TÁC GIẢ: 8 SYNC

CỘNG ĐỒNG

- Kevin Nguyễn
- Nhóm Chia Sẻ Công Nghệ
- Nhóm BlockChain
- Tiktok: 8 Sync
- Youtube: 8 Sync Dev
- Zalo

KHÓA HỌC:

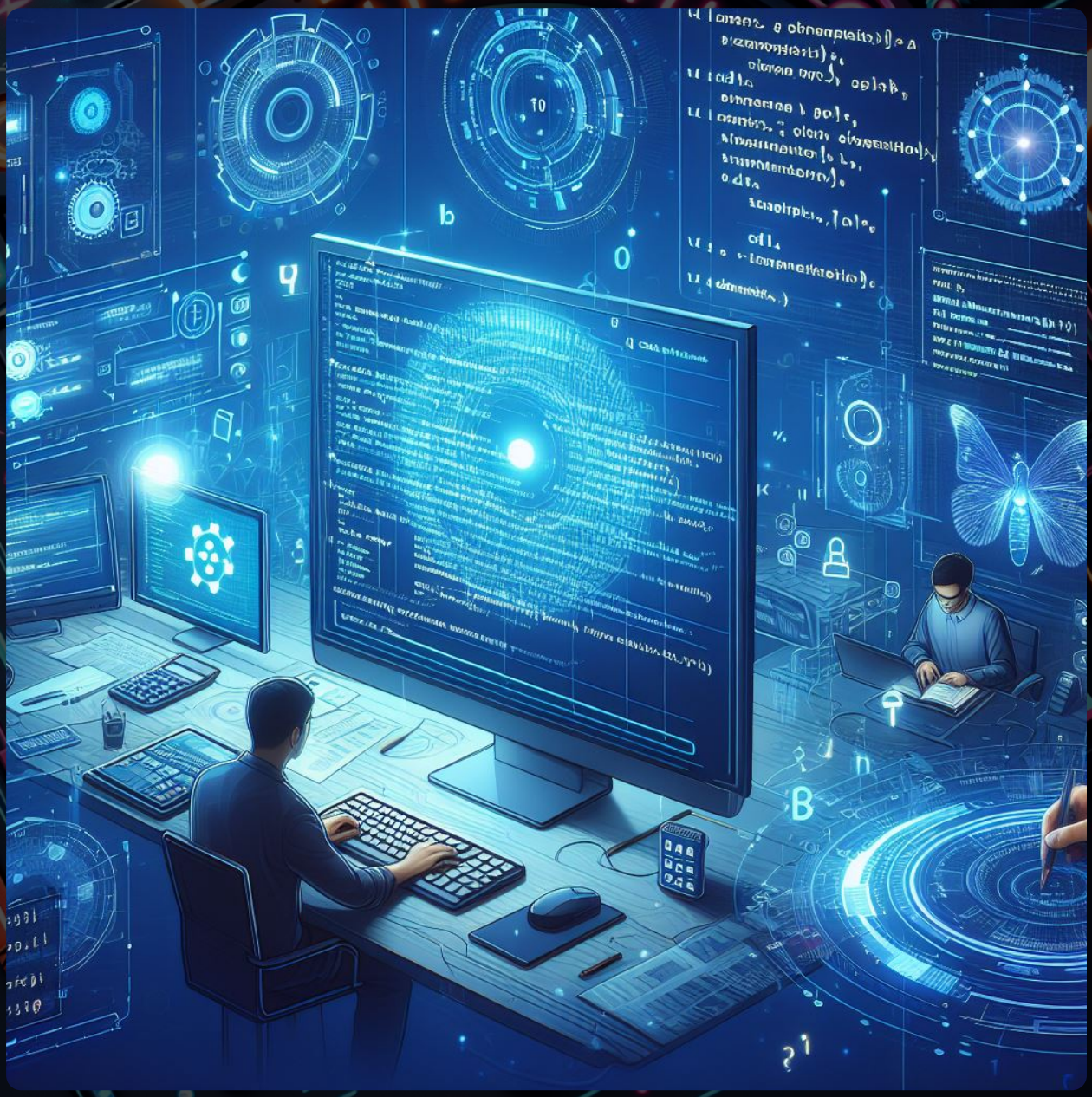
- Fullstack Python
- Fullstack Nextjs
- Fullstack Android-IOS

Tài liệu sẽ được cập nhật định kì và thông báo trong group nên các bạn chú ý nhen .

BÀI 0. HÀM TRONG PYTHON

Trong Python, một hàm là một khối mã có tổ chức, có thể tái sử dụng được, được sử dụng để thực hiện một hành động duy nhất, liên quan. Các hàm cung cấp tính tách biệt tốt hơn cho ứng dụng của bạn và một mức độ cao của việc tái sử dụng mã.

