

Python Programming

Assignment # 1



Submitted by
Viquar Fathima

Submitted to
Bhavani kairika

Department of Computer Science
Vaagdevi Engineering College

(Q1) Define Artificial Intelligence (AI) and provide examples of its applications

Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition, and machine vision.

Artificial intelligence examples:

> Google Translate: Uses deep learning algorithms to translate text from one language to another.

> Netflix: Uses machine learning algorithms to create personalized recommendation engines for users based on their previous viewing history.

> Tesla: Uses computer vision to power self-driving features on their cars.

> ChatGPT: Uses large language models (LLMs) to generate text in response to questions or comments posed to it.

(Q2) What is Python? Discuss its main features and advantages.

Python is a dynamic, high-level, free open source, and interpreted programming language. It supports object-oriented programming as well as procedural-oriented programming. In Python, we don't need to declare the type of variable because it is a dynamically typed language. For example, `x = 10` Here, `x` can be anything such as String, int, etc. In this article, we will see what characteristics describe the Python programming language

Features in Python:

- > Free and Open Source*
- > Easy to code*
- > Easy to Read*
- > Object-Oriented Language*
- > GUI Programming Support*
- > High-Level Language=*
- > Easy to Debug*

(Q3) What are the advantages of using Python as a programming language for AI and ML?

Python is an outstanding language majorly because it doesn't need compiling into machine language instruction to be executed. A developer can directly run a program written in Python.

But other than this, there are many more benefits of developing AI projects using Python.

> A huge library ecosystem:

Python offers a vast choice of libraries for AI development, which contain base-level items that save coding time. These libraries also make it easy to access, handle, and transform data

> High readability

Python is famous for its compact, readable code, and is practically unmatched with regards to usability, especially for new developers. This has made it a preferred language for AI and deep learning

> The flexibility of the language

Python for AI is an extraordinary language, as it is truly flexible

Python is an open-source programming language and is supported by a ton of assets and top-notch documentation. It additionally flaunts a huge and dynamic network of developers ready to give guidance and help through all phases of the development procedure.

(Q4) Discuss the importance of indentation in Python code.

When writing a Python program, programmers usually write a block of code (a collection of statements) for conditional statements, functions, and loops. When the program increases in size, it gets more difficult to read and comprehend due to these conditional statements, functions, and loops. Furthermore, the Python interpreter needs help determining the sequence of execution of statements. Indentation is used to simplify things. We indent the code after breaking it into many sections. This indentation aids the Python interpreter in comprehending the following:

The order in each block of code or statement should be executed

Which statement belongs to which code block Python indentation informs the interpreter that a set of statements belongs to a certain block of code.

Rough Example Of How Python Indentation Works:

Statement: Code block 1 statement begins.

If condition: Code block one condition continues.

If condition: Code block 2 statement begins.

Statement: Code block 3 statement begins.

Else: Code block 2 statement continues.

Statement: Code block 3 continues.

Statement: Code block 1 continues.

(Q5) Define a variable in Python. Provide examples of valid variable names.

Variable Names

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).

Rules for Python variables:

A variable name must start with a letter or the underscore character. A variable name cannot start with a number. A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)

Variable names are case-sensitive (age, Age, and AGE are three different variables)

Examples :

myvar = "hello"

my_var = "hello"

_my_var = "hello"

myVar = "hello"

MYVAR = "hello"

myvar2 = "hello"

(Q6) Explain the difference between a keyword and an identifier in Python.

Python Keywords:

Keywords are predefined, reserved words used in Python programming that have special meanings to the compiler. We cannot use a keyword as a variable name, function name, or any other identifier. They are used to define the syntax and structure of the Python language.

All the keywords except True, False, and None are in

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

lowercase and they must be written as they are.

Python Identifiers:

Identifiers are the names given to variables, classes, methods(functions), etc.

For example,


```
language = 'Python'
```

Here, language is a variable (an identifier) that holds the value 'Python'. We cannot use keywords as variable names as they are reserved names that are built-in in Python.

```
For example, continue = 'Python'
```

The above code is wrong because we have used continue as a variable name.

