곳으로 돌려주는 값을 말한다. 함수가 어떠한 기능을 수행하고 돌려줄 값이 있으면 return 명령을 사용한다.

데이터 실행하기

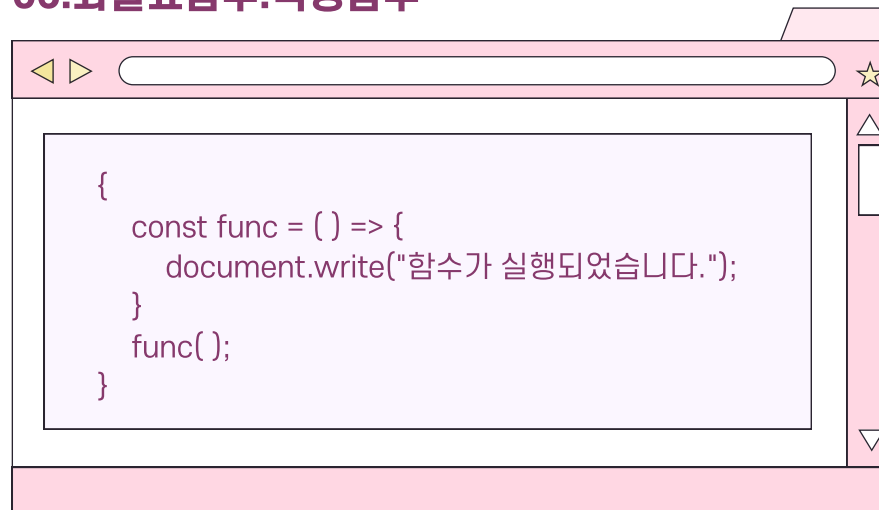
05. 화살표 함수: 선언적



함수가 실행되었습니다.

선언적 함수를 화살표 함수로 축약 한것으로 function(){} 대신 function생략후 ()와 {}사이에 => 화살표를 넣어줌으로서 함수가 완성된다.

06. 화살표 함수: 익명 함수

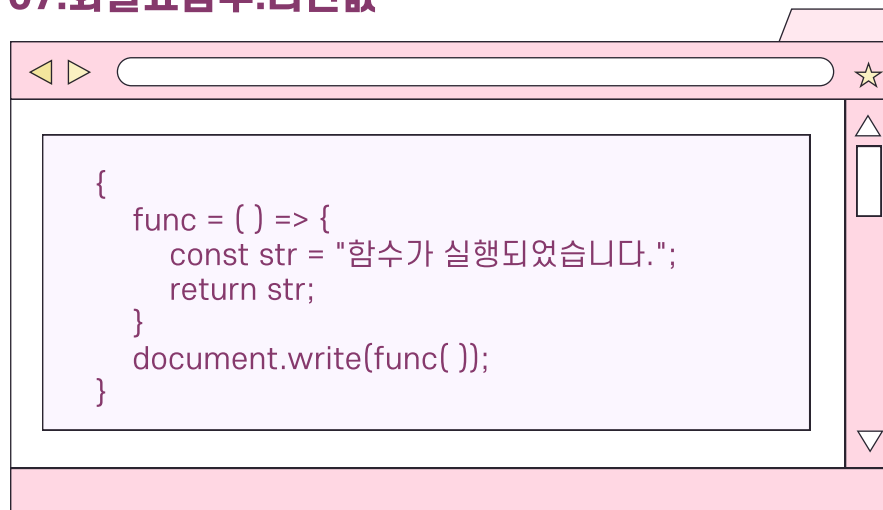


함수가 실행되었습니다.

변수를 사용하여 func를 익명으로 만들어주어 그후 화살표 함수를 사용하였다. 화살표 함수를 사용하는 이유는 간결한 구문으로 콜백을 정리할때 유용하다.

