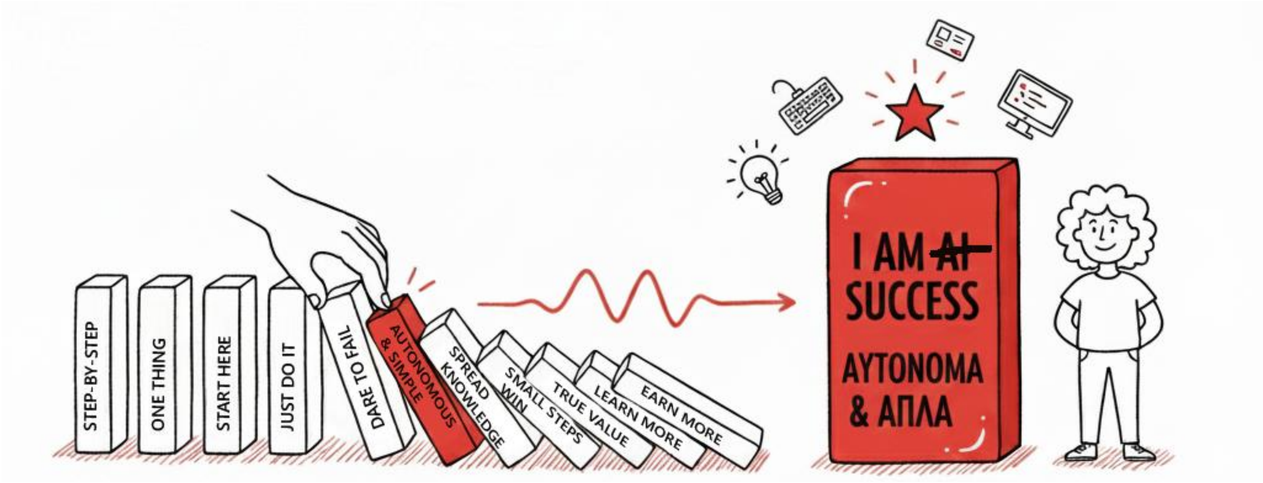
Computing using Python Quiz



Quiz Statistics:

Total Questions: 55+15

Section 1 (Questions 1-35): Multiple Choice Questions

- Part A (Questions 1-10): Basic Concepts
- Part B (Questions 11-17): Algorithms & Representation
- Part C (Questions 18-25): Variables & Data Types
- Part D (Questions 26-35): Programming Concepts

SECTION 2: Code Writing & Error Correction (20 questions)

- Part A (Questions 36-40): Basic Python Commands
- Part B (Questions 41-45): Variables & Calculations
- Part C (Questions 46-50): Comments & Data Types
- Part D (Questions 51-55): Input/Output & Logic

SECTION 3: (15 questions) Procedures, Functions, Lists & Time Library

QUICK NOTES/ TIPS:

- Why no space?
- Why no Caps? - print is lowercase, not Print
- Strings need QUOTES, numbers don't
- Variable names: NO spaces, NO starting with numbers
- Always Remember parentheses () when calling functions
- Comments use # not //
- Assignment = vs Comparison ==
- Always Remember the COLON : after if, def, for, while
- Indentation MATTERS in Python
- Lists need COMMAS between items
- Time library: strftime() needs BOTH format AND gmtime()
- BONUS MEMORY TRICKS

Recommended Time: 55 minutes

SECTION 1: Multiple Choice Questions (35 questions)

Instructions: Choose the best answer for each question.

Part A: Basic Concepts (Questions 1-10)

