Féidearthachtaí as Cuimse
Infinite Possibilities

# Programming for Analytics

## Lecture 2: Control Structures and Functions

Bojan Božić
School of Computer Science
TU Dublin, Grangegorman

bojan.bozic@tudublin.ie

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

## Overview

- Understand how to use control structures in Python
- Write conditional logic using if, elif and else
- Use loops to repeat tasks (for and while)
- Define and call functions with arguments and return values

# Conditional Statements: if, elif, else

- Use `if` to run code based on a condition
- `elif` checks additional conditions if the first is false
- `else` runs if all above conditions fail
- Conditions use comparison operators: ==, !=, <, >, etc.

## Example conditional statement

```
age = 18
if age >= 18:
  print("You are an adult.")
else:
  print("You are a minor.")
```

# Logical Operators

- `and`: True if both conditions are true
- `or`: True if at least one condition is true
- `not`: Reverses the boolean value of the condition
- Useful for combining multiple conditions in `if` statements

## Loops: for and while

- `for` loop: iterate over a sequence (list, range, string).
- `while` loop: repeats while a condition is `true`.
- Use `break` to exit the loop early.
- Use `continue` to skip to next iteration.

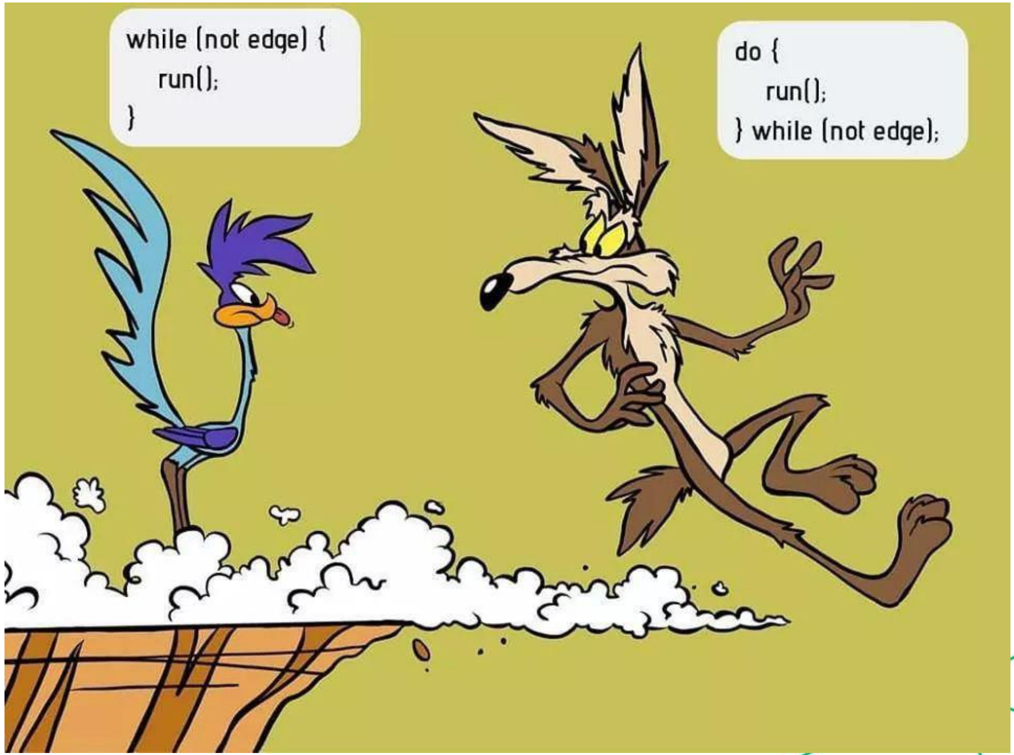
## Example: for loop

```
for I in range(5):
    print(i) # prints 0 to 4
```