데이터 실행하기

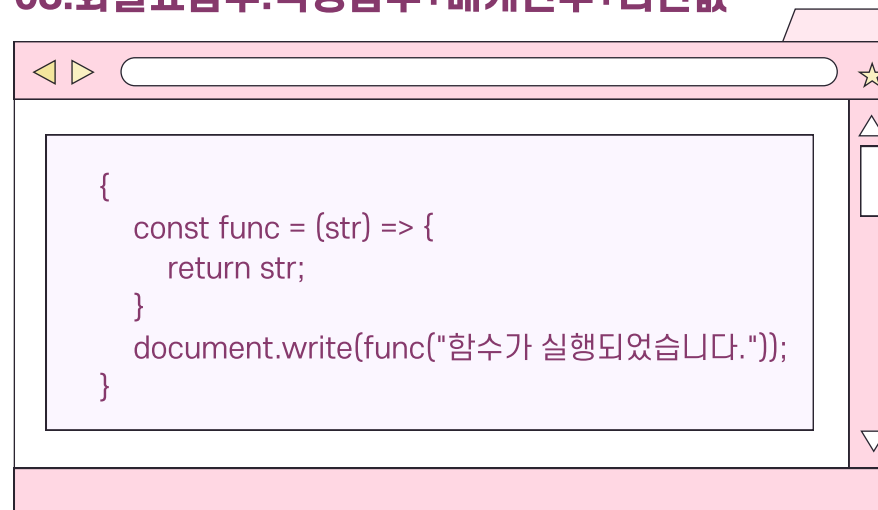
07. 화살표 함수: 리턴값



함수가 실행되었습니다.

리턴값이 있는 함수를 화살표 함수로 사용하게 되면 이렇게 사용한다.

08. 화살표 함수: 익명 함수 + 매개변수 + 리턴값

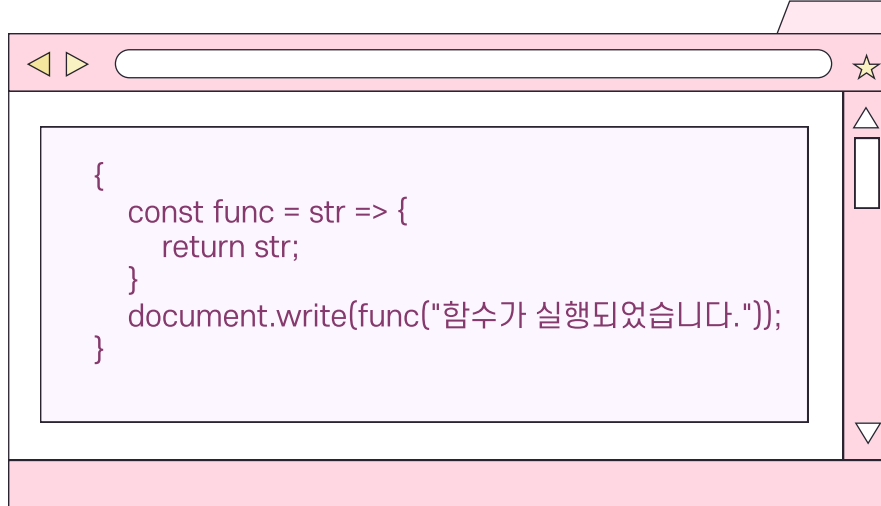


함수가 실행되었습니다.

매개변수 str을 사용하여 리턴을 해줌으로서, 화살표 함수를 사용하기 전보다 보다 간결하게 코드를 작성할 수 있다. 매개변수가 한개 일때 괄호를 생략 할 수 있다.

