הטכניון
היחידה ללימודי המשך
ולימודי חוץ עייש עזריאלי

Functions

Understanding Code Reusability and Modularity

TECHNION
Azrieli Continuing Education and External Studies Division

Functions

1. Defining functions
2. Working with parameters
3. Return
4. Scope

Functions

Definition:

A function is a reusable block of code that performs a specific task.

Len()

Functions

- **Why Use Functions?**
  - Avoid repeating code.
  - Modularity
  - Make code more organized and readable.
  - Easier debugging and testing.

Writing Functions

1. Start with the **def** keyword
2. **Name** the function
3. Add **():** after the name

```
def my_function():
    print('my function')
```

Naming Convention

- Variables:
  - All small letters.
  - Separate words with _.
  - Don’t start with a number.
- Functions:
  - All small letters.
  - Separate words with _.
  - Don’t start with a number.

Arguments

The difference between a parameter and an argument is:

A **parameter** is a variable inside the **function parenthesis**

An **argument** is data you add inside the **function call parenthesis**.

```
def function_name(parameters):
    # code block
    return value

function_name (arguments) # calling the function
```

Functions

Parameters

```
def function_name(parameters):
    # code block
    return value
```

Default value

```
def some_function(parameter= default_value):
    # do somthing
```

Return

```
def some_function(parameter):
    #do something
    return value
```

Input function

```
input ("enter something")
```

```
enter something ↑↓ for history. Search history with c-↑/c-↓
```

Writing Functions

Components:

def: Keyword to define a function.

function_name: Name you give the function.

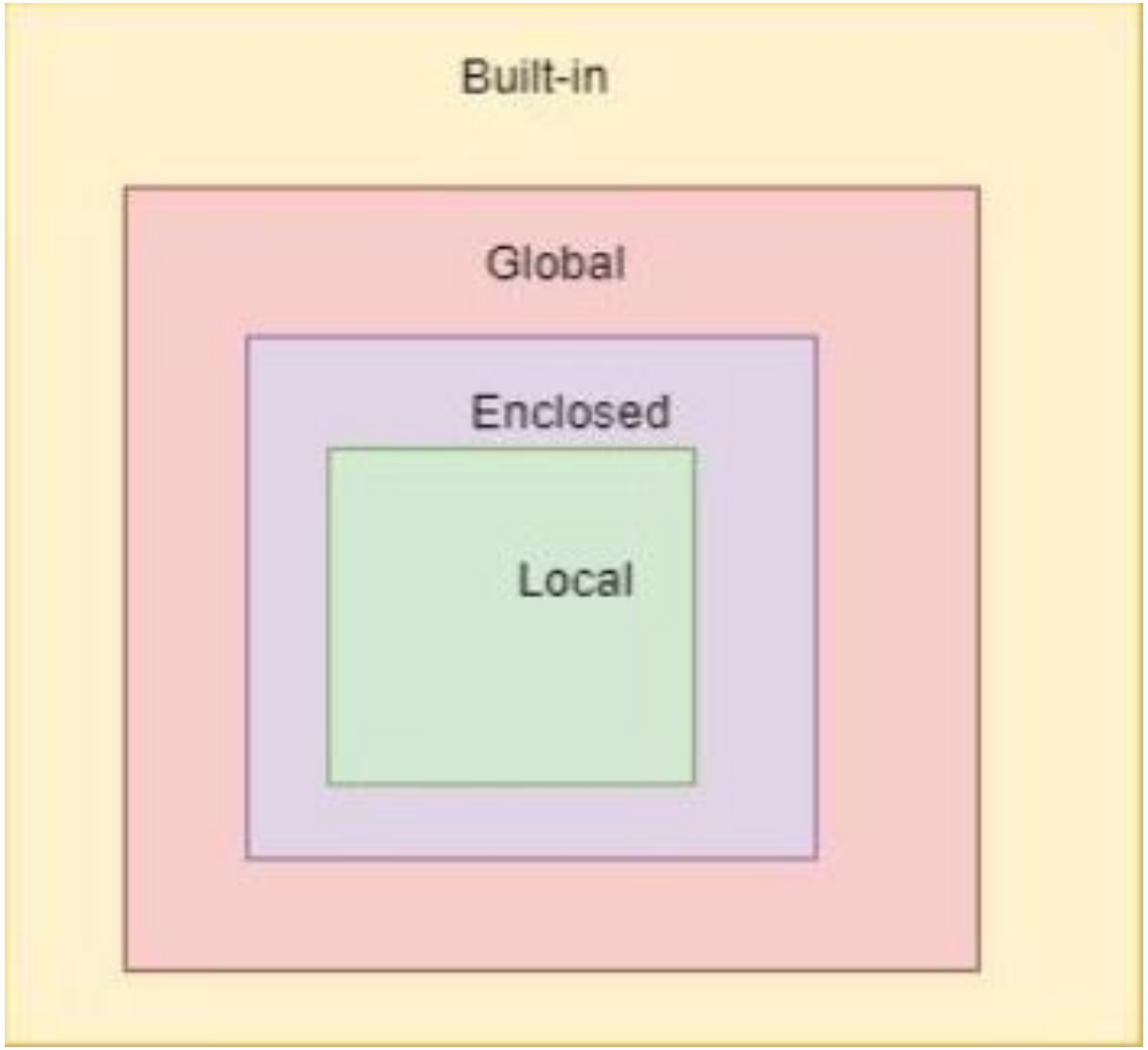
parameters: Optional input values.

return: (Optional) Outputs a value when the function finishes.

```
def function_name(parameters):
    # code block
    return value

function_name (arguments) # calling the function
```

SCOPE



- ✓ **Defining functions**
- ✓ **Working with parameters**
- ✓ **Return**
- ✓ **Scope**

Thank you