

Assignment Part-1

Q1. Why do we call Python as a general purpose and high-level programming language?

Answer:

Python is a general-purpose programming language that can be used for a wide range of applications. It is considered high-level because it is designed to be easy to read and write, with a syntax that is closer to human language than machine language. This makes it easier for developers to write code quickly and efficiently. Python is also an interpreted language, which means that it does not need to be compiled before it can be run. This makes it easier to use for beginners who may not have experience with compiling code.

Python is often used for web development, scientific computing, data analysis, artificial intelligence, and more. Its versatility and ease of use make it a popular choice for developers of all levels of experience.

Q2. Why is Python called a dynamically typed language?

Answer:

Python is called a dynamically typed language because it does not require you to declare the type of a variable when you define it. Instead, the type of the variable is determined at runtime based on the value that is assigned to it. This makes Python more flexible and easier to use than statically typed languages like C or Java.

In Python, you can assign any type of value to a variable, and you can change the type of a variable simply by assigning a new value to it. This makes it easy to write code quickly and efficiently, without worrying about the details of data types.

Q3. List some pros and cons of Python programming language?

Answer:

Python is a popular programming language that has many advantages and disadvantages. Here are some of the pros and cons of Python:

Pros

Beginner-friendly

Large community

Flexible and extensible

Extensive libraries

Work environment

Cons

Issues with design

Slower than compiled languages

Security issues

Not ideal for mobile development

Python is a great choice for beginners because it has a simple syntax that is easy to learn. It also has a large community of developers who are always willing to help out. Python is also flexible and extensible, which means that it can be used for a wide range of applications.

However, Python does have some drawbacks. One issue with Python is that it can be slower than compiled languages like C or Java. Python also has some security issues that need to be addressed. Additionally, Python is not the best choice for mobile development.

Q4. In what all domains can we use Python?

Answer:

Python is a versatile programming language that can be used in many different domains. Here are some of the most popular applications of Python:

Web development

Data analysis and visualization

Machine learning and artificial intelligence

Scientific computing

Game development

Desktop applications

Python is also used in many other domains, including finance, education, and healthcare. Its flexibility and ease of use make it a popular choice for developers who want to build applications quickly and efficiently.

Q5. What are variable and how can we declare them?

Answer:

In Python, a variable is a name that refers to a value. You can think of a variable as a container that holds data. To create a variable in Python, you simply assign a value to a name using the = operator.

Here's an example:

```
x = 5
```

Copy

In this example, we've created a variable named x and assigned it the value 5. We can then use the variable x in our code to refer to this value.

Python is a dynamically typed language, which means that you don't need to declare the type of a variable when you create it. The type of the variable is determined at runtime based on the value that is assigned to it.

Q6. How can we take an input from the user in Python?

Answer:

In Python, you can use the input() function to get input from the user. The input() function takes a single argument, which is a prompt that is displayed to the user before they enter their input.

Here's an example:

```
name = input("What is your name? ")
```

```
print("Hello, " + name + "!")
```

Copy

In this example, we're using the input() function to get the user's name. We're then using the print() function to display a greeting that includes the user's name.

When you run this code, you'll see a prompt that says "What is your name?" You can then enter your name, and the program will display a greeting that includes your name.

Q7. What is the default datatype of the value that has been taken as an input using input() function?

Answer:

In Python, the input() function always returns a string. This means that if you use the input() function to get a number from the user, you'll need to convert it to an integer or a float before you can use it in mathematical operations.

Here's an example:

```
age = input("What is your age? ")
```

```
age = int(age)
```

Copy

In this example, we're using the input() function to get the user's age. We're then using the int() function to convert the age to an integer.

If you try to perform mathematical operations on a string that represents a number, you'll get a TypeError. For example:

```
age = input("What is your age? ")
```

```
print(age + 5)
```

Copy

This code will raise a TypeError because you can't concatenate a string and an integer. To fix this, you'll need to convert the age to an integer first:

```
age = input("What is your age? ")
```

```
age = int(age)
```

```
print(age + 5)
```

Q8. What is type casting?

Answer:

Type casting is the process of converting a value from one data type to another. In Python, you can use built-in functions like int(), float(), and str() to convert values from one data type to another.

Here are some examples:

```
# Convert a string to an integer
```

```
age = "25"
```

```
age = int(age)
```

```
# Convert a string to a float
```

```
price = "19.99"
```

```
price = float(price)
```

```
# Convert an integer to a string
```

```
count = 10
```

```
count = str(count)
```

Copy

In these examples, we're using the `int()`, `float()`, and `str()` functions to convert values from one data type to another.

Type casting is useful when you need to perform operations on values of different data types. For example, if you have a string that represents a number, you'll need to convert it to an integer or a float before you can perform mathematical operations on it.

