Script Programming 2

Flow Control

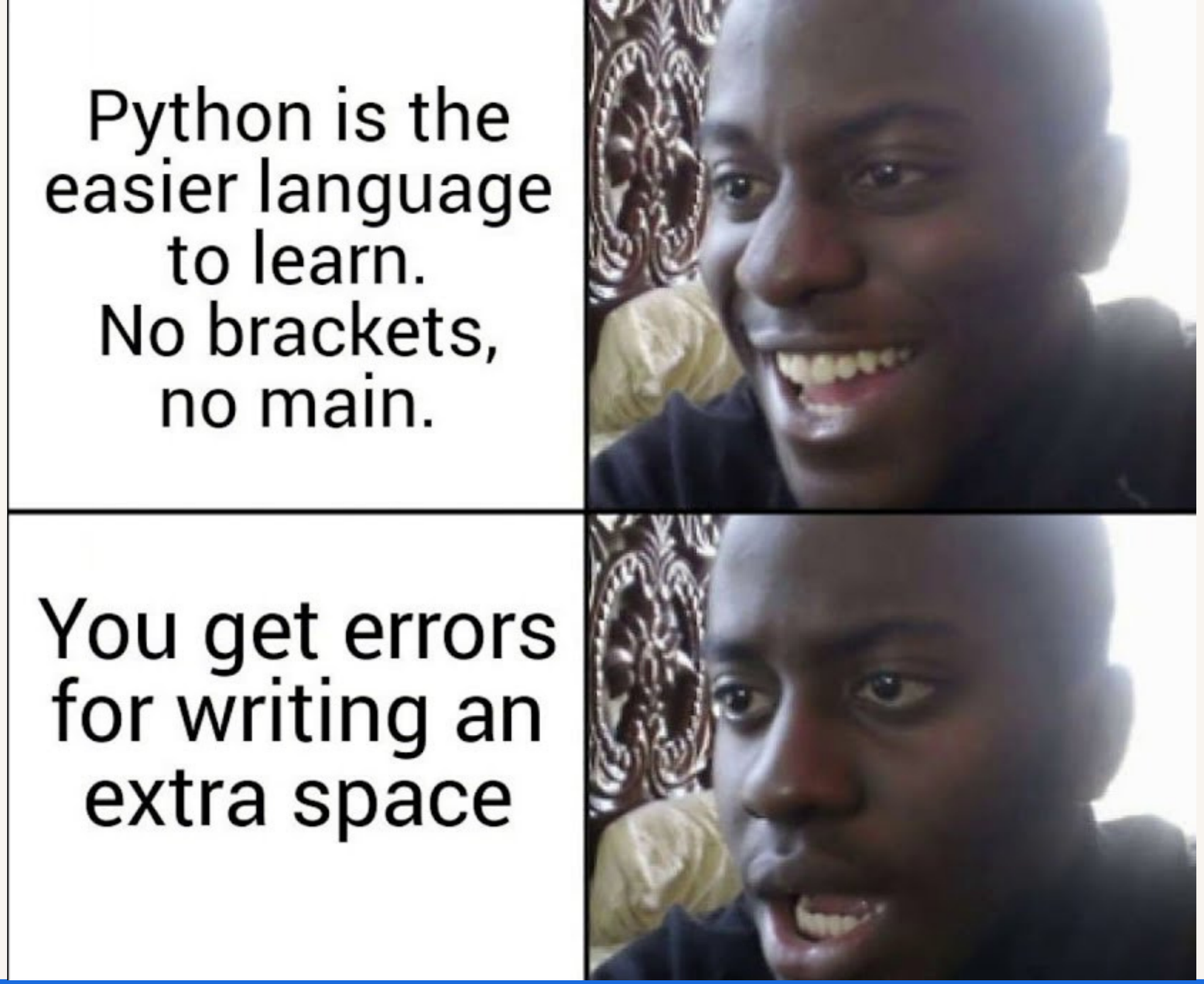
Merxhan Bajrami

2023 March 16

Week 2

Contents

- Boolean Values
- Comparison Operators,
- Boolean Operators
- Mixing Boolean and Comparison Operators
- Elements of Flow Control
- Program Execution
- Flow Control Statements
- Importing Modules
- Ending a Program Early with sys.exit().


Python is the easier language to learn. No brackets, no main.
You get errors for writing an extra space

Boolean values

Boolean values represent one of two possible states, True or False. They are often used in programming as a way of representing conditions, such as whether a statement is true or false. In Python, boolean values are represented by the keywords "True" and "False" (note the capitalization).