Một phương pháp từ trên xuống để xây dựng logic xử lý liên quan đến việc định nghĩa các khối hàm tái sử dụng độc lập. Một hàm Python có thể được gọi từ bất kỳ hàm nào khác bằng cách chuyển dữ liệu cần thiết (gọi là tham số hoặc đối số). Hàm gọi trả về kết quả của nó lại cho môi trường gọi.



LOẠI HÀM TRONG PYTHON

Python cung cấp các loại hàm sau:

1. Hàm được tích hợp sẵn.
2. Hàm được định nghĩa trong các mô-đun được tích hợp sẵn.
3. Hàm do người dùng tự định nghĩa.

Thư viện tiêu chuẩn của Python bao gồm một số hàm được tích hợp sẵn. Một số hàm được tích hợp sẵn trong Python là print(), int(), len(), sum(), v.v. Những hàm

này luôn có sẵn, vì chúng được tải vào bộ nhớ máy tính ngay khi bạn bắt đầu trình thông dịch Python.

Thư viện tiêu chuẩn cũng đóng gói một số mô-đun. Mỗi mô-đun xác định một nhóm các hàm. Những hàm này không sẵn có ngay lập tức. Bạn cần phải nhập chúng vào bộ nhớ từ các mô-đun tương ứng của họ.

Ngoài các hàm được tích hợp sẵn và các hàm trong các mô-đun được tích hợp sẵn, bạn cũng có thể tạo ra các hàm của riêng bạn. Những hàm này được gọi là hàm do người dùng tự định nghĩa.

ĐỊNH NGHĨA MỘT HÀM PYTHON

Bạn có thể định nghĩa các hàm tùy chỉnh để cung cấp chức năng cần thiết. Dưới đây là các quy tắc đơn giản để định nghĩa một hàm trong Python:

- Khối hàm bắt đầu với từ khóa def tiếp theo là tên hàm và dấu ngoặc đơn ().
- Bất kỳ tham số đầu vào hoặc đối số nào cũng phải được đặt trong các dấu ngoặc đơn này. Bạn cũng có thể định nghĩa các tham số bên trong các dấu ngoặc đơn này.
- Câu lệnh đầu tiên của một hàm có thể là một câu lệnh tùy chọn; chuỗi tài liệu của hàm hoặc docstring.
- Khối mã bên trong mỗi hàm bắt đầu bằng một dấu hai chấm (

  😀

  và được thụt lề.
- Câu lệnh return [biểu thức] kết thúc một hàm, tùy chọn truyền lại một biểu thức cho người gọi. Một câu lệnh return không có đối số tương đương với return None.

CÚ PHÁP ĐỂ ĐỊNH NGHĨA MỘT HÀM PYTHON

< 8 Sync Dev />

```
def tên_hàm(tham_số):
   "chuỗi_tài_liệu_hàm"
```

```
    khối_mã_hàm
    return [biểu_thức]
```

Mặc định, các tham số có hành vi vị trí và bạn cần thông báo cho chúng theo cùng một thứ tự mà chúng đã được định nghĩa.

Sau khi hàm được định nghĩa, bạn có thể thực thi nó bằng cách gọi nó từ một hàm khác hoặc trực tiếp từ dấu nhắc Python.

VÍ DỤ VỀ ĐỊNH NGHĨA MỘT HÀM PYTHON

Dưới đây là một ví dụ về cách định nghĩa một hàm greetings(). Dấu ngoặc đơn là trống nên không có tham số nào.

Dòng đầu tiên là chuỗi tài liệu. Khối hàm kết thúc bằng câu lệnh return. khi hàm này được gọi, thông báo Hello World sẽ được in ra.

< 8 Sync Dev />

```
def greetings():
    "Đây là chuỗi tài liệu của hàm greetings"
    print("Hello World")
    return

greetings()
```

GỌI MỘT HÀM PYTHON

Định nghĩa một hàm chỉ đưa ra tên cho nó, xác định các tham số cần được bao gồm trong hàm và cấu trúc các khối mã.

Khi cấu trúc cơ bản của một hàm được hoàn thành, bạn có thể thực thi nó bằng cách gọi nó từ một hàm khác hoặc trực tiếp từ dấu nhắc Python.

VÍ DỤ VỀ GỌI MỘT HÀ

m Python

Dưới đây là ví dụ về cách gọi hàm printme():

< 8 Sync Dev />

```
# Định nghĩa hàm ở đây
def printme(str):
   "Hàm này in chuỗi đã được truyền vào"
   print(str)
   return;

# Bây giờ bạn có thể gọi hàm printme
printme("Đây là cuộc gọi đầu tiên đến hàm được xác định bởi người dùng!")
printme("Lần gọi thứ hai đến cùng một hàm")
```

Khi mã trên được thực thi, nó sẽ tạo ra đầu ra sau:

< 8 Sync Dev />

```
Đây là cuộc gọi đầu tiên đến hàm được xác định bởi người dùng!
Lần gọi thứ hai đến cùng một hàm
```