Q9. Can we take more than one input from the user using single `input()` function? If yes, how? If no, why?

Answer:

In Python, you can use the `input()` function to get input from the user. However, the `input()` function only allows you to get one input at a time.

If you need to get multiple inputs from the user, you can call the `input()` function multiple times. For example:

```
name = input("What is your name? ")
```

```
age = input("What is your age? ")
```

Copy

In this example, we're using the `input()` function twice to get the user's name and age.

You can also use string concatenation to create a prompt that asks for multiple inputs. For example:

```
info = input("What is your name and age? ")
```

```
name, age = info.split()
```

Copy

In this example, we're using the `split()` method to split the user's input into two separate values.

Q10. What are keywords?

Answer:

In Python, keywords are reserved words that have a special meaning and cannot be used as variable names. Keywords are used to define the syntax and structure of the Python language.

Here are some examples of Python keywords:

and

as

assert

break

class

continue

def

del

elif

else

except

False

finally

for

from

global

if

import

in

is

lambda

None

nonlocal

not

or

pass

raise

return

True

try

while

with

yield

Q11. Can we use keywords as a variable? Support your answer with reason.

Answer:

You cannot use keywords as variable. If you try to use a keyword as a variable name, you'll get a syntax error. For example:

```
class = "MyClass"
```

Q12. What is indentation? What's the use of indentation in Python?

Answer:

Indentation refers to the spaces at the beginning of a code line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code.

Q13. How can we throw some output in Python?

Answer:

The basic way to do output is the print statement.

```
print('Hello, world')
```

To print multiple things on the same line separated by spaces, use commas between them:

```
print('Hello,', 'World')
```

This will print out the following:

Hello, World

Q14. What are operators in Python?

Answer:

In Python, operators are special symbols that designate that some sort of computation should be performed. The values that an operator acts on are called operands.

Here is an example:

```
>>> a = 10
```

```
>>> b = 20
```

```
>>> a + b
```

```
30
```


Q15. What is difference between / and // operators?

Answer:

In Python, the / operator performs floating-point division, while the // operator performs integer division.

Here's an example:

```
# Floating-point division
```

```
result = 10 / 3
```

```
print(result) # Output: 3.3333333333333335
```

```
# Integer division
```

```
result = 10 // 3
```

```
print(result) # Output: 3
```

Copy

In this example, we're using the / operator to perform floating-point division and the // operator to perform integer division.

Floating-point division returns a floating-point number, which means that the result can have a decimal part. Integer division returns an integer number, which means that the result is rounded down to the nearest integer.

Q16. Write a code that gives following as an output.

```
...
```

```
iNeuroniNeuroniNeuroniNeuron
```

```
...
```

Answer:

Here's a Python code that will give you the output you're looking for:

```
word = "Neuron"
```

```
result = "i" + word * 4
```

```
print(result)
```

Copy

In this code, we're creating a variable called word that contains the string "Neuron". We're then creating a variable called result that contains the string "i" followed by the word string repeated four times. Finally, we're printing the result variable.

Q17. Write a code to take a number as an input from the user and check if the number is odd or even.

Answer:

Here's a Python code that will take a number as input from the user and check if the number is odd or even:

```
number = int(input("Enter a number: "))
```

```
if number % 2 == 0:
```

```
    print("The number is even.")
```

```
else:
```

```
    print("The number is odd.")
```

Copy

In this code, we're using the input() function to get a number from the user. We're then using the int() function to convert the input to an integer. We're then using an if statement to check if the number is even or odd. If the number is even (i.e., the remainder of the division by 2 is 0), we print "The number is even." If the number is odd (i.e., the remainder of the division by 2 is not 0), we print "The number is odd."

Q18. What are boolean operator?

Answer:

In Python, boolean operators are used to perform logical operations on boolean values. Boolean values are either True or False.

Here are the three boolean operators in Python:

and: Returns True if both operands are True.

or: Returns True if at least one operand is True.

not: Returns the opposite of the operand.

Here are some examples:

```
# and operator
```

```
result = True and False
```

```
print(result) # Output: False
```

```
# or operator
```

```
result = True or False
```

```
print(result) # Output: True
```

```
# not operator
```

```
result = not True
```

```
print(result) # Output: False
```

Copy

In these examples, we're using the and, or, and not operators to perform logical operations on boolean values.

Q19. What will the output of the following?

```
'''
```

```
1 or 0
```

```
0 and 0
```

```
True and False and True
```

```
1 or 0 or 0
```

...

