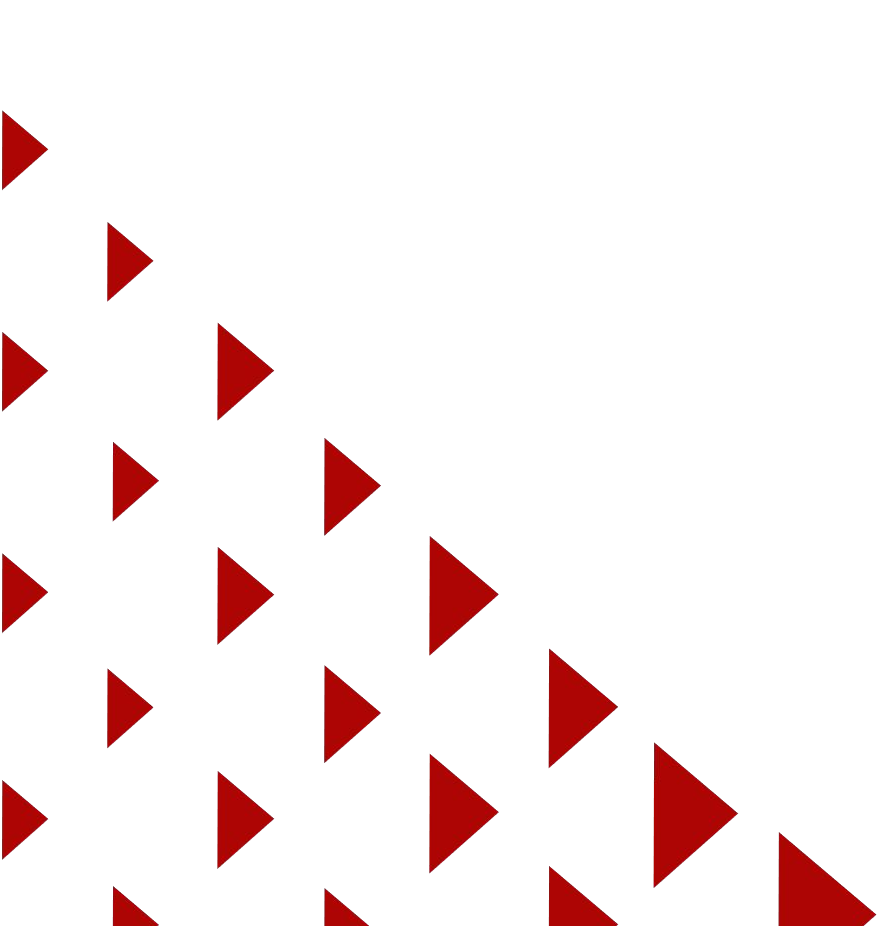
**SESSION 10: FUNCTION**

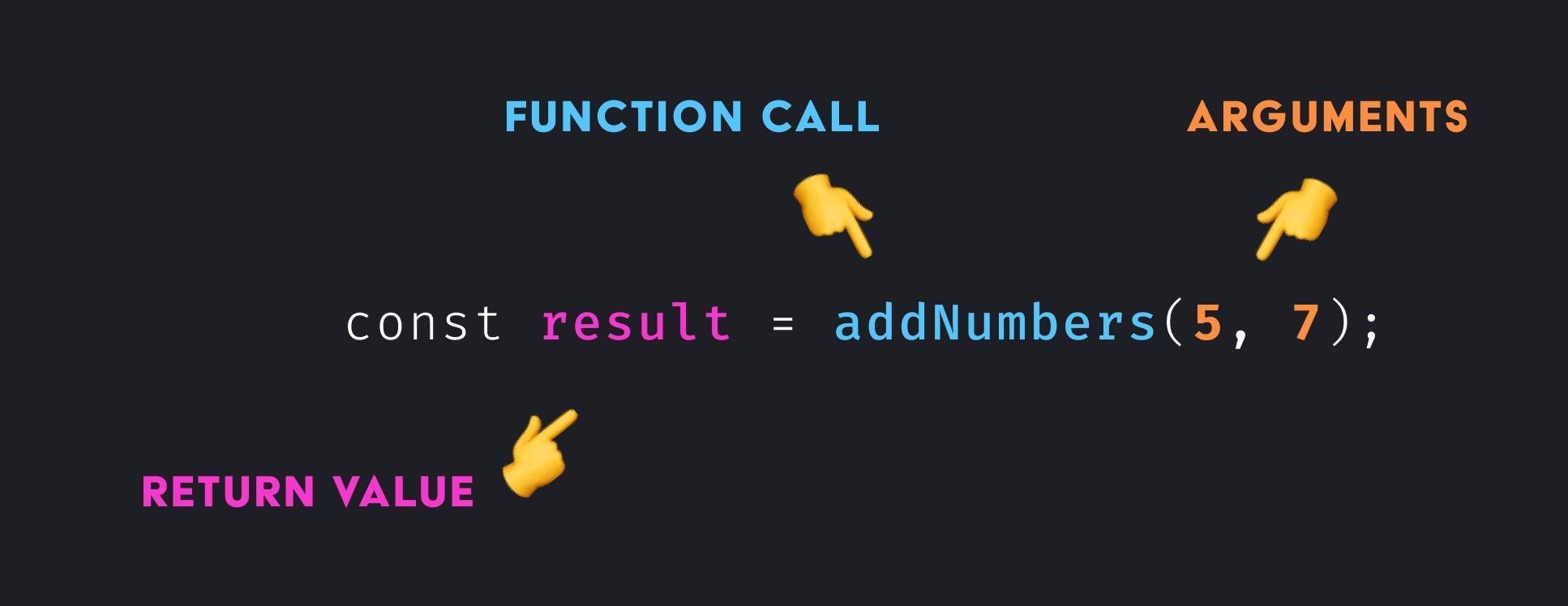
Module 1: Web Front-end Fundamental Version: 3.0





