Untitled

March 26, 2023

[ ]:
```
1.Who developed  python programming language?

A:   Python programming language was developed by Guido van Rossum in the late
 ↪1980sand early 1990s.
     He developed python programming language while working at National
 ↪Research Institute for Mathematics and
     Computer Science in the Netherlands.
     Guido aimed to create a language that was easy to learn and us,while still
 ↪being powerful and expressive.
     He named the language after Monty Python's Flying circus,a British comedy
 ↪group.
     Python has become one of the most popular programming languages in the
 ↪world, used for a wide range of
     applications, including web development, scientific computing, data
 ↪analysis, and artificial intelligence.
     It is known for its clear and concise syntax,which makes it easy to read
 ↪and write code in Python.
```

[ ]:
```
 2.Which type of programming does python support?

A:   Python is a high-level ,general-purpose programming language that supports
 ↪several programming paradigms, including:
                          1.Procedural programming
                          2.Object-Oriented Programming(OOP)
                          3.functional Programming
                          4. Aspect-Oriented Programming(AOP)
                          5.Structured programming
                          6.Imperative Programming
                          7.Event-Driven Programming
  Hence,python supports multiple programming paradigms.
```

[ ]:
```
 3.Is python case sensitive when dealing with identifiers?

 A: Yes, Python is case sensitive programming language when it comes to
 ↪identifiers,which means that it distinguishes
    between uppercase and lowercase letters in identifiers(such as  variable
 ↪names, function names, and class names).
```

```
   For example, the following are 3 different   identifiers in   python:
                                    1.myVariable
                                    2.MyVariable and
                                    3. myvariable
   If we use these identifiers interchangeably or in a different case,python␣
↪will treatthem as separate entities
   and may cause errors in your program.Therefore, it's essential to be␣
↪consistent with the case of your identifiers
   throughout your code.

4.  What is the correct extension of the python file?

A:  The correct extension for Python files is ".py".
     For example, if you have a Python program named "my_result", you would␣
↪save it in a file named "my_result.py" .
     The ".py" extension is used  to identify files containing python code,␣
↪and it is recognized by the python interpreter
     as a valid file type for execution.


5.Is python code compiled or interpreted?

A: Python code is interpreted because Python is an interpreted language.
     When you write Python code and run it, the Python interpreter reads the␣
↪code line by
     line and translates it into a low-level representation called bytecode.␣
↪The  bytecode isthen executed by the python
     virtual machine.This process of interpretation and translation happens␣
↪each time the Python program is run.
     Python's interpretation and compilation process make it a highly flexible␣
↪and dynamic language that allows for rapid
     development and prototyping. Because the interpretercan provide immediate␣
↪feedback on syntax and other errors, it makes
     it easier for developers to write and test their code quickly.
 Overall, Python's interpretation and compilation process, combined with its␣
↪readabilityand ease of use, make it a popular
     choice for a wide range of applications, from simple scripts to␣
↪large-scale software systems.
```

[ ]:
```
6.Name a few blocks of code used to define in python language.


A:  In Python, code blocks are defined using indentation. Here are a few␣
↪examples of code blocks in Python:
```

```
    1.Function definition block:
Ex:
   def  my_function(p1,p2):
          result=p1+p2
          return result
    In this example, the code block for the my_function function is defined by the
    indentation level of the code within the function.

    2.Conditional statement block:
       Ex:
             If  x>0:
                  print("x is positive")
           Else:
                 print("x is not positive")
 In this example, the code block for the if statement is defined by the indentation level of the code within the
 if block and the code block for the else statement is defined by the indentation level of the code within the else block.

    3. Loop block:
  Ex:
       For i in range(10):
              print(i)
In this example, the code block for the for loop is defined by the indentation level of the code within the loop block.
Overall, in Python, any code that is indented at the same level is considered to be part of the same code block.
```

[ ]:
```
7.State a character used to give single-line comments in Python?


A: In Python, the character used to give a single-line comment is the hash symbol (#).
   Any text that follows the hash symbol on the same line is considered a comment and
    is ignored by the python interpreter.
    For example:
    #This is  a single line comment in python.
     print("hello world") # This is another single-line comment.
    In this code, the first line is a comment that does not affect the code execution.
    The second line prints the string "Hello, World!" to the console,and the text after the hash
    symbol is another comment that is also ignored by the interpreter.
```