Answer:

Here are the outputs of the expressions you provided:

1 or 0 # Output: 1

0 and 0 # Output: 0

True and False and True # Output: False

1 or 0 or 0 # Output: 1

Copy

In the first expression, 1 or 0, the or operator returns the first operand that evaluates to True, which is 1.

In the second expression, 0 and 0, the and operator returns the first operand that evaluates to False, which is 0.

In the third expression, True and False and True, the and operator returns the first operand that evaluates to False, which is False.

In the fourth expression, 1 or 0 or 0, the or operator returns the first operand that evaluates to True, which is 1.

Q20. What are conditional statements in Python?

Answer:

Conditional statements (if, else, and elif) are fundamental programming constructs that allow you to control the flow of your program based on conditions that you specify. They provide a way to make decisions in your program and execute different code based on those decisions.

Q21. What is use of 'if', 'elif' and 'else' keywords?

Answer:

The if / elif / else structure is a common way to control the flow of a program, allowing you to execute specific blocks of code depending on the value of some data

Q22. Write a code to take the age of person as an input and if age ≥ 18 display "I can vote". If age is < 18 display "I can't vote".

Answer:

Here's a Python code that will take the age of a person as input and display "I can vote" if the age is greater than or equal to 18, and "I can't vote" if the age is less than 18:

```
age = int(input("Enter your age: "))
```

```
if age  $\geq$  18:
```

```
    print("I can vote.")
```

```
else:
```

```
    print("I can't vote.")
```

Copy

In this code, we're using the input() function to get the age of the person as input from the user. We're then using the int() function to convert the input to an integer. We're then using an if statement to check if the age is greater than or equal to 18. If the age is greater than or equal to 18, we print "I can vote." If the age is less than 18, we print "I can't vote."

Q23. Write a code that displays the sum of all the even numbers from the given list.

```
...
```

```
numbers = [12, 75, 150, 180, 145, 525, 50]
```

```
...
```

Answer:

Here's a Python code that will display the sum of all the even numbers from the given list:

```
numbers = [12, 75, 150, 180, 145, 525, 50]
```

```
sum = 0
```

```
for number in numbers:

    if number % 2 == 0:

        sum += number


print("The sum of all the even numbers in the list is:", sum)
```

Copy

In this code, we're creating a list called numbers that contains the given numbers. We're then creating a variable called sum and initializing it to 0. We're then using a for loop to iterate over each number in the list. We're then using an if statement to check if the number is even (i.e., the remainder of the division by 2 is 0). If the number is even, we add it to the sum variable. Finally, we're printing the sum of all the even numbers in the list.

Q24. Write a code to take 3 numbers as an input from the user and display the greatest no as output.

Answer:

Here's a Python code that will take three numbers as input from the user and display the greatest number as output:

```
number1 = int(input("Enter the first number: "))
number2 = int(input("Enter the second number: "))
number3 = int(input("Enter the third number: "))
```

```
if number1 > number2 and number1 > number3:

    print("The greatest number is:", number1)
elif number2 > number1 and number2 > number3:

    print("The greatest number is:", number2)
else:

    print("The greatest number is:", number3)
```

Copy

In this code, we're using the input() function to get three numbers from the user. We're then using the int() function to convert the inputs to integers. We're then using an if statement to check which of the three numbers is the greatest. If number1 is greater than both number2 and number3, we print "The greatest number is:" followed by number1. If number2 is greater than both number1 and

number3, we print "The greatest number is:" followed by number2. Otherwise, we print "The greatest number is:" followed by number3.

Q25. Write a program to display only those numbers from a list that satisfy the following conditions

- The number must be divisible by five
- If the number is greater than 150, then skip it and move to the next number
- If the number is greater than 500, then stop the loop

...

```
numbers = [12, 75, 150, 180, 145, 525, 50]
```

...

Answer:

Here's a Python code that will display only those numbers from the given list that satisfy the following conditions:

The number must be divisible by five.

If the number is greater than 150, then skip it and move to the next number.

If the number is greater than 500, then stop the loop.

```
numbers = [12, 75, 150, 180, 145, 525, 50]
```

```
for number in numbers:
```

```
    if number % 5 == 0:
```

```
        if number > 150:
```

```
            continue
```

```
        elif number > 500:
```

```
            break
```

```
        else:
```

```
            print(number)
```

Copy

In this code, we're creating a list called `numbers` that contains the given numbers. We're then using a `for` loop to iterate over each number in the list. We're then using an `if` statement to check if the number is divisible by five (i.e., the remainder of the division by 5 is 0). If the number is divisible by five, we're then using another `if` statement to check if the number is greater than 150. If the number is greater than 150, we're using the `continue` statement to skip it and move to the next number. If the number is greater than 500, we're using the `break` statement to stop the loop. Otherwise, we're printing the number.