데이터 실행하기

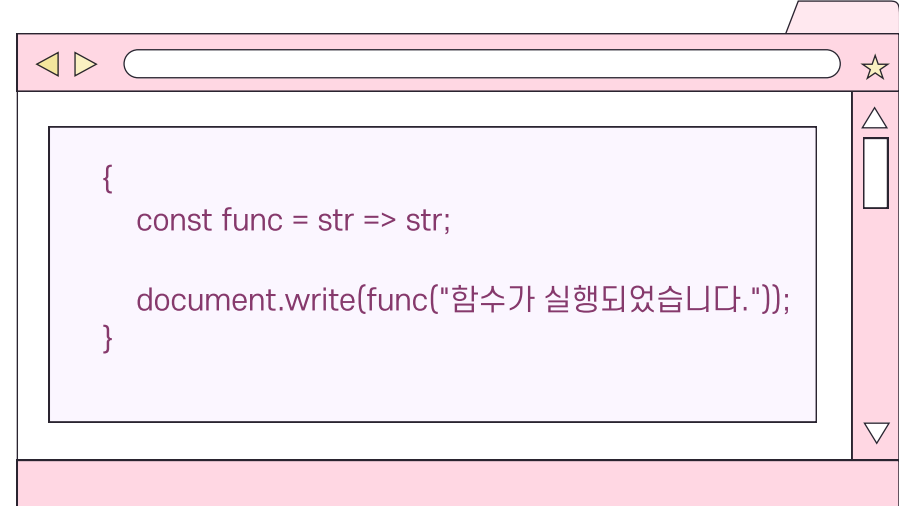
09. 화살표 함수 : 익명 + 매개변수 + 리턴값 + 괄호생략



함수가 실행되었습니다.

function(){}이라고 작성할 때에는 괄호는 생략 불가능한 것이지만, 화살표 함수를 적용했을 때에는, 괄호도 생략하여 더 간결하게 표현할 수 있다.