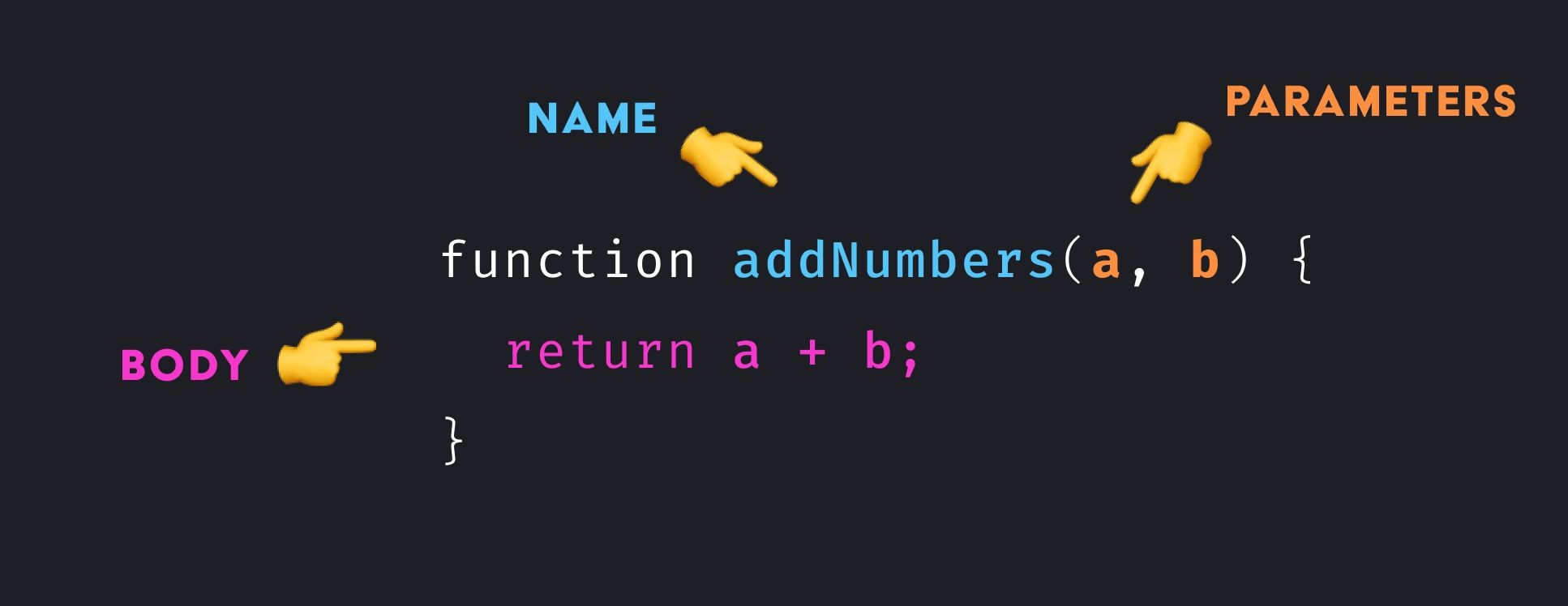
**NỘI DUNG**

1. **Tổng quan function**
2. **Các loại function**



# TỔNG QUAN FUNCTION - 1

## Định nghĩa

* + Hàm trong javascript là một **chương trình con** bao gồm một **tập hợp các câu lệnh** thực hiện **một tác vụ** cụ thể
  + Có thể tái sử dụng và bảo trì tốt hơn giúp nâng cao hiệu suất công việc