## Example: while loop

```
count = 0
while count < 5:
    print(count)
    count += 1
```

# Make sure your loop ends when it should!

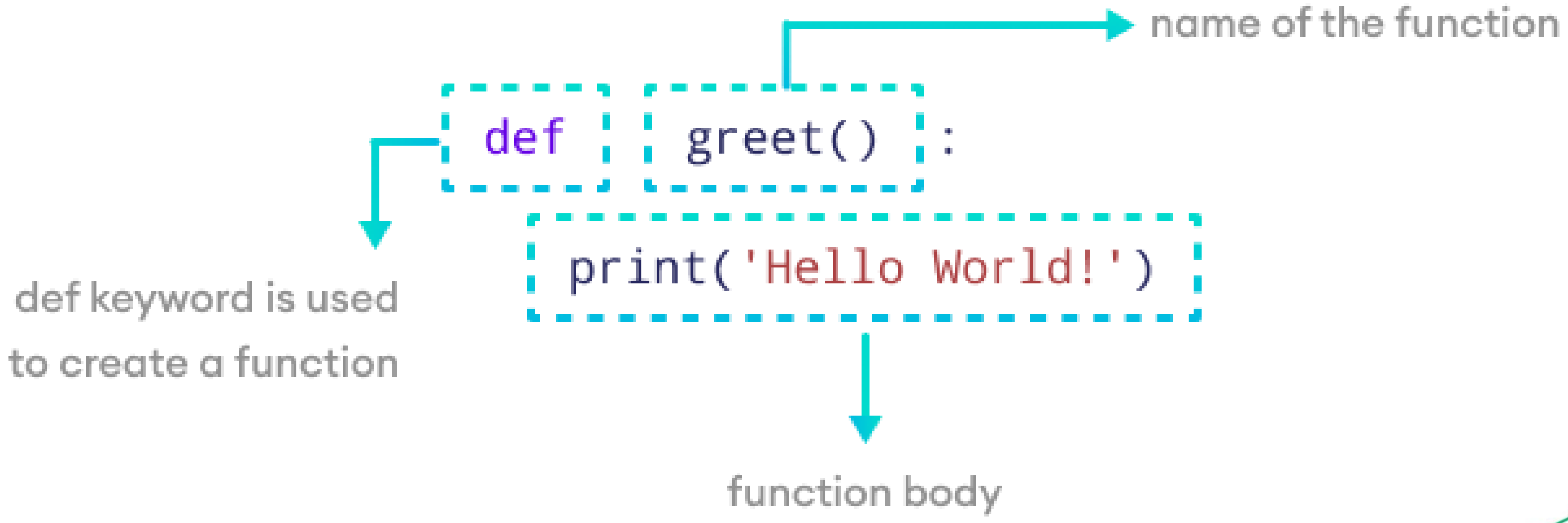

```
while (not edge) {
    run();
}
```

```
do {
    run();
} while (not edge);
```

## Functions: writing reusable code

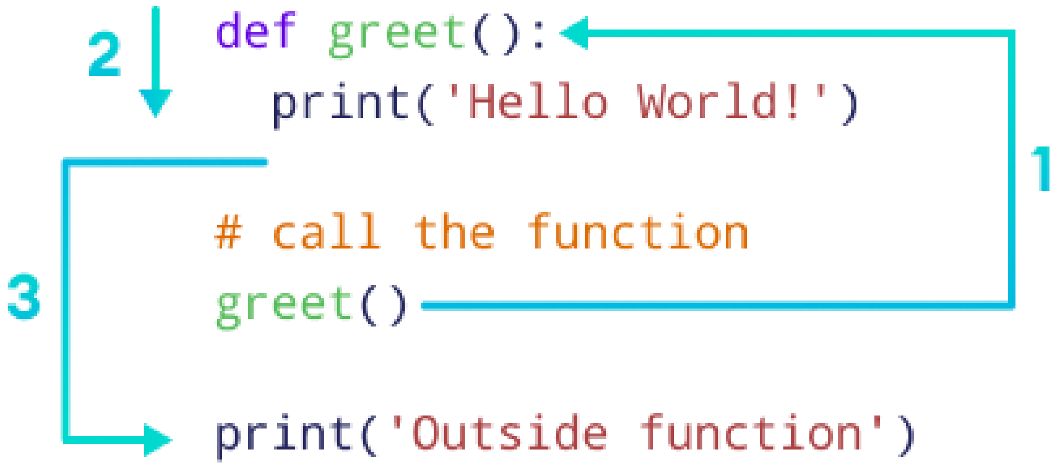
- Use def to define a function.
- Functions can take arguments and return a value.
- Helps avoid repetition and improves structure.