10. 화살표 함수 : 익명 + 매개변수 + 리턴값 + 괄호/리턴 생략

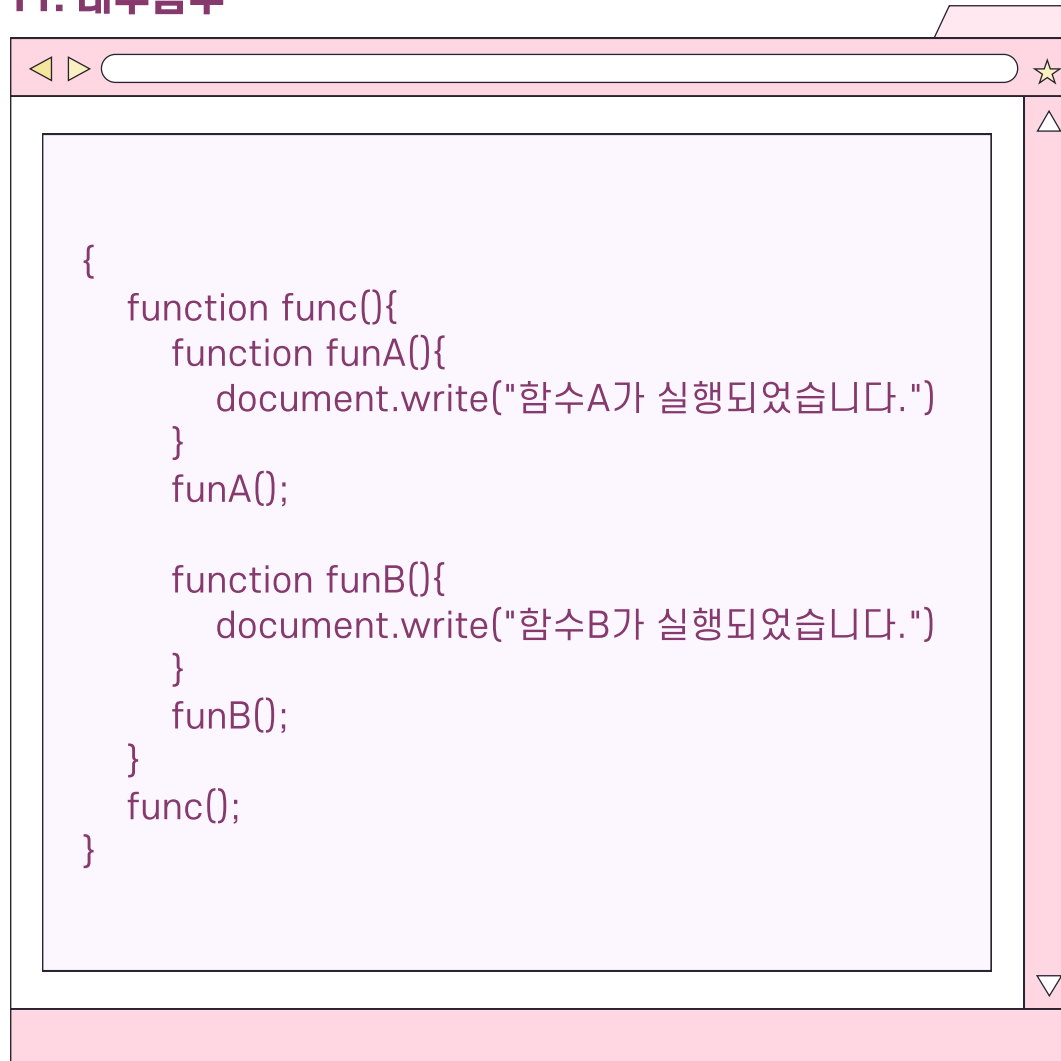


함수가 실행되었습니다.

괄호생략에 리턴값도 생략한 것으로서 함수를 한줄로 작성할 수 있다.

데이터 실행하기

11. 내부함수



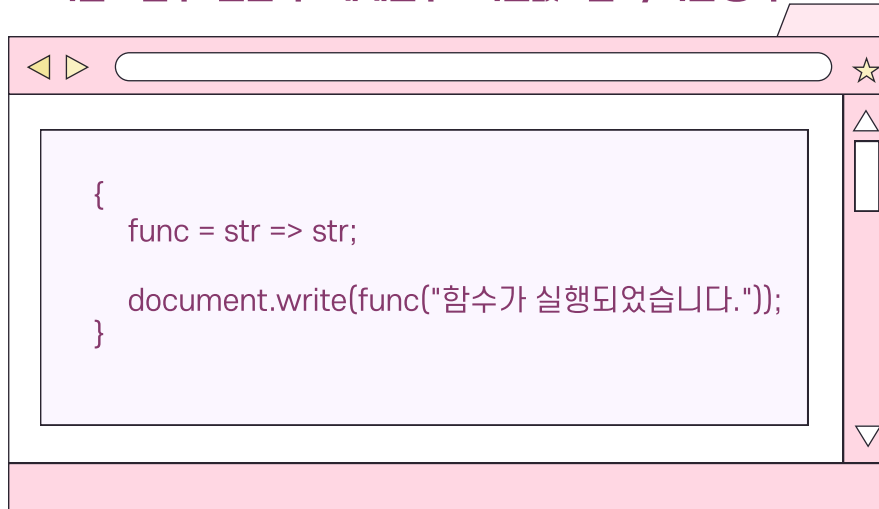
함수A가 실행되었습니다.

함수B가 실행되었습니다.

hasOwnProperty()는 boolean으로서 프로퍼티를 포함 하고있는지 판단한다.예기치 못한 상황에 따른 버그 대비와 코드의 가독성 향상 등을 위해 사용된다.

데이터 실행하기

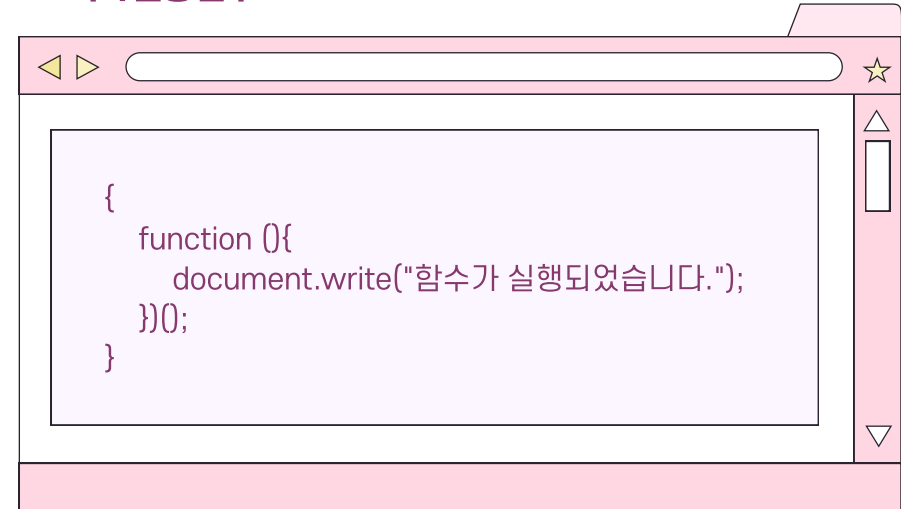
12. 화살표 함수: 선언적 + 매개변수 + 리턴값 + 괄호/리턴 생략



함수가 실행되었습니다.