E.g

```
x = True
y = False

print(x)  # output: True
print(y)  # output: False
```

Comparison Operators

Comparison operators are used to compare two values and return a boolean value (True or False) based on whether the comparison is true or false. Some common comparison operators in Python include "==" (equal to), "!=" (not equal to), ">" (greater than), "<" (less than), ">=" (greater than or equal to), and "<=" (less than or equal to).

E.g

```
x = 10
y = 5
print(x == y)  # output: False
print(x != y)  # output: True
print(x > y)  # output: True
print(x < y)  # output: False
print(x >= y)  # output: True
print(x <= y)  # output: False
```

Boolean Operators

Boolean operators are used to combine boolean values and return a new boolean value. Some common boolean operators in Python include "and" (returns True if both operands are True), "or" (returns True if either operand is True), and "not" (returns the opposite boolean value of the operand).

```
x = True
y = False

print(x and y)  # output: False
print(x or y)  # output: True
print(not x)  # output: False
```

Mixing Boolean and Comparison Operators

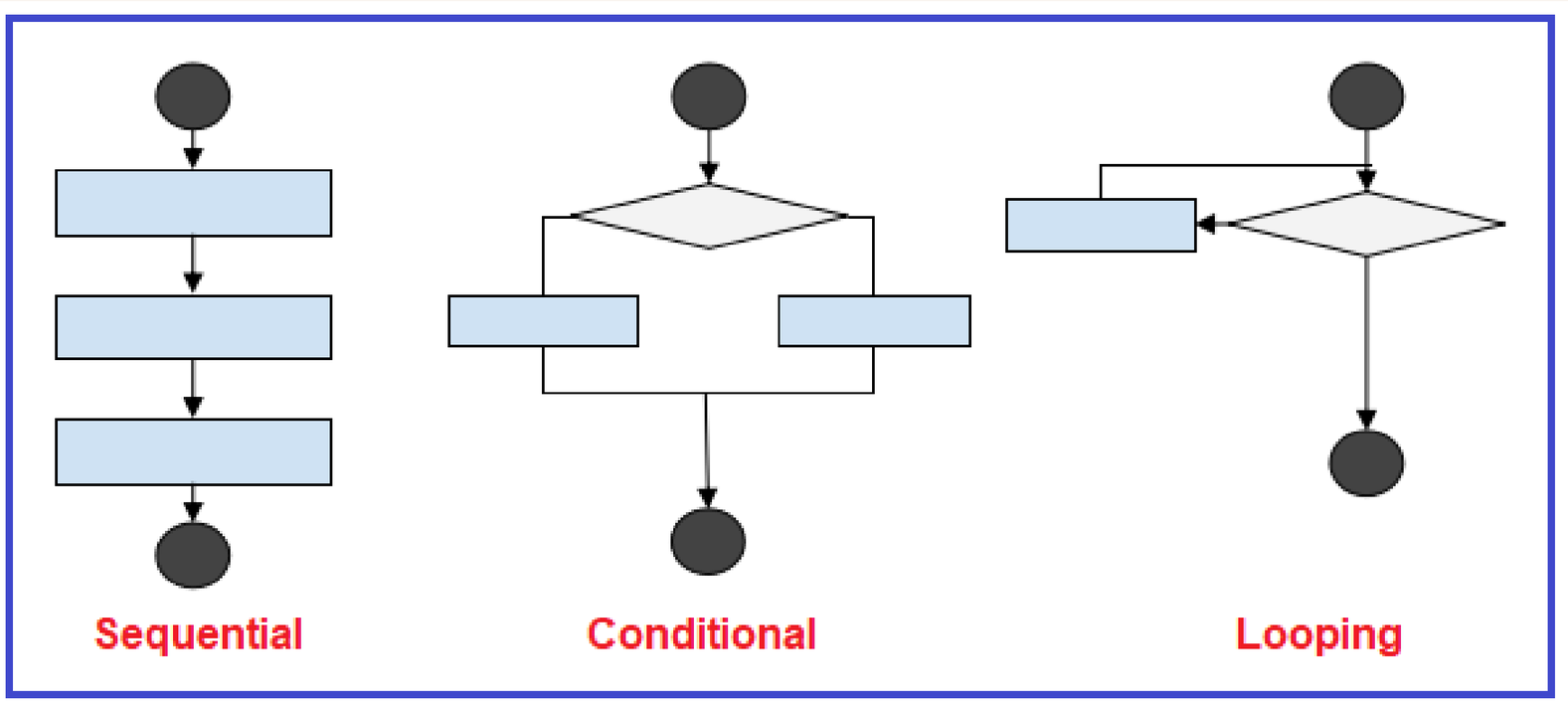
You can mix boolean and comparison operators to create more complex conditions. For example, you might use "and" to combine two comparison operations to check if a value is within a certain range.

```
x = 10

print(x > 5 and x < 15)  # output: True
print(x > 15 or x < 5)  # output: False
```

