JOMHACK
INSPIRE INNOVATION

PYTHON
BOOTCAMP

www.jomhack.com

FUNCTIONS

Functions:

- Reusable block of code that do specific task

```python
# Functions with parameters
def greet_person(name):
    print(f"Hello, {name}!")

greet_person("Alice")

# Functions with return values
def add_numbers(a, b):
    return a + b

result = add_numbers(5, 3)
print(result)  # 8

# Default parameters
def greet_with_title(name, title="Mr."):
    return f"Hello, {title} {name}!"

print(greet_with_title("Smith"))           # "Hello, Mr. Smith!"
print(greet_with_title("Johnson", "Dr."))  # "Hello, Dr. Johnson!"
```

FUNCTIONS

JOMHACK
INSPIRE INNOVATION

args :

- access by index: args[0]
- unpacking: func(*list)

kwargs :

- access by key: kwargs[‘key’]
- unpacking: func(**dict)

```python
# *args - variable number of arguments
def sum_all(*args):
    return sum(args)

print(sum_all(1, 2, 3, 4, 5))  # 15
```

```python
# **kwargs - keyword arguments
def print_info(**kwargs):
    for key, value in kwargs.items():
        print(f"{key}: {value}")

print_info(name="Alice", age=25, city="New York")
```

FUNCTIONS

JOMHACK
INSPIRE INNOVATION

args & kwargs :

```
36  # Combining *args and **kwargs
37  def flexible_function(*args, **kwargs):
38      print("Positional arguments:", args)
39      print("Keyword arguments:", kwargs)
40
41  flexible_function(1, 2, 3, name="Alice", age=25)
```

FUNCTIONS

JOMHACK
INSPIRE INNOVATION

lambda :

- anonymous function
- small function

```
43    # Lambda functions (anonymous functions)
44    square = lambda x: x**2
45    print(square(5))  # 25
46
47    add = lambda x, y: x + y
48    print(add(3, 4))  # 7
```

FUNCTIONS

Exercises :

1. Write a function that checks if a number is prime.
2. Build a temperature converter function. (Celsius to Fahrenheit)