## Khai báo hàm

**function functionName(param1,param2,…..){ Statements;**

**}**

* *Trong đó:*
  + function: từ khóa khai báo hàm
  + functionName: tên hàm, quy tắc đặt tên giống đặt tên biến (nên là động từ)
  + param1, param2…: các tham số của hàm (0…N tham số)
  + Statements: tập hợp các câu lệnh nằm trong thân hàm được bao quanh bởi {}
  + Hàm được hoisting



# TỔNG QUAN FUNCTION - 2

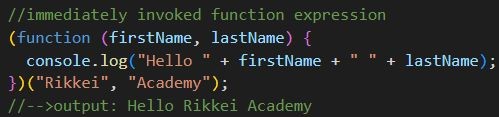
## Tham số và đối số

* **Tham số**
  + Biến cục bộ của hàm được khai báo khi khai báo hàm
  + Hàm có thể có 0, 1 hay nhiều tham số
* **Đối số**
  + Giá trị truyền vào khi gọi hàm
  + Tham trị
    - Truyền giá trị có kiểu dữ liệu là primitive
    - Sao chép (copy) giá trị của biến truyền vào sang tham số của hàm)
  + Tham chiếu
    - Truyền giá trị có kiểu dữ liệu là non\_primitive như Array, Object, Function.
    - Truyền địa chỉ ô nhớ vào tham số của hàm

## Gọi function

* + Gọi tên và truyền các đối số tương ứng với tham số của hàm
  + Hàm kết thúc khi gặp **return**

## Function Expression – biểu thức hàm



* + Function được định nghĩa và lưu trữ trong một biến như kiểu giá trị
  + Cú pháp

**var | let | const variableName = function(parameters){ Statements;**

**}**

* + Gọi hàm như thông thường

## Immediately invoked function expression (IIFE)

* + Là Function Expression được thực thi ngay sau khi định nghĩa
  + Cú pháp

**(function(parameters){ Statements;})(arguments)**

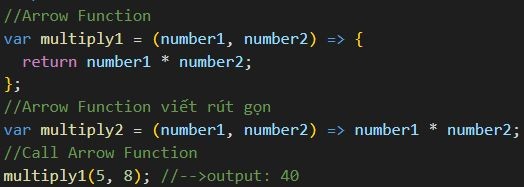
* + Sử dụng khi xử lý logic mà chỉ chạy đúng 1 lần

## Arrow function

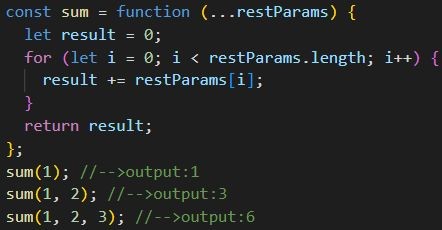
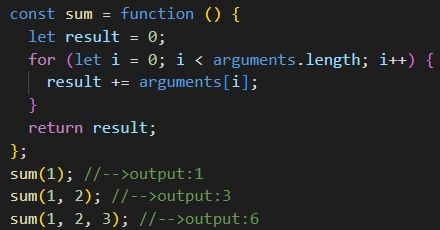
* + Cách viết function mới ra mắt trong ES6
  + Giúp tiết kiệm thời gian code cũng như đơn giản hóa phạm vi function
  + Không phụ thuộc vào từ khóa this
  + Cú pháp

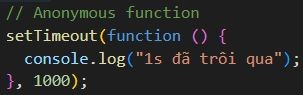
**var | let | const functionName = (parameters) =>{ Statements;**

**}**



## Function có nhiều tham số chưa xác định (Rest parameter – ES6)



* **Anonymous function – Hàm ẩn danh**
  + Hàm ẩn danh không có tên
  + Cú pháp:

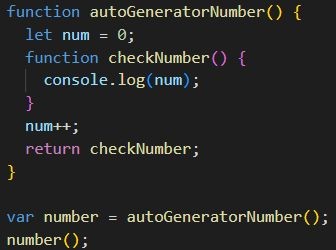
**function(){**

**Statements;**

**}**

## Closures

* Cho phép lồng các function trong nhau

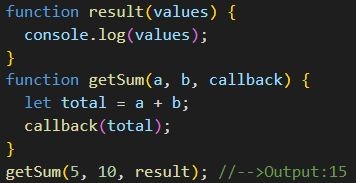


* Các function con được định nghĩa bên trong fuction cha chứa nó
* Function con truy cập được các biến của function cha
* Function cha không thể truy cập biến của function con
* Cú pháp:

**function father\_func\_name(parameters){ function children\_func\_name(parameters){**

© 2023 By Rikkei Academy - Rikkei Education - All rights reserved.

