Python Programming

Level 1 (Easy)

https://cedricf6.github.io/python-course/

Python Programming - Level 1 (Easy) - Questions

Section A: Output Statements & Variables

1. What is the output of the following code snippet?

```
print("Hello")
print("World")
```

Answer:

---

2. What is the output of the following code snippet that uses an f-string?

```
name = "Kim"
score = 95
print(f"{name} scored {score}%")
```

Answer:

---

3. Define the term variable in the context of programming.

Answer:

---

4. Identify the data type of the variable `is_raining` in the statement below:

```
is_raining = True
```

Answer:

---

5. What does the `input()` function do in a Python program?

Answer:

---

6. What is the output of the following code snippet?

```
num = input("Enter a number: ") # User enters '10'
print(num * 2)
```

Answer:

---

7. The following code causes an error. Why?

```
total = "5" + 3
```

Answer:

Section B: Operators

8. What is the value of `result` after the following operations?a = 10

```
b = 3
result = a % b
```

Answer:

9. What is the value of `x` after the following operation is executed?

```
x = 15
x //= 4
```

Answer:

10. Which arithmetic operator would you use to find the remainder after division?

Answer:

Section C: Decision Statements (if, elif, else)

11. What is the output of the following code snippet?

```
age = 20
if age >= 18:
    print("Adult")
else:
    print("Minor")
```

Answer:

---

12. Which logical operator would you use to check if two conditions are both True?

Answer:

---

13. What is the output of the following code snippet?

```
password = "SECret123"
if "secret" in password.lower():
    print("Password contains 'secret'")
else:
    print("Password is okay")
```

Answer:

---

14. The following code checks if a number is positive, negative, or zero. Complete the missing condition on the second line.

```
num = -5
if num > 0:
    print("Positive")
_____ num < 0: # Fill in the blank
    print("Negative")
else:
    print("Zero")
```

Answer:

Section D: Iteration (Loops)

15. How many times will the word "Loop" be printed?

```
for i in range(5):
    print("Loop")
```

Answer:

---

16. What is the output of the following `while` loop?

```
count = 3
while count > 0:
    print(count)
    count -= 1
print("Go!")
```

Answer:

---

17. What does the `break` statement do inside a loop?

Answer:

Section E: Functions and Modules

18. What is the key difference between a built-in function (e.g., `print()`, `input()`) and a user-defined function?

Answer:

---

19. What is the purpose of the `return` statement in a function?

Answer:

---

20. What does the following `import` statement do?

```
from math import sqrt
```

Answer:

---

21. What will the following code snippet most likely output?

```
import random
print(random.randint(1, 6))
```

Answer:

Section F: String Functions

22. What is the value of the variable `result`?

```
text = " Hello World "
result = text.strip()
```

Answer:

---

23. What is the value of the variable `section`?

```
message = "Python is fun"
section = message[7:9]
```

Answer:

---

24. What does the `message.upper()` method do to the string stored in the variable `message`?

Answer:

Section G: Data Structures (Lists, Tuples, Dictionaries)

25. What is the main difference between a list and a tuple?

Answer:

---

26. What does the `append()` method do when used on a list?

Answer:

---

27. What is the output of the following code?

```
colours = ["red", "blue", "green"]
colours.pop(1)
print(colours)
```

Answer:

---

28. Which dictionary method returns the value for a given key, and returns a default value (e.g., "Not Found") if the key doesn't exist?

Answer:

---

29. What is the output of the following code?

```
student = {"name": "Emma", "grade": 85}
print(student["name"])
```

Answer:

Section H: Good Practices & Errors

30. List two examples of good programming practices mentioned in the syllabus.

Answer:

Python Programming - Level 1 (Easy) - Answers

1.

```
Hello
World
```

2. Kim scored 95%

3. A variable is a named memory location used to store a value.

4. Boolean (bool)

5. It pauses the program and waits for the user to type input, which is returned as a string.

6. **1010** (The input is a string, so `*` performs repetition, not multiplication).
7. It tries to concatenate a string (`"5"`) and an integer (`3`), which are incompatible types for the `+` operator in this context.

8. **1** (remainder of 10 divided by 3)

9. **3** (15 floor-divided by 4 is 3.75, which floors to 3).

10. The modulus operator `%`.

11. Adult

12. The `and` operator.

13. **Password contains 'secret'** (Because `.lower()` converts the password to lowercase before the check).

14. `Elif`

15. 5 times.

16.

```
3
2
1
Go!
```

17. It immediately terminates the loop it is in, and the program continues with the next statement after the loop.

18. Built-in functions are pre-defined in Python, while user-defined functions are created by the programmer to perform a specific task.

19. It specifies the value that a function should send back to the part of the program that called it.

20. It imports **only** the `sqrt` function from the `math` module, allowing it to be used directly without the `math.` Prefix.

21. A random integer between 1 and 6 (inclusive), such as 4.

22. "**Hello World**" (The leading and trailing spaces are removed).

23. "**is**" (Characters from index 7 up to, but not including, index 9).
24. It returns a new string where all alphabetic characters are converted to uppercase.

25. Lists are mutable (can be changed), while tuples are immutable (cannot be changed).

26. It adds a new element to the end of the list.

27. `['red', 'green']` (The element at index 1, "blue", is removed).

28. **The `.get()` method** (e.g., `value = my_dict.get("key", "Not Found")`).

29. Emma

30. Any two from: using inline comments, proper code indentation, and meaningful variable names