1. What is Computing?

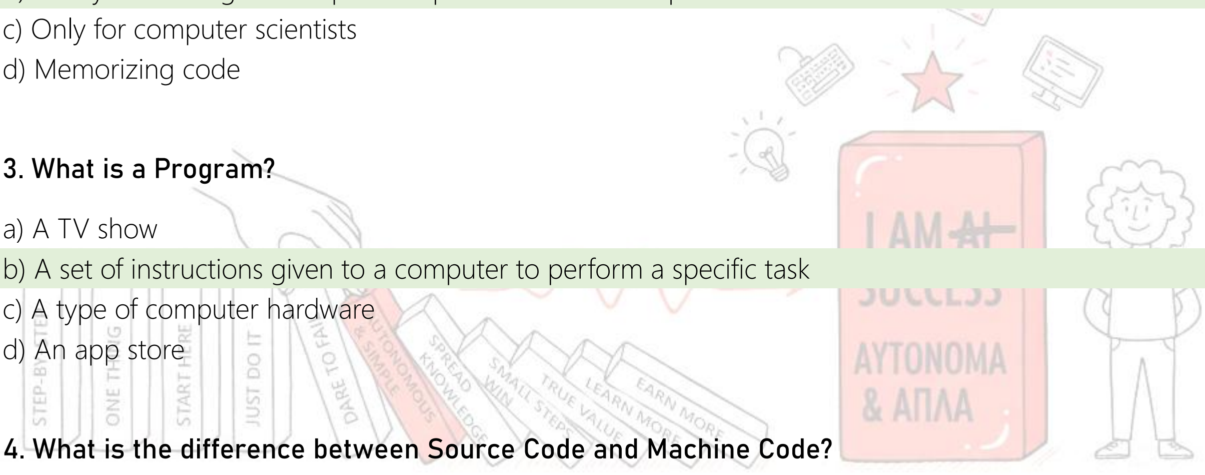
a) Only using computers for games
b) The science of how computers process information and solve problems
c) Learning to type fast
d) Using social media

2. What is Computational Thinking?

a) Thinking about computers all day
b) A way of thinking that helps solve problems like a computer would
c) Only for computer scientists
d) Memorizing code

3. What is a Program?

a) A TV show
b) A set of instructions given to a computer to perform a specific task
c) A type of computer hardware
d) An app store



4. What is the difference between Source Code and Machine Code?

a) There is no difference
b) Source code is written by humans; machine code is in 0s and 1s understood by computers
c) Machine code is written in Python
d) Source code is faster

5. Which is an example of a Formal Language?

a) Greek
b) English
c) Python programming language
d) Casual conversation

6. What is Decomposition in computational thinking?

a) Breaking a large problem into smaller, manageable parts
b) Deleting old files

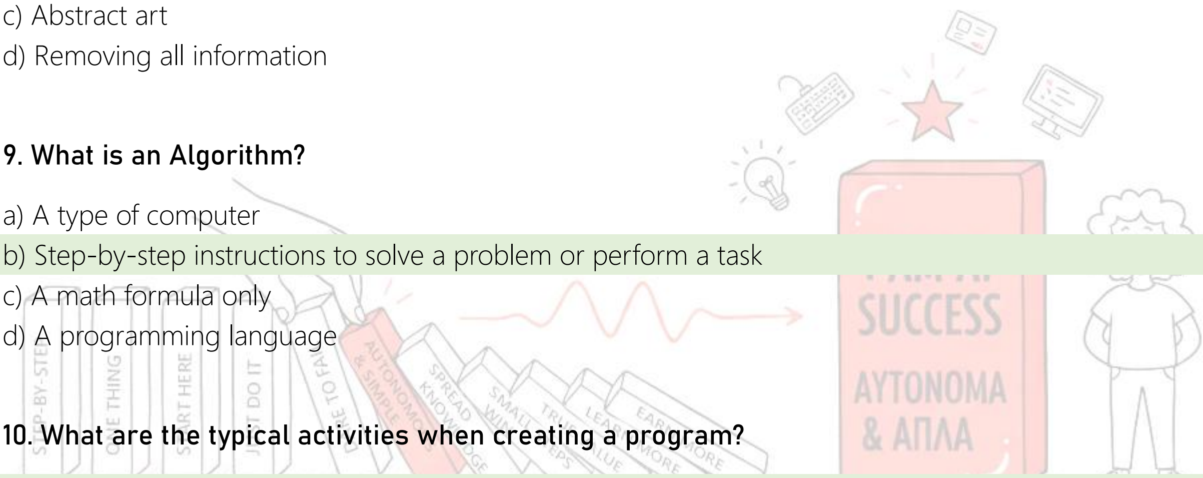
c) Making problems more complicated
d) Ignoring the problem

7. What is Pattern Recognition?

a) Recognizing shapes only
b) Identifying similarities or repeating elements in problems or data
c) Learning patterns by heart
d) Drawing patterns

8. What is Abstraction?

a) Making things more complicated
b) Focusing only on important information and ignoring unnecessary details
c) Abstract art
d) Removing all information



