Functions

Basic Python for Cybersecurity

Hal Teknis Terkait Function

Definisi

- blok code yang memiliki fungsi spesifik dan dapat dipakai berulang (reusable)
- ada parameter/input (opsional)
- ada output (opsional)
- function didefinisikan sekali dan dapat dipanggil berkali-kali

Contoh syntax

```
# definisikan function
def greet(name):
  return f"Hello, {name}!"

# panggil function di manapun setelah definisi
print(greet("Sepuh"))
# Output: Hello, Sepuh!
```

Lambda Function

- function dengan syntax spesial untuk operasi pendek (*one-liner*)
- tidak perlu diberi nama fungsi

```
# function standar
def add(x, y):
    return x + y

# dalam bentuk lambda function
add = lambda x, y: x + y
print(add(2, 3))  # Output: 5
```

Behavior dari Function

- function baru bisa dipanggil setelah didefinisikan

```
say_hello()  # ❌ NameError

def say_hello():
  print("Hello")
```

- panggil fungsi sesuai jumlah parameter, atau kena error

```
def add(a, b):
    return a + b

add(5)  # ❌ TypeError: missing 1 required positional argument
```

- jangan lupa return jika ingin ada output

```python
def add(a, b):
  result = a + b  # No return

x = add(1, 2)
print(x)  # None
```

- jika function didefinisikan lebih dari sekali, yang dipakai adalah yang terakhir

```python
def greet():
    print("Hello")

def greet():  # override yang di atas
    print("Hi")

greet()  # Output: Hi
```

- function bisa dibungkus function lain

```python
result = str(len("cybersecurity"))
print(result)  # Output: "13" (string)
```

- function bisa memanggil dirinya sendiri ⟶ **recursion**

```python
def countdown(n):
  if n <= 0:
    print("0")
  else:
    print(f"{n}")
    countdown(n-1)
```

- function bisa jadi parameter/input function lain

```python
def shout(text):
  return text.upper()

def whisper(text):
  return text.lower()

def speak(func, message):
  return func(message)

print(speak(shout, "hello"))  # Output: HELLO
print(speak(whisper, "HELLO"))  # Output: hello
```

- function bisa disimpan sebagai variable

Contoh

Studi Kasus: Refactor Script Analisis Log

- Kamu sebagai peneliti cybersecurity sedang memeriksa log dan perlu cek apakah ada data confidential yang tercantum.
- Adanya data confidential di log adalah praktik yang buruk karena bisa saja ada pihak yang memanfaatkan informasi tersebut.
- Sebagai langkah awal, kamu diminta untuk identifikasi data confidential dari key.
- Di tahapan pertama, kamu berhasil membuat program seperti ini

```
log_data = """
[2025-06-05 10:00:12] INFO  Starting service on port 8080
[2025-06-05 10:00:13] INFO  Connected to database at 127.0.0.1:5432
[2025-06-05 10:00:14] INFO  Environment: production
[2025-06-05 10:00:15] DEBUG Loaded configuration file: config.yml
[2025-06-05 10:00:16] INFO  Health check: OK
[2025-06-05 10:00:20] INFO  Received request: POST /api/v1/login
[2025-06-05 10:00:20] DEBUG Request payload: {"username": "admin", "password": "hunter2"}
[2025-06-05 10:00:21] INFO  Auth success for user admin
[2025-06-05 10:00:30] INFO  Received request: GET /api/v1/profile
[2025-06-05 10:00:30] DEBUG Auth token: Bearer abcdef1234567890abcdef
[2025-06-05 10:00:32] INFO  Request processed in 34ms
[2025-06-05 10:00:35] INFO  Scheduled job 'backup' started
[2025-06-05 10:00:35] INFO  Backup target: /var/backups/app
[2025-06-05 10:00:40] INFO  Received request: POST /api/v1/api-key
[2025-06-05 10:00:40] DEBUG Generated API key for user 'alice': sk_live_78tghasdhb7126asdad
[2025-06-05 10:00:41] INFO  Key creation successful
[2025-06-05 10:00:50] WARN  High memory usage detected
[2025-06-05 10:00:55] INFO  Received request: GET /api/v1/report
[2025-06-05 10:00:55] DEBUG Report parameters: {"date": "2025-06-01", "format": "csv"}
[2025-06-05 10:00:56] INFO  Report generated successfully
[2025-06-05 10:01:00] INFO  Received request: POST /api/v1/login
[2025-06-05 10:01:00] DEBUG Request payload: {"username": "bob", "password": "12345678"}
[2025-06-05 10:01:01] INFO  Auth failed for user bob
[2025-06-05 10:01:05] INFO  Received request: DELETE /api/v1/user
[2025-06-05 10:01:05] DEBUG Payload: {"user_id": "789", "confirmed": true}
[2025-06-05 10:01:06] INFO  User 789 deleted by admin
[2025-06-05 10:01:10] INFO  Received request: POST /api/v1/settings
[2025-06-05 10:01:10] DEBUG Payload: {"email": "test@example.com", "password": "qwerty123!"}
[2025-06-05 10:01:11] INFO  Settings updated
[2025-06-05 10:01:15] INFO  Gracefully shutting down
[2025-06-05 10:01:16] INFO  Service stopped
"""

# Search for 'username'
if "username" in log_data:
  print("Found possible username in log.")

# Search for 'password'
if "password" in log_data:
  print("Found possible password in log.")

# Search for 'api_key'
if "api_key" in log_data:
  print("Found possible API key in log.")
```

