

Questions

Basics

1. What is an expression?

Q: An **expression** is a combination of values, variables, operators, and calls to functions. **Expressions** need to be evaluated.

2. What is a syntax error?

Q: An error that occurs when a compiler or interpreter cannot understand the source code statement in order to generate machine code.

3. What is PEP8?

Q: PEP 8, sometimes spelled PEP8 or PEP-8, is a document that provides guidelines and best practices on how to write Python code. The primary focus of PEP 8 is to improve the readability and consistency of Python code. PEP stands for Python Enhancement Proposal, and there are several of them.

4. What does a linter do?

Q: lint, or a linter, is a static code analysis tool used to flag programming errors, bugs, stylistic errors, and suspicious constructs. The term originates from a Unix utility that examined C language source code.

5. What is the result of this expression: `""" * 10`

Q: Prints the string `* 10` times which is `*****`

6. What is CPython?

Q: CPython is the reference implementation of the Python programming language. Written in C and Python, CPython is the default and most widely used implementation of the language. CPython can be defined as both an interpreter and a compiler as it compiles Python code into bytecode before interpreting it.

7. How is CPython different from Jython?

Q: CPython is the original implementation, written in C. (The "C" part in "CPython" refers to the language that was used to write the Python interpreter itself.) Jython is the same language (Python), but implemented using Java.

8. How is CPython different from IronPython?

Q: IronPython interpreter was written in C#.

Primitive Types

1. What is a variable?

Q: Variable is name given to memory, its value can be changed as many times as we wants

2. What are the primitive built-in types in Python?

Q: Python has four primitive types: integers, floats, booleans and strings.

3. When should we use `"""` (tripe quotes) to define strings?

Q: Spanning strings over multiple lines can be done using python's tripe quotes. It is also used for long comments in code. Special characters like TABs, verbatim or NEWLINEs

can also be used within the triple quotes.

4. Assuming (name = "John Smith"), what does name[1] return?

Q: Return character at index 1 which is 'o'

5. What about the name[-2]?

Q: Returns second last character.

6. What about the name[1:-1]?

Q: Return substring from index 1 to second last character.

7. How to get the length of a name?

Q: By using len function len(name)

8. What are the escape sequences in Python?

Q:

Backslash (\)

Single quote (')

Double quote (")

newline (\n)

Backspace(\b)

9. What is the result of f"{2+2}+{10%3}"?

Q: 4+1

10. Given (name = "john smith"), what will name.title() return?

Q: John Smith

The title() function in python is the Python String Method which is used to convert the first character in each word to Uppercase and remaining characters to Lowercase in the string and returns a new string.

11. What does name.strip() do?

Q: string.strip() without any arguments removed the whitespaces from the left and right of string.

12. What will name.find("Smith") return?

Q: Return index of the substring. If substring not found returns -1. Here returns -1 because it is case sensitive.

13. What will be the value of the name after we call name.replace("j", "k")?

Q: kohn smith

14. How can we check to see if the name contains "John"?

Q: print("John" in name); or print(name.find("John"))

15. What are the 3 types of numbers in Python?

Q: Numeric Types — int , float , complex. There are three distinct numeric types: integers, floating point numbers, and complex numbers.

Control Flow

1. What is the difference between 10 / 3 and 10 // 3?

Q: However, in Python3, dividing two ints results in a float, but using // acts as integer division.

10/3 answer 3.333333333

10//3 answer 3

2. What is the result of `10 ** 3`?

Q: `**` acts as exponential, that is 10 power 3 = 1000

3. Given (`x = 1`), what will be the value of after we run (`x += 2`)?

Q: Equivalent to `x=x+2`, which means `x=3`

4. How can we round a number?

Q: The `round()` function returns a floating point number that is a rounded version of the specified number, with the specified number of decimals.

The default number of decimals is 0, meaning that the function will return the nearest integer.

`x = round(5.76543, 2)`

5. What is the result of `float(1)`?

Q: 1.0

6. What is the result of `bool("False")`?

Q: Return True. Boolean value of string having content is true and empty string has boolean value false.