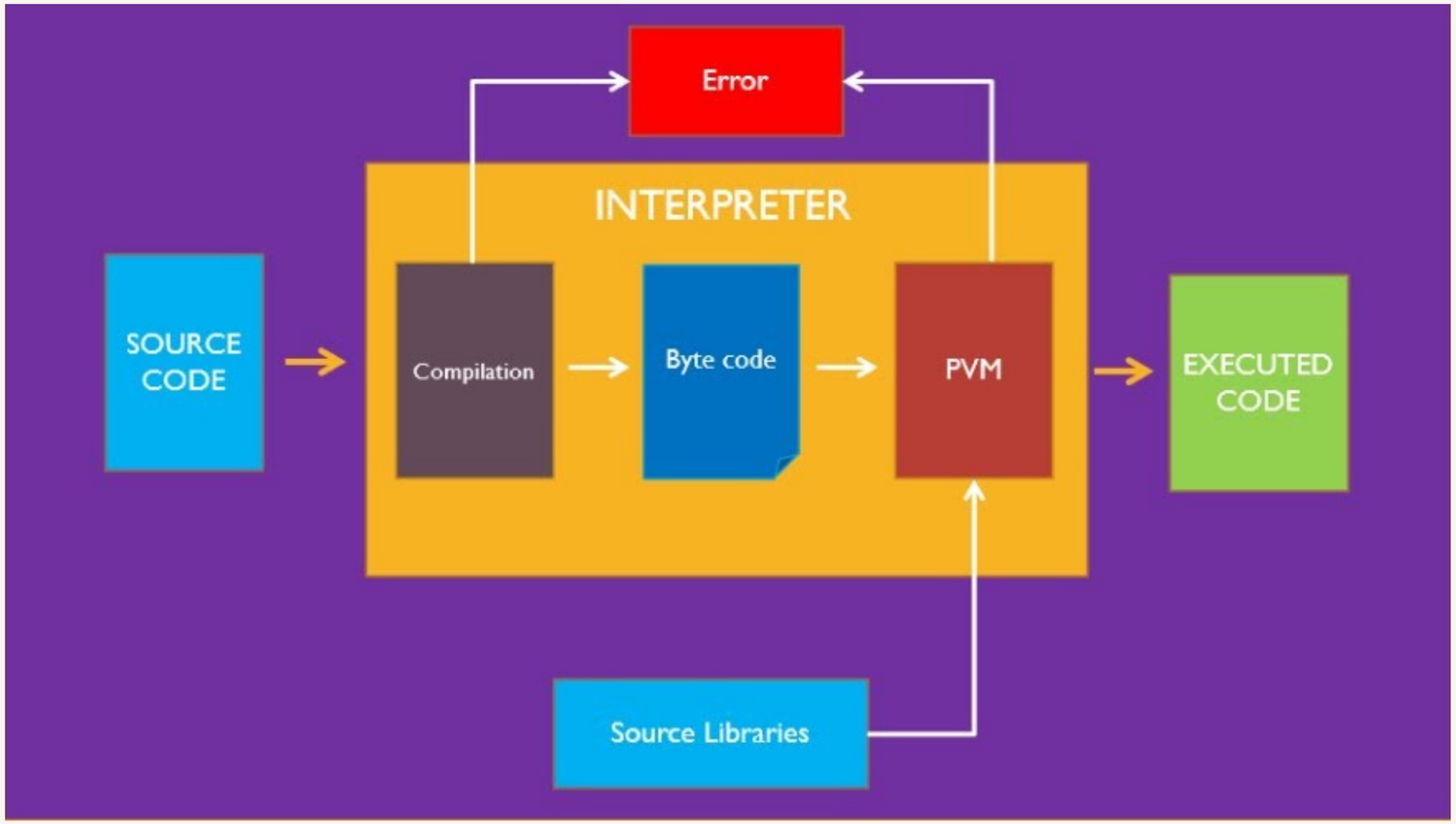
Elements of Flow Control

Flow control refers to the use of statements and conditions to control the execution of a program. Some common elements of flow control in Python include conditional statements (if/else), loops (for/while), and function calls.