9. What is an Algorithm?

a) A type of computer
b) Step-by-step instructions to solve a problem or perform a task
c) A math formula only
d) A programming language

10. What are the typical activities when creating a program?

a) Analysis, design, implementation, testing, improvement
b) Only writing code
c) Just testing
d) Playing games

Part B: Algorithms & Representation (Questions 11-17)

11. What is a Flow Chart?

a) A chart showing water flow
b) A visual map showing the steps of an algorithm or program
c) A type of graph
d) A calendar

12. Which symbol represents "Start/Stop" in a flow chart?

a) Rectangle
b) Diamond

c) Oval/Rounded rectangle
d) Arrow

13. Which symbol represents a "Decision" in a flow chart?

a) Rectangle
b) Diamond
c) Oval
d) Circle

14. What is Pseudocode?

a) Fake code that doesn't work
b) A description of a program in simple words that looks like code but isn't in a programming language
c) Code with errors
d) Machine code

15. What does "sequence" mean in programming?

a) Random order
b) Steps executed one after another in order
c) Skipping steps
d) Repeating steps

16. Which flowchart symbol represents "Input/Output"?

a) Diamond
b) Parallelogram
c) Rectangle
d) Circle

17. What connects different parts of a flowchart?

a) Lines
b) Arrows
c) Dots
d) Colors

Part C: Variables & Data Types (Questions 18-25)

18. What is a Variable?

a) A box in computer memory that stores information that can change
b) Something that never changes
c) A type of computer
d) An error

19. Which data type would you use to store someone's name?

a) Integer
b) Float
c) String
d) Boolean

20. Which data type is used for whole numbers?

a) String
b) Float
c) Integer
d) Character

21. Which data type is used for decimal numbers?

a) Integer
b) Float
c) String
d) Boolean

22. What values can a Boolean data type have?

a) Any number
b) True or False only
c) Any text
d) 0 to 100

23. Which is a good variable name?

a) x
b) varStudentAge
c) 123name
d) my variable

24. What is an Array or List?

a) A single value

b) A collection that stores multiple values together

c) A type of loop

d) An error message

25. In Python, which symbol is used to assign a value to a variable?

a) ==

b) =

c) :

d) ->

Part D: Programming Concepts (Questions 26-35)

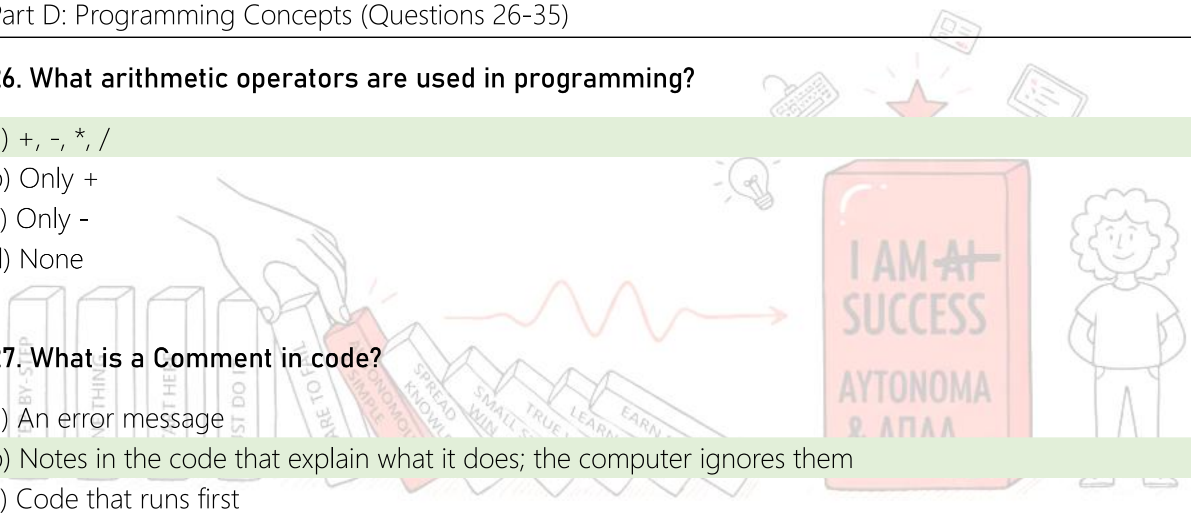
26. What arithmetic operators are used in programming?