[ ]:
```
8. Mention the  functions which can help us to find the version of python that
 ↪we are
   currently working on.

 A: In python, there are several functions that can help us find the version of
 ↪Python we're currently working on:
    1.sys.version: This function returns a string containing the version number
 ↪of the current Python interpreter, along
   with other information about the build.
   2.platform.python_version(): This function returns a string containing the
 ↪version number of the current Python interpreter
   , in a format that's easier to parse than the output of sys.version.
   3. sys.version_info: This function returns a tuple containing several pieces
 ↪of information about the current Python
      interpreter, including the major and minor version numbers.
  Here's an example of how you can use these functions to find the version of
 ↪Python
  you're currently working on.
    Ex:
               import sys
               import platform
               print("using python version:",sys.version)
               print("using python version:",platform.python_version())
               print("using python version:",sys.version_info)
  When we run this code, it will output information about the version of Python
 ↪you're currently working on,
  depending on which function you use.



9.Python supports the creation of anonymous functions at runtime,using
 ↪construct called "lambda functions"
  Lambda functions, also known as anonymous functions, are small,
 ↪single-expression functions that don't have a name.
  They are defined using the lambda keyword and can take any number of
 ↪arguments.

The syntax for defining a lambda function is as follows:

python
Copy code
lambda arguments: expression


Here, arguments refers to the arguments that the lambda function takes, and
 ↪expression is the single expression that
```

```
the function evaluates and returns.
For example, here is a lambda function that adds two numbers:

csharp
Copy code
add = lambda x, y: x + y


10.What does pip stand for python.

A: pip stands for "pip installs packages" (or "Pip Installs Python",
 ↪according to its documentation).
Pip is a package manager for Python that is used to install, upgrade, and
 ↪manage packages or modules that
are used in Python projects.
Pip is a command-line utility that comes pre-installed with most Python
 ↪distributions, including Python 2.7.9 and later,
and Python 3.4 and later. Pip simplifies the process of installing and managing
 ↪Python packages by automating the process
of downloading, building, and installing packages and their dependencies.
Overall, pip is an essential tool for Python developers as it makes it easy to
 ↪manage and install third-party packages or
modules, which can save a lot of time and effort in the development process.



11.Mention a few built-in functions in python?

A:1.print( ) function.
  2.type( ) function.
  3.input( ) function.
  4.abs( ) function.
  5.pow( ) function.
  6.dir( ) function.
  7.sorted( ) function.
  8.max( ) function.
  9. divmod( ) function
  10.len( ) function
  11.sum( ) function



12. What is the maximum possible length of an identifier in python?

A: An identifier can have a maximum length of 79 characters in
   Python.
 1.Python, particularly when combined with identifiers, is   case-sensitive.
```

```
 2.When writing or using identifiers in Python, it has a maximum of 79␣
 ↪characters.
 3.Unlikely, Python gives the identifiers unlimited length.However, the layout␣
 ↪of PEP-8 prevents
   the user from breaking the rules and includes a 79-character limit.


13.What are the benefits of using python?

A:1.Simple and easy to learn
  2.Cross-platform compatibility
  3.Large standard library
  4.High-level language
  5.Interpreted language
  6.Dynamic typing
  7.Productivity has increased
Overall, the simplicity, ease of use, cross-platform compatibility, and␣
 ↪extensive library support make
Python a popular choice for a wide range of applications, including web␣
 ↪development, scientific computing, data analysis, artificial intelligence,␣
 ↪and more.



14. How memory is managed in python?

A: In Python, memory is managed using a private heap space that is managed by␣
 ↪the Python memory manager.
   The memory manager uses a combination of techniques such as 1.reference␣
 ↪counting
                                  2.garbage collection and
                                  3.memory pooling  to manage memory␣
 ↪efficiently.
1.Reference counting: Python uses a technique called reference counting to keep␣
 ↪track of how many references
  there are to an object
2.Garbage collection: Python also uses a technique called garbage collection to␣
 ↪reclaim memory that is no longer being used
3.Memory pooling: Python uses a technique called memory pooling to speed up the␣
 ↪allocation of memory for small objects.



15.How to install python on windows and set path variables?

A: Here are the steps to install Python on Windows and set the path variables:
```

```
step1:Download the Python installer from the official Python website (https://
↪www.python.org/downloads/windows/).
step2:Run the installer and follow the on-screen instructions to install␣
↪Python. Make sure to select the option to
      add Python to your system PATH during the installation process.
step3:Once the installation is complete, open the Windows Start menu and␣
↪search for "Environment Variables".
      Click on the "Edit the system environment variables" option.
step4:In the System Properties window, click on the "Environment Variables"␣
↪button.
step5:In the Environment Variables window, scroll down to the "System␣
↪Variables" section and find the "Path" variable.
      Click on the "Edit" button.
step6:In the Edit Environment Variable window, click on the "New" button and␣
↪enter the path to the Python installation
      directory. For example, if Python is installed in the C:\Python39␣
↪directory, enter "C:\Python39" in the "New" field.
step7:Click "OK" on all of the windows to close them and save the changes.
step8:Open a new command prompt window and type "python" to verify that Python␣
↪is installed and the path variables are set
      correctly.
That's it! we should now have Python installed on your Windows system and the␣
↪path variables set correctly so that we can
run Python from any directory in the command prompt.



16.Is indentation required in python?

A:Yes, indentation is required in Python. In fact, indentation is a fundamental␣
↪aspect of Python's syntax and
  is used to indicate the structure of a program.
  In Python, blocks of code are defined by their indentation level, rather␣
↪than by braces or other delimiters
  like in other programming languages. This means that the amount of␣
↪indentation in a Python program determines
  the nesting of statements within blocks, such as loops and conditional␣
↪statements.
For example, here's a simple Python program that demonstrates the use of␣
↪indentation:

EX:
if x > 0:
    print("x is positive")
else:
```

```
    print("x is zero or negative")

In this program, the if statement and the else statement are indented at the␣
 ↪same level, indicating that they are
part of the same block of code. The print statements inside each block are␣
 ↪indented further to indicate that they are
executed only if the corresponding condition is true.
Python uses a consistent number of spaces for indentation, typically 4 spaces␣
 ↪per level. Mixing spaces and tabs can cause
syntax errors in our code.
```