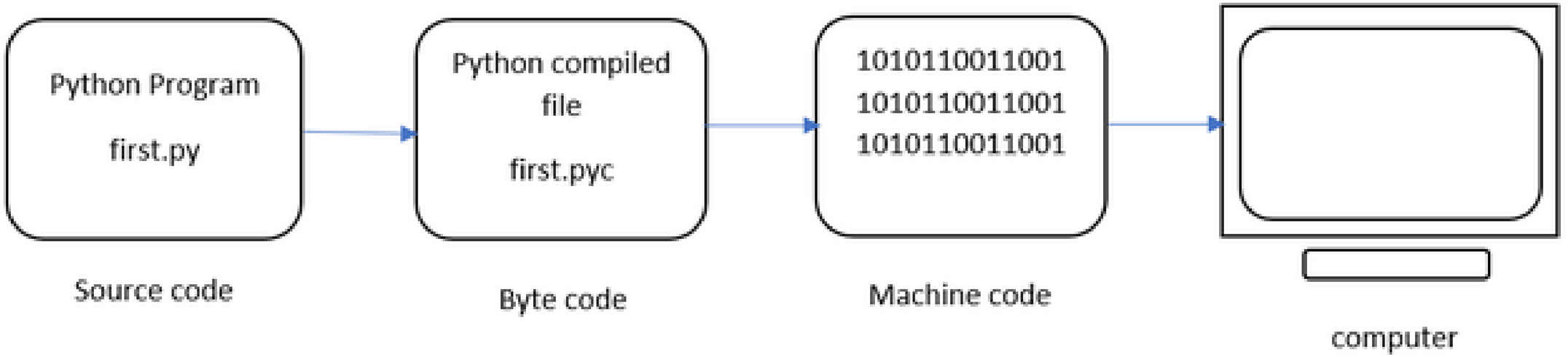


Program Execution

When a Python program is executed, the interpreter reads the code line by line and executes each statement in order. However, you can use flow control statements to alter the order of execution or skip certain statements altogether.



Program Execution



Exercices

Flow Control Statements

Flow control statements are used to control the flow of a program. Some common flow control statements in Python include "if/else" (used for conditional execution), "for/while" (used for looping), and "break/continue" (used for altering the flow of a loop).

```
x = 10
if x > 5:
    print("x is greater than 5")
for i in range(5):
    if i == 3:
        continue
    print(i)
```

Importing Modules

Python modules are pre-written code libraries that can be imported into your program to provide additional functionality. You can use the "import" keyword to import a module, and then use its functions and classes in your program.

```
import math
x = 16
sqrt_x = math.sqrt(x)
print(sqrt_x)
```

```
from math import sqrt
x = 16
sqrt_x = sqrt(x)
print(sqrt_x)
```

```
import math as m
x = 16
sqrt_x = m.sqrt(x)
print(sqrt_x)
```

```
from math import sqrt, sin, cos
```

Flow control

- Problem1: Write a program that asks the user for their age and determines if they are old enough to vote (age 18 or over).
- Problem2: Write a program that generates a random number between 1 and 100 and asks the user to guess the number. The program should give hints (higher/lower) until the user guesses correctly.

Boolean values

- Problem1: Write a program that asks the user for two numbers and checks if they are equal.
- Problem2: Write a program that asks the user for three numbers and checks if they form a Pythagorean triple (i.e. a^2 + b^2 = c^2).

Comparison operators

- Problem1: Write a program that asks the user for a number and checks if it is positive or negative.
- Problem2: Write a program that asks the user for two dates (in the format YYYY-MM-DD) and determines which date comes first.

Boolean operators

- Problem1: Write a program that asks the user for their age and checks if they are between 18 and 65 years old.
- Problem2: Write a program that asks the user for three numbers and checks if they are all odd or all even.

Mixing Boolean and Comparison operators

- Problem1: Write a program that asks the user for a password and checks if it meets certain criteria (e.g. at least 8 characters long, contains at least one uppercase letter and one number).
- Problem2: Write a program that asks the user for a number and checks if it is a prime number.

Elements of Flow Control

- Problem1: Write a program that asks the user for a number and prints all the numbers from 1 to that number.
- Problem2: Write a program that asks the user for a string and prints all the permutations of that string.

Program Execution

- Problem1: Write a program that calculates the sum of all the numbers from 1 to 100 and prints the result.
- Problem2: Write a program that generates a list of all the prime numbers between 1 and 1000 and prints the list.

Flow control statements

- Problem1: Write a program that asks the user for a number and prints whether it is even or odd using a if...else statement.
- Problem2: Write a program that generates a list of Fibonacci numbers (up to a certain limit) using a while loop.

Importing modules

- Problem1: Write a program that uses the random module to generate a random number between 1 and 10.
- Problem2: Write a program that uses the numpy module to generate a 2D array of random numbers and calculates the dot product of the array with itself.

Script Programming 2

Questions?

Merxhan Bajrami

2023 March 16

Week 2