a) +, -, *, /

b) Only +

c) Only -

d) None



27. What is a Comment in code?

a) An error message

b) Notes in the code that explain what it does; the computer ignores them

c) Code that runs first

d) A type of variable

28. In Python, what symbol starts a comment?

a) //

b) #

c) /*

d) --

29. What is a Logic Test?

a) A test at school

b) A check in a program to make decisions (e.g., if something is true or false)

c) Testing computer hardware

d) A math quiz

30. Which Boolean expression checks if two values are equal?

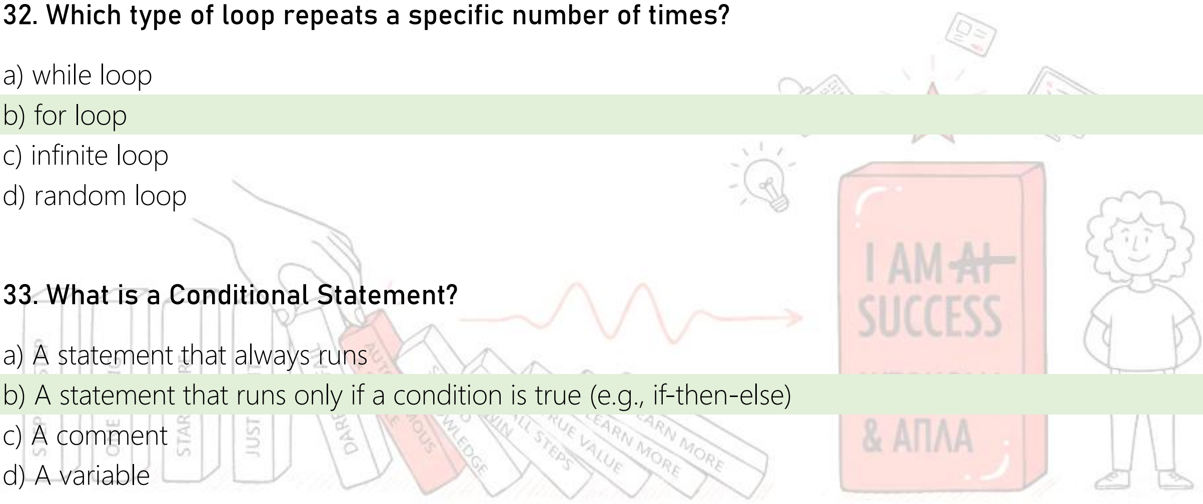
a) =
b) ==
c) !=
d) ===

31. What is a Loop?

a) A type of variable
b) A structure that repeats a set of instructions until a condition is met
c) An error
d) A comment

32. Which type of loop repeats a specific number of times?

a) while loop
b) for loop
c) infinite loop
d) random loop



33. What is a Conditional Statement?

a) A statement that always runs
b) A statement that runs only if a condition is true (e.g., if-then-else)
c) A comment
d) A variable

34. What is a Function?

a) A piece of code that performs a specific task and can be reused
b) A type of variable
c) An error message
d) A comment

35. What are the two main types of errors in programming?

a) Big and small errors
b) Syntax errors and logic errors
c) Red and blue errors
d) Fast and slow errors

SECTION 2: Code Writing & Error Correction (20 questions)

Instructions: Write one line of Python code OR correct the error in the given code.

Part A: Basic Python Commands (Questions 36-40)

36. Write a Python statement to display "Hello, World!" on the screen.

```
print("Hello, World!")
```

37. Write a Python statement to display "Welcome to Nicosia!" on the screen.

```
print("Welcome to Nicosia!")
```

38. Correct the error in this code:

print(Hello, students!)	print("Hello, students!")