7. What are the falsy values in Python?

Q:

`bool(False)`

`bool(None)`

`bool(0)`

`bool("")`

`bool()`

`bool([])`

`bool({})`

8. What is the result of `10 == "10"`?

Q: Returns false because of type mismatch

9. What is the result of `"bag" > "apple"`?

Q: It's true because the first character of bag the alphabet 'b' comes after the first character of apple the alphabet 'a'.

10. What is the result of `not(True or False)`?

Q: Returns false

Any condition is either true or false so false hence (`True or False`) is always true, therefore `not(True or False)` is false.

11. Under what circumstances does the expression `18 <= age < 65` evaluate to True?

Q: If age is between 18 to 65 including 18 but excluding 65.

12. What does `range(1, 10, 2)` return?

Q: 1,3,5,7,9

Starting from 1 then incremented by 2 until number less than 10.

13. Name 3 iterable objects in Python.

Q: Lists, tuples, dictionaries, and sets are all iterable objects. They are iterable containers which you can get an iterator from.

Functions

1. What is the difference between a parameter and an argument?

Q: Function parameters are the names listed in the function's definition. Function arguments are the real values passed to the function. Parameters are initialized to the values of the arguments supplied.

2. All functions in Python by default return ...?

Q: A Python function will always have a return value. There is no notion of procedure or routine in Python. So, if you don't explicitly use a return value in a return statement, or if you totally omit the return statement, then Python will implicitly return a default value for you.

3. What are keyword arguments and when should we use them?

Q: Keyword arguments (or named arguments) are values that, when passed into a function, are identifiable by specific parameter names. A keyword argument is preceded by a parameter and the assignment operator, = .

4. How can we make a parameter of a function optional?

Q: A Python optional argument is a type of argument with a default value. You can assign an optional argument using the assignment operator in a function definition or using the Python

5. What happens when we prefix a parameter with an asterisk (*)?

Q: For numeric data types, * is used as multiplication operator

For sequences such as string, list and tuple, * is a repetition operator

6. What about two asterisks (**)?

Q: For numeric data types double asterisk (**) is defined as exponentiation operator

7. What is scope?

Q: Scope is the area of the program where an item (be it variable, constant, function, etc.) that has an identifier name is recognized. In our discussion, we will use a variable and the place within a program where the variable is defined determines its scope.

8. What is the difference between local and global variables?

Q: Local variable is declared inside a function whereas Global variable is declared outside the function. Local variables are created when the function has started execution and is lost when the function terminates, on the other hand, Global variable is created as execution starts and is lost when the program ends.

9. Why is using the **global** statement a bad practice?

Q: Global variables are dangerous because they can be simultaneously accessed from multiple sections of a program. This frequently results in bugs.

Coding Exercises (in Jupyter Notebook)

1. Write a function that returns the maximum of two numbers.
2. Write a function called **fizz_buzz** that takes a number.
 1. If the number is divisible by 3, it should return “Fizz”.
 2. If it is divisible by 5, it should return “Buzz”.
 3. If it is divisible by both 3 and 5, it should return “FizzBuzz”.
 4. Otherwise, it should return the same number.
3. Write a function for checking the speed of drivers. This function should have one parameter: speed.
 1. If speed is less than 70, it should print “Ok”.
 2. Otherwise, for every 5km above the speed limit (70), it should give the driver one demerit point and print the total number of demerit points. For example, if the speed is 80, it should print: “Points: 2”.
 3. If the driver gets more than 12 points, the function should print: “License suspended”
4. Write a function called **showNumbers** that takes a parameter called **limit**. It should print all the numbers between 0 and limit with a label to identify the even and odd numbers. For example, if the limit is 3, it should print:
 - 0 EVEN
 - 1 ODD
 - 2 EVEN
 - 3 ODD
5. Write a function that returns the sum of multiples of 3 and 5 between 0 and **limit** (parameter). For example, if the limit is 20, it should return the sum of 3, 5, 6, 9, 10, 12, 15, 18, 20.
6. Write a function called **show_stars(rows)**. If **rows** is 5, it should print the following:
 - **
 - ***
 - ****
 - *****
7. Write a function that prints all the prime numbers between 0 and **limit** where limit is a parameter.