- Code di atas terlalu banyak perulangan. Jika kamu perlu menambah keyword lagi, misalkan "secret", "phone", "client", maka kamu perlu copas
- Bagaimana caranya agar code bisa lebih rapi dan fleksibel menerima berbagai key?

Solusi

- Untuk menghindari copy-paste code, gunakan function dengan keyword sebagai input.
- Tujuan: jika tambah keyword, cukup tambah parameter input, code tidak berubah. Lebih mudah dikelola
- perhatikan dan pahami code sebelum dan setelah

```
log_data = """
[2025-06-05 10:00:12] INFO  Starting service on port 8080
[2025-06-05 10:00:13] INFO  Connected to database at 127.0.0.1:5432
[2025-06-05 10:00:14] INFO  Environment: production
[2025-06-05 10:00:15] DEBUG Loaded configuration file: config.yml
[2025-06-05 10:00:16] INFO  Health check: OK
[2025-06-05 10:00:20] INFO  Received request: POST /api/v1/login
[2025-06-05 10:00:20] DEBUG Request payload: {"username": "admin", "password": "hunter2"}
[2025-06-05 10:00:21] INFO  Auth success for user admin
[2025-06-05 10:00:30] INFO  Received request: GET /api/v1/profile
[2025-06-05 10:00:30] DEBUG Auth token: Bearer abcdef1234567890abcdef
[2025-06-05 10:00:32] INFO  Request processed in 34ms
[2025-06-05 10:00:35] INFO  Scheduled job 'backup' started
[2025-06-05 10:00:35] INFO  Backup target: /var/backups/app
[2025-06-05 10:00:40] INFO  Received request: POST /api/v1/api-key
[2025-06-05 10:00:40] DEBUG Generated API key for user 'alice': sk_live_78tghasdhb7126asdad
[2025-06-05 10:00:41] INFO  Key creation successful
[2025-06-05 10:00:50] WARN  High memory usage detected
[2025-06-05 10:00:55] INFO  Received request: GET /api/v1/report
[2025-06-05 10:00:55] DEBUG Report parameters: {"date": "2025-06-01", "format": "csv"}
[2025-06-05 10:00:56] INFO  Report generated successfully
[2025-06-05 10:01:00] INFO  Received request: POST /api/v1/login
[2025-06-05 10:01:00] DEBUG Request payload: {"username": "bob", "password": "12345678"}
[2025-06-05 10:01:01] INFO  Auth failed for user bob
[2025-06-05 10:01:05] INFO  Received request: DELETE /api/v1/user
[2025-06-05 10:01:05] DEBUG Payload: {"user_id": "789", "confirmed": true}
[2025-06-05 10:01:06] INFO  User 789 deleted by admin
[2025-06-05 10:01:10] INFO  Received request: POST /api/v1/settings
[2025-06-05 10:01:10] DEBUG Payload: {"email": "test@example.com", "password": "qwerty123!"}
[2025-06-05 10:01:11] INFO  Settings updated
[2025-06-05 10:01:15] INFO  Gracefully shutting down
[2025-06-05 10:01:16] INFO  Service stopped
"""

def check_sensitive_data(log_text, search_words):
  for line in log_text.splitlines():
    for word in search_words:
      if word in line:
        print(f"[{word.upper()}] {line}")

# keywords jadi bisa dibuat terpisah. perlu keywords tambahan tinggal ubah line berikut
keywords = ["username", "password", "api_key", "secret"]

check_sensitive_data(log_data, keywords)
```

Referensi

-