39. Correct the error in this code:

Print("Good morning")	print("Good morning")

40. Write a Python statement to calculate and display the result of 15 + 27.

```
print(15 + 27)
```

Part B: Variables & Calculations (Questions 41-45)

41. Write a Python statement to create a variable called age and assign it the value 15.

```
VarAge = 15
```

42. Write a Python statement to create a variable price with value 9.99 (a decimal number).

```
price = 9.99
```

43. Correct the error in this code:

student name = "Sina Rokina"	StudentName = " Sina Rokina "
	OR
	StudentName = str(" Sina Rokina ")

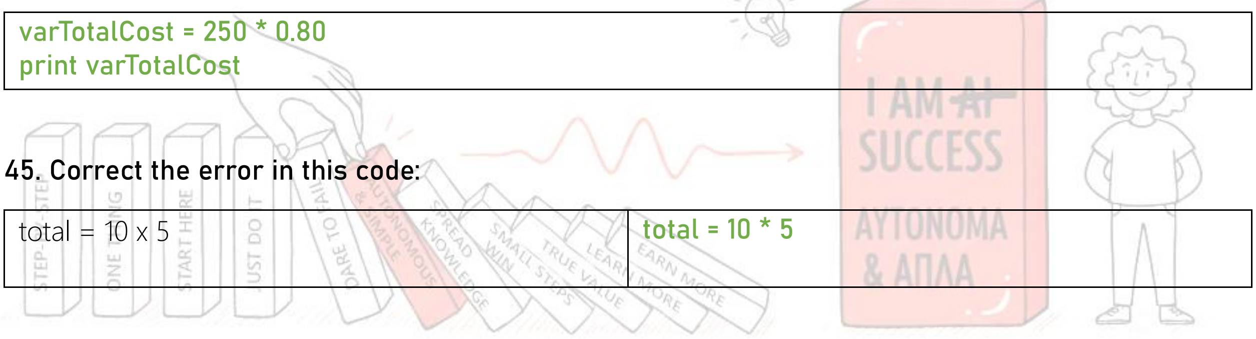
Both works, since " Sina Rokina " is already a string, but for the exam use str().

44. Write Python code to calculate the total cost: 250 kilos × 0.80 euros per kilo.

```
varTotalCost = 250 * 0.80
print varTotalCost
```

45. Correct the error in this code:

total = 10 x 5	total = 10 * 5



Part C: Comments & Data Types (Questions 46-50)

46. Write a comment in Python that says "This is my first program".

```
# This is my first program
```

47. Correct the error in this code:

// This is a comment in Python	# This is a comment in Python

48. Write a Python statement to create a variable is_student with a Boolean value of True.

```
is_student = True
```

49. Write a Python statement to create a variable name and assign it your name as a string.

```
name = "YourName"
```

50. Correct the error in this code:

number = "25"
result = number + 10
	number = 25
	or
	result = int(number) + 10

Part D: Input/Output & Logic (Questions 51-55)

51. Write a Python statement to get input from the user and store it in a variable called username.

```
username = input("Enter username: ")
```

52. Correct the error in this code:

age = input("Enter your age:") print(You are, age, years old)	print("You are", age, "years old")

53. Write a Python if statement that checks if a variable score is greater than 50.

```
if score > 50:
```

54. Correct the error in this code:

if age = 18:
    print("Adult")
	if age == 18:
	   print(“Adult”)

55. Write a Python for loop that prints numbers from 1 to 5.

```
for i in range(1, 6):
```

SECTION 3: Procedures, Functions, Lists & Time Library (15 questions)

Instructions: This section includes multiple choice, code writing, and code correction questions.

Part A: Multiple Choice (Questions 56-60)