화살표 함수중에 생략할수 있는 모든것을 다 생략한 버전으로서 function이나 ()괄호가 들어가지 않아도 함수라는 것을 잘 알아 보아야 한다.

13. 즉시 실행 함수

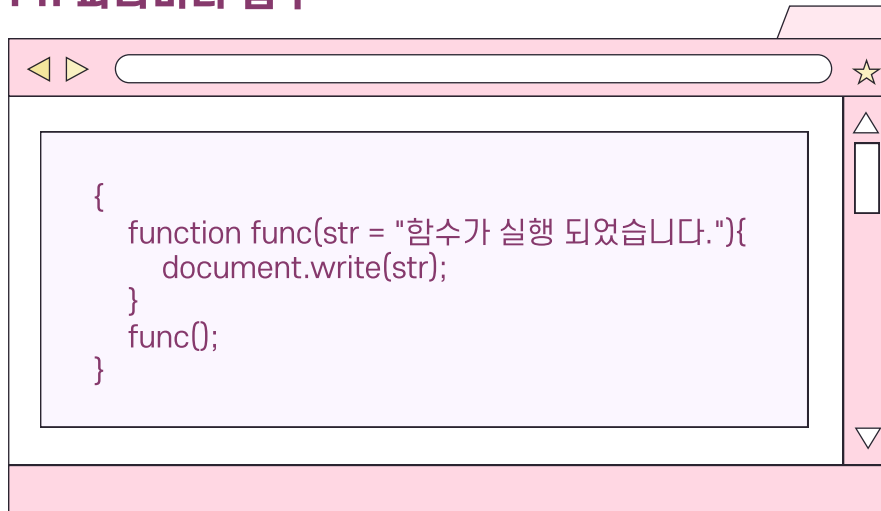


함수가 실행되었습니다.

화살표 함수 : 익명 + 매개 + 리턴

데이터 실행하기

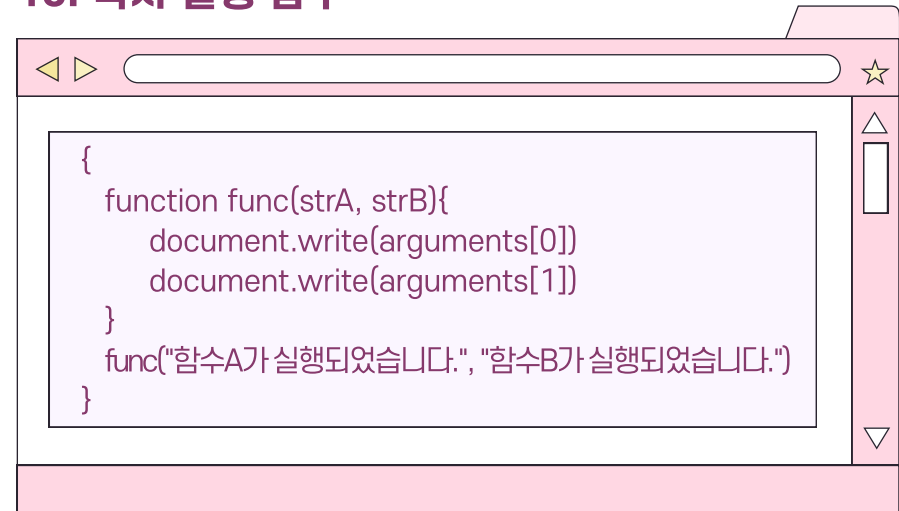
14. 파라미터 함수



함수가 실행되었습니다.

화살표 함수중에 생략할수 있는 모든것을 다 생략한 버전으로서 function이나 ()괄호가 들어가지 않아도 함수라는 것을 잘 알아 보아야 한다.

13. 즉시 실행 함수



함수A가 실행되었습니다.
함수B가 실행되었습니다.

화살표 함수 : 익명 + 매개 + 리턴