Rules for Naming an Identifier

- > Identifiers cannot be a keyword.*
- > Identifiers are case-sensitive.*
- > It can have a sequence of letters and digits. However, it must begin with a letter or _. The first letter of an identifier cannot be a digit.*
- > It's a convention to start an identifier with a letter rather than _.*
- > Whitespaces are not allowed.*

> We cannot use special symbols like !, @, #, \$

(Q7) List the basic data types available in Python.

Python Data Types :

> Built-in Data Types

In programming, data type is an important concept.

Variables can store data of different types, and different types can do different things. Python has the following data types built-in by default, in these categories:

Text Type: str

Numeric Types: int, float, complex

Sequence Types: list, tuple, range

Mapping Type: dict

Set Types: set, frozenset

Boolean Type: bool

Binary Types: bytes, bytearray, memoryview

None Type: NoneType

(Q8) Describe the syntax for an if statement in Python.

Conditional Statements are statements in Python that provide a choice for the control flow based on a condition. It means that the control flow of the Python program will be decided based on the outcome of the condition.

Types of Conditional Statements in Python

- 1. If Conditional Statement in Python*
- 2. If else Conditional Statement in Python*
- 3. Nested if..else Conditional Statement in Python*
- 4. If-elif-else Conditional Statement in Python*
- 5. Ternary Expression Conditional Statement in Python*

1. If Conditional Statement in Python If the simple code of block is to be performed if the condition holds then the if statement is used. Here the condition mentioned holds then the code of the block runs otherwise not.

Syntax of If Statement:

if condition:

 # Statements to execute if

 # condition is true

2. If else Conditional Statements in Python

In a conditional if Statement the additional block of code is merged as an else statement which is performed when if condition is false.

Syntax of Python If-Else:

if (condition):

 # Executes this block if

 # condition is true

else:

Executes this block if

condition is false

3. Nested if..else Conditional Statements in Python

Nested if..else means an if-else statement inside another if statement. In simple words first, there is an outer if statement, and inside it another if-else statement is present and such type of statement is known as a nested if statement. We can use one if or else if statement inside another if or else if statement.

4. If-elif-else Conditional Statements in Python

The if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the ladder is bypassed. If none of the conditions is true, then the final “else” statement will be executed.

5. Ternary Expression Conditional Statements in Python

The Python ternary Expression determines if a condition is true or false and then returns the appropriate value by the result. The ternary Expression is useful in cases where we need to assign a value to a variable based on a simple condition, and we want to keep our code more concise — all in just one line of code.

Syntax of Ternary Expression

Syntax: [on_true] if [expression] else [on_false]

expression: conditional_expression | lambda_expr

(Q9) Explain the purpose of the elif statement in Python.

The elif keyword is Python's way of saying "if the previous conditions were not true, then try this condition". Python elif (short for else if) is used to execute a continuous chain of the conditional logic ladder.

In elif, there are multiple conditions and the corresponding statement(s) as a ladder. Only one of the blocks gets executed when the corresponding boolean expression evaluates to true.

Syntax of elif :

if boolean_expression_1:

statement(s)

elif boolean_expression_2:

statement(s)

elif boolean_expression_3:

statement(s)

else

statement(s)

EXAMPLPLE: *a = 33*

b = 33

if b > a:

print("b is greater than a")

elif a == b:

print("a and b are equal")

(Q10) Differentiate between supervised and unsupervised learning techniques in ML.

	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Input Data</i>	<i>Uses Known and Labeled Data as input</i>	Uses Unknown Data as input
<i>Computational Complexity</i>	Less Computational Complexity	More Computational Complex
<i>Real-Time</i>	Uses off-line analysis	Uses Real-Time Analysis of Data
<i>Number of Classes</i>	The number of Classes is known	The number of Classes is not known
<i>Accuracy of Results</i>	Accurate and Reliable Results	Moderate Accurate and Reliable Results
<i>Output data</i>	The desired output is given.	The desired, output is not given.
<i>Model</i>	In supervised learning, it is not possible to learn larger and more complex models than in unsupervised learning	In unsupervised learning, it is possible to learn larger and more complex models than in supervised learning
<i>Training data</i>	In supervised learning training data is used to infer model	In unsupervised learning training data is not used.

	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Another name</i>	Supervised learning is also called classification.	Unsupervised learning is also called clustering.
<i>Test of model</i>	We can test our model.	We can not test our model.
<i>Example</i>	Optical Character Recognition	Find a face in an image.