56. What is the difference between a Procedure and a Function in Python?

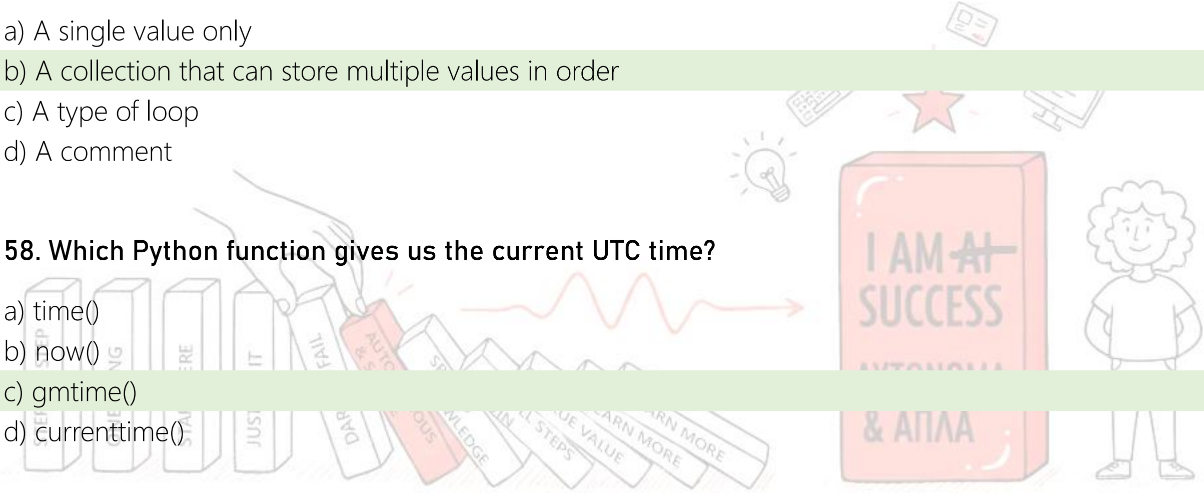
a) There is no difference
[✓ b) A function can return a value; a procedure just executes code]
c) Procedures are faster
d) Functions cannot be reused

57. What is a List in Python?

a) A single value only
[✓ b) A collection that can store multiple values in order]
c) A type of loop
d) A comment

58. Which Python function gives us the current UTC time?

a) time()
b) now()
[✓ c) gmtime()]
d) currenttime()



59. What does the format code %m represent in strftime()?

a) Minutes
[✓ b) Month]
c) Machine time
d) Milliseconds

60. How do you access the first element in a Python list called fruits?

[✓ a) fruits[0]]
b) fruits[1]
c) fruits.first
d) fruits(1)

Part B: Code Writing (Questions 61-68)

61. Write a Python statement to create a list called colors with three items: "red", "blue", "green".

```
varColors = ["red", "blue", "green"]
```

62. Write a Python statement to add the item "yellow" to the end of the list colors.

```
colors.append("yellow")
```

63. Write a Python function definition called greet that prints "Hello!". (Just the function definition line with def)

```
def greet():
```

64. Write a Python statement that imports strftime and gmtime from the time library.

```
from time import strftime, gmtime
```

65. Write Python code to print today's date in the format DD/MM/YYYY using the time library.

```
print(strftime("%d/%m/%Y", gmtime()))
```

66. Write Python code to print the current year only using the time library.

```
print(strftime("%Y", gmtime()))
```

67. Write a Python function called add_numbers that takes two parameters a and b and returns their sum. (Write the complete function)

```
def add_numbers(a, b):
    varSum = a + b
return varSum
```

68. Write Python code to print the current time in the format HH:MM:SS using the time library.

```
print(strftime("%H:%M:%S", gmtime()))
```

Part C: Code Correction (Questions 69-70)

69. Correct the error in this list creation:

VarMyList = ["apple" "banana" "orange"]	**VarMyList = ["apple", "banana", "orange"]**

(Missing commas between list items)

70. Correct the error in this function definition:

def CalculateTotal(price, quantity)
    return price * quantity
	**def CalculateTotal(price, quantity):
	    return price * quantity**

(Missing commas between list items)

71. Correct the error in this time formatting code:

from time import strftime, gmtime print(strftime("%d-%m-%Y"))	**print(strftime("%d-%m-%Y", gmtime()))**

(Missing gmtime() as second parameter)

72. Correct the error in accessing a list element:

numbers = [10, 20, 30, 40]
print(numbers(2))
	print(numbers[2])

(Should use square brackets [ ] not parentheses ( ))

73. Correct the error in this function call:

def ShowMessage():
    print("Welcome!")
 	
ShowMessage
	ShowMessage()

(Missing parentheses to actually call the function)

74. Correct the error in this time format code:

print(strftime("%H-%M-%S", gmtime))	**print(strftime("%H:%M:%S", gmtime()))**

(Missing parentheses after gmtime)

75. Correct the error in this function that should return a value:

def multiply(x, y):
    result = x * y
	def multiply(x, y): result = x * y return result
	OR
	def multiply(x, y): return x * y

(Missing return statement)

10 QUICK TIPS TO AVOID COMMON PYTHON MISTAKES

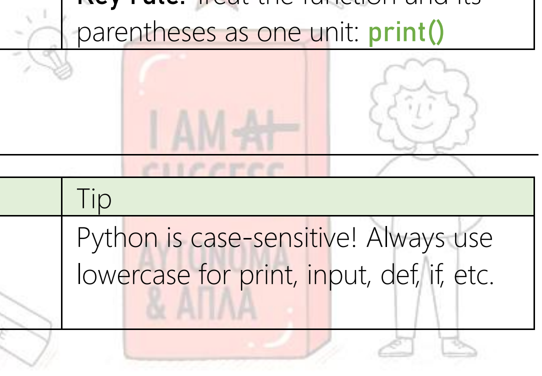
Why no space?