## Callback function

* + Truyền đối số là một hàm
  + Được sử dụng:
    - Xử lý đồng bộ trong javascript
    - Xử lý event
  + Cú pháp:

**function func\_A(parameters){ Statements;**

**}**

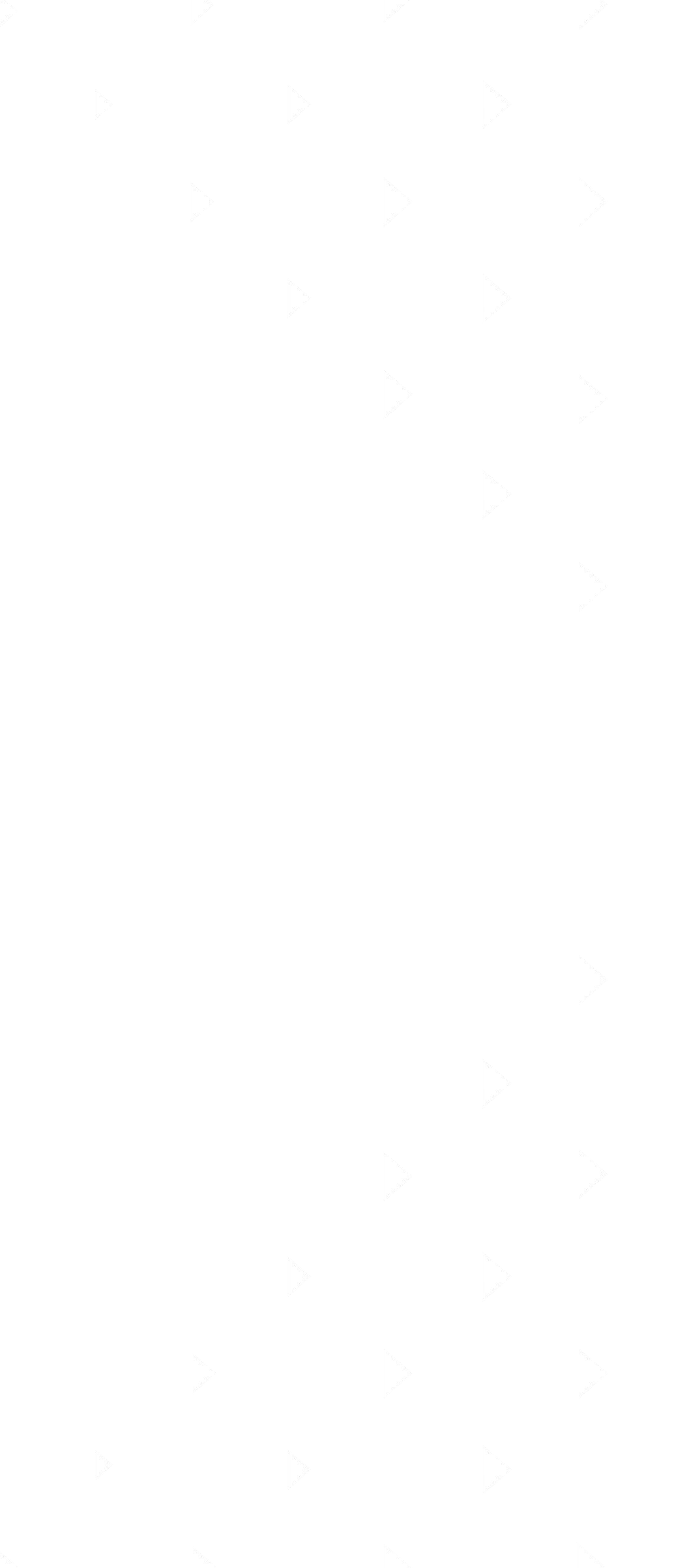
**function func\_B(parameters,callback){ Statements;**

**callback(arguments);**

**}**

**func\_B(arguments,func\_A);**

© 2023 By Rikkei Academy - Rikkei Education - All rights reserved.



**KẾT THÚC**

HỌC VIỆN ĐÀO TẠO LẬP TRÌNH CHẤT LƯỢNG NHẬT BẢN

© 2023 By Rikkei Academy - Rikkei Education - All rights reserved.