

Python Keywords, Identifiers, Statement and Comment

Assignment Answers

Q1. Significance of Python Keywords

Python keywords are reserved words that have special meanings in the Python language. They are integral to Python's syntax and cannot be used as identifiers (names for variables, functions, class etc.). Keywords define the structure and flow of the language. Examples include if, for, while, def, and class.

Examples of Python Keywords:

- a. if: Used for conditional statements.

```
if x > 10:  
    print("x is greater than 10")
```

- b. for: Used for looping over a sequence.

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

- c. while: Used for looping as long as a condition is true.

```
while x < 5:  
    x += 1
```

- d. def: Used to define a function.

```
def my_function():  
    return "Hello"
```

- e. class: Used to define a class.

```
class MyClass:  
    pass
```

Q2. Rules for Defining Identifiers in Python

Identifiers are names used to identify variables, functions, classes, and other objects in Python. The rules for defining identifiers are:

- a. Names must start with a letter (a-z, A-Z) or an underscore (_).
- b. Names can be followed by letters, digits (0-9), or underscores.
- c. Names are case-sensitive (e.g., variable, Variable, and VARIABLE are different identifiers).
- d. Identifiers cannot be a keyword.

Q3. Comments in Python

Comments are used to explain and document code. They are ignored by the Python interpreter and are useful for making the code more understandable for others (or yourself) and for documenting the purpose and functionality of the code.

Types of Comments with example are:

- a. Single-line comments: (Start with a #)

```
# This is a single-line comment
x = 10 # This is also a comment
```

- b. Multi-line comments: Enclosed in triple quotes (' ' ' or " " ").

```
"""
This is a multi-line comment.
It can span multiple lines.
"""
y = 20
```

Why Comments Are Useful:

- a. They help explain complex code.
- b. They provide context or rationale for code choices.
- c. They aid in future code maintenance and updates.

Q4. Importance of Proper Indentation in Python

Indentation is crucial in Python because it defines the structure of the code. Python uses indentation to delimit blocks of code, such as those inside loops, functions, and conditionals. Unlike other programming languages that use braces { } to define code blocks, Python relies on consistent indentation. It also helps us to understand the beginning of loop body more effectively.

Q5. Incorrect indentation will lead to 'IndentationError' or 'SyntaxError'. Python requires consistent use of indentation (usually 1 tab) within the same block of code. Mixing tabs and spaces can also cause errors. It also makes difficult for the programmer to understand the beginning and ending of a loop.

Q6. Difference between Expression and Statement in Python

- **Expression:** An expression is a piece of code that evaluates to a value. It can be a variable, a constant, or a combination of operators and operands.
- **Statement:** A statement is a line of code that performs an action. Statements include assignments, function calls, loops, and conditionals.

Ex: 5 + 3 # Expression that evaluates to 8

 x = 10 # Assignment statement, but `10` is an expression