According to PEP 8 (Python's official style guide), you should not put a space between a function name and the opening parenthesis. This is the standard convention that all Python programmers follow.

This applies to ALL functions:

Correct	Tip

print(xxx)
input(xxx)
strftime(xxx, xxx)
	Write print() with no space — it's the professional, correct way!
	Key rule: Treat the function and its parentheses as one unit: print()



Why no Caps? - print is lowercase, not Print

	Tip
❌ Print("Hello") # WRONG - Capital P ✅ print("Hello") # CORRECT - lowercase p	Python is case-sensitive! Always use lowercase for print, input, def, if, etc.

Strings need QUOTES, numbers don't

	Tip
❌ name = Nicole # WRONG - no quotes ✅ name = "Nicole" # CORRECT - strings need quotes ❌ age = "15" # WRONG if you want to do math ✅ age = 15 # CORRECT - numbers without quotes	If it's text → use quotes. If it's a number for calculations → no quotes.

Variable names: NO spaces, NO starting with numbers

	Tip
❌ student name = "Nicole" # WRONG - space ❌ 1student = "Nicole" # WRONG - starts with number ✅ student_name = " Nicole" # CORRECT - use underscore ✅ studentName = " Nicole" # CORRECT - camelCase	Use underscores (_) instead of spaces!

Remember parentheses () when calling functions

	Tip
❌ show_message # WRONG - just the name ✅ show_message() # CORRECT - needs () to run	Functions are like doorbells—you must press () to make them work!

Comments use # not //

	Tip
❌ // This is a comment # WRONG - that's JavaScript/C++ ✅ # This is a comment # CORRECT - Python uses	In Python, the hashtag # is your friend for comments!

Assignment = vs Comparison ==

	Tip
❌ if age = 18: # WRONG - single = assigns ✅ if age == 18: # CORRECT - double == compares	One equals (=)** → "store this value" Two equals (==)** → "are these equal?"

Always Remember the COLON : after if, def, for, while

	Tip
❌ def greet() # WRONG - missing colon ✅ def greet(): # CORRECT ❌ if age > 18 # WRONG ✅ if age > 18: # CORRECT	Think of the colon as saying "here's what comes next!"

Indentation MATTERS in Python

	Tip

❌ def greet():
print("Hello") # WRONG - no indent
✅ def greet():
    print("Hello") # CORRECT - indented
	After def, if, for, while → press TAB or 4 spaces for the next line!

Lists need COMMAS between items

	Tip
❌ colours = ["red" "blue" "green"] # WRONG - no commas ✅ colours = ["red", "blue", "green"] # CORRECT	Every item in a list needs a comma after it (except the last one)!

❌ numbers = [1 2 3 4] # WRONG ✅ numbers = [1, 2, 3, 4] # CORRECT	

Time library: strftime() needs BOTH format AND gmtime()

	Tip
❌ print(strftime("%d/%m/%Y")) # WRONG - missing gmtime() ✅ print(strftime("%d/%m/%Y", gmtime())) # CORRECT ❌ print(strftime("%H:%M:%S", gmtime)) # WRONG - missing () on gmtime ✅ print(strftime("%H:%M:%S", gmtime())) # CORRECT	Always use the pair: strftime("format", gmtime())

BONUS MEMORY TRICKS:

Multiplication in Python	Multiplication in Math
5 * 3 *(use asterisk *)*	5 × 3

List Index starts at 0	Tip
fruits = ["apple", "banana", "orange"] fruits[0] # → "apple" (FIRST item) fruits[1] # → "banana" (SECOND item)	Think of it like floors in some European buildings—the ground floor is 0, first floor is 1! In Python lists, counting always starts at position 0, not 1. So if you want the FIRST item, ask for [0]!