# Anatomy of a Function



```
def greet():
    print('Hello World!')
```

# Workflow of a Function


```
def greet():
    print('Hello World!')

# call the function
greet()

print('Outside function')
```

# Example: function

```
def greet(name):
    print(f“Hello, {name}!”)

greet(“Whiskeyjack”)
```

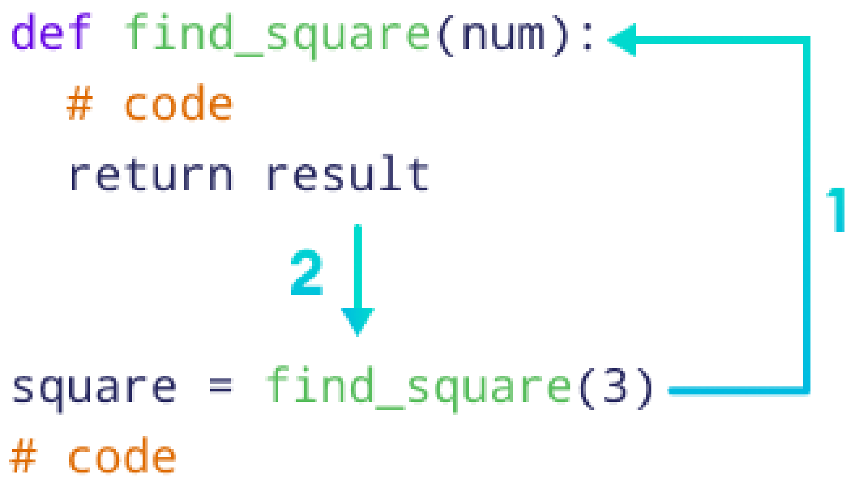
# Functions with Return Values

- Use return to send a result back to the caller

Example:

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

## Parameters and Return Value of a Function



```
def find_square(num):
  # code
  return result

square = find_square(3)
# code
```

# Function Parameters and Scope

- Parameters are variables passed to a function.
- Local variables exist only inside the function.
- Global variables can be accessed outside the function too.
- Avoid using too many global variables!

## Recap and Next Steps

- You learned how to use if/else, loops and functions.
- Practice writing programs with logic and repetition.
- Next week: data structures – lists, tuples, dictionaries, and sets.

## Nested Conditionals

- You can place if statements inside other if statements.
- Useful for checking multiple layers of conditions

## Example

```
if score >= 50:
    if score >= 80:
        print(“Excellent”)
    else:
        print(“Pass”)
```

## Common Mistakes with Conditionals

- Using `=` instead of `==` in comparisons.
- Incorrect indentation (Python relies on it!)
- Not covering all branches (missing `else`).
- Confusing `and` / `or` logic.

# Looping through Collections

- Use for loops to iterate through lists, tuples, and strings.

Example:

```
continents = [“Quon Tali”, “Genabackis”,
“Lether”]

for continent in continents:
    print(continent)
```

# Loop Control: break and continue

- `break`: stop the loop entirely
- `continue`: skip the rest of the current loop iteration
- Use sparingly to maintain code clarity.

## Practical Example: Summing Numbers

```
# sum all numbers from 1 to 100
total = 0
for i in range(1, 100)
    total += i
print(“Sum is ”, total)
```

## Functions with Default Parameters

- Functions can have default values for parameters.

```
def greet(name, greeting = “Hello”):
   print(f”{greeting}, {name}!”)

greet(“Toc the Younger”)
# -> “Hello, Toc the Younger!”

greet(”Anomander Rake”, “Hi”)
# -> “Hi Anomander Rake”
```

# Docstrings and Comments

- Use docstrings to describe what a function does
- Use # for inline comments

Example:

```
def square(n):
    “””Return the square of a number“””
    return n**2
```

# Activity: Voting Eligibility Checker

- Ask the user for their age and nationality.
- If they are over 18 and have Irish nationality, tell them they're eligible to vote.
- Otherwise tell them they're not eligible to vote.

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

# Activity: Password Validator

- Prompt the user to enter a password.
- Check if it has at least 8 characters and a number.
- Use `if`, `len()`, and `.isdigit()` or regex (if known).
- Print ”valid password” or appropriate message.

# Mini Quiz

- What's the output of: "`if 3 > 2 and 1 <2:`"?
- What does `break` do in a loop?
- What's the difference between a function and a loop?
- How would you loop through characters in a string?

Questions?