Lesson:

Python Keywords, Identifiers, Comments, Indentation and Statements

pw skills

1. Explain the significance of Python keywords and provide examples of five keywords

Python keywords are predefined reserved words with special meanings and functionalities within the Python programming language. These keywords serve specific purposes and cannot be used as identifiers (such as variable names or function names). Understanding Python keywords is crucial for writing clear, concise, and efficient Python code. They dictate the structure and behavior of Python programs, facilitating tasks such as defining control flow

Example- ‘if’ , ‘for’ ,’ while’ , ‘def’ , ‘return’ e.t.c

a) IF:- if: The **’IF’** keyword is used to define conditional statements. It allows you to execute a block of code only if a specified condition is true.

```
x = 10

if x > 5:

    print("x is greater than 5")
```

B) FOR:-The '**FOR'** keyword is used to create loops.it iterates over elements of a sequence(such as lists, tuples,string)or any iterables object.

Code:-

```
for i in range(5):

    print(i)
```

C) While:- The ' **WHILE**' keyword is used to create a loop that continues executing as long as a specified condition evaluates to true.

CODE:-

```
count = 0

while count < 5:

    print(count)

    count += 1
```

D) Def :- The '**Def'** keyword is used to define a function. It indicates the beginning of a function definition block.

CODE :-

```
def greet(name):
```

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

E):- RETURN :- The ‘**Return**’ keyword is used inside a function to exit the function and return a value to the caller.

CODE :-

```
def add(x, y):
    return x + y
result = add(3, 4)
print(result)
# Output: 7
```

2. Describe the rules for defining identifiers in Python and provide examples.

1. **Rules for Naming an Identifier**
2. **Identifiers cannot be a keyword.**
3. **Identifiers are case-sensitive.**
4. **It can have a sequence of letters and digits.**
5. **It's a convention to start an identifier with a letter rather .**

6. White Spaces are not allowed.

7. We cannot use special symbols like !, @, #, $, and so on.

- **Example-**

```
# Valid identifiers

my_variable = 10

myVar = 20

MyVar = 30

_my_var = 40

MY_VAR = 50

var123 = 60




# Invalid identifiers

# 123 var = 70      # Cannot start with a digit
```

```
# my-var = 80      # Cannot contain special characters except underscore

# for = 90         # Cannot use reserved keyword as identifier

# my variable = 100 # Cannot have spaces
```

3. What are comments in Python, and why are they useful

Provide an example.

Python comments are simple sentences that we use to make the code easier to understand. They explain your way of thinking and describe every step that you take to solve a coding problem. These sentences are not read by the Python interpreter when it executes the code.

Comments can be used to explain Python code. Comments can be used to make the code more readable. Comments can be used to prevent execution when testing code.

Example-

```
# This program calculates the area of a circle
```

```
# Constants

PI = 3.14159

# Input radius from the user

radius = float(input("Enter the radius of the circle: "))

# Calculate area

area = PI * radius**2

# Display the result

print("The area of the circle with radius", radius, "is", area)
```

4. Why is proper indentation important in Python

Ans:- If you're coding in Python, one of the most important things to remember is the proper indentation of your code. Good Python indentation is necessary to make your code readable and easy to understand, both for yourself and others who may read your code later.

5.What happens if indentation is incorrect in Python

Ans:- In Python, incorrect indentation can lead to syntax errors, logical errors, and reduced code readability. Indentation is vital for defining code structure and scope, such as loops and conditional statements. Syntax errors arise when indentation is inconsistent or incorrect, preventing code execution until fixed. Logical errors may occur due to misinterpreted code structure, resulting in unexpected behavior. Proper indentation enhances code readability, making it easier to understand and maintain. Consistent adherence to Python's indentation rules is essential for error-free code execution and improved code quality.I

6.Differentiate between expression and statement in Python with examples

Ans:- In Python, expressions and statements are fundamental

components of code, but they serve distinct purposes:

1. **Expression**:

   - An expression is a combination of values, variables, operators, and function calls that evaluates to a single value.

   - It represents a computation or operation that yields a result.

   - Expressions can be simple, like a single constant or variable, or complex, involving multiple sub-expressions.

   - Examples of expressions:

     ```python

     x = 5  # Assignment expression

     y = x + 3  # Arithmetic expression

     z = (x * y) / 2  # Complex expression```
     ```

- In Python, expressions can be used within statements to perform calculations, comparisons, or any operation that returns a value.

2. **Statement**:

  - A statement is a complete line of code that performs an action or controls the flow of execution.

  - It represents a complete instruction that the interpreter can Execute.

 - Python statements include assignment statements, control flow statements (if, for, while), function definitions, import statements, etc.

- Examples of statements:

    ```python

    x = 5  # Assignment statement
    ```

```
if x > 0:  # Control flow statement

    print("Positive number") ```
```

- Unlike expressions, statements do not necessarily return a value; they may modify the program's state, control the program flow, or define structures.

  - Python requires at least one statement per line, although a single line can contain multiple statements separated by semicolons (`;`). However, this practice is not common or recommended for clarity reasons.

In summary, expressions are components of statements that produce values, while statements are complete instructions that perform actions or control the program flow in Python.