

Python – Expressions & Variables



Understanding Expressions

Definition:

An expression in Python is a **combination of values, variables, operators, and function** calls that the interpreter can evaluate to produce a result.

Example:

```
5 + 3          # Adds 5 and 3
x * 7          # Multiplies the value of x by 7
(a + b) / 2    # Calculates the average of a and b
len("Hello")   # Finds the length of the string "Hello"
```

Operands:

The values or variables on which operation is performed. In $5 + 3$, the operands are 5 and 3.

Operator:

Symbols that represent specific operations. In $5 + 3$, the operator is +.

Understanding Variables

Definition:

A variable is a **named storage location in memory** used to **hold data that can be updated or reused**.

Example:

```
age = 25          # Assigning an integer value to a variable
name = "Alice"    # Assigning a string value to a variable
pi = 3.14159      # Assigning a floating-point value to a variable
is_active = True  # Assigning a boolean value to a variable
```

Understanding Variables

Rules of Naming Variables:

- Must start with a letter or an underscore (_).

`my_var` or `_temp`

`1var`

- After the first character, you can use letters, numbers, or underscores.
- Variable names are case-sensitive.

`Name` \neq `name`

- Avoid using Python's reserved words like ***if, for, while***, etc., as variable names.

Expression within Variables

```
# Example 1: Storing the result of a simple expression  
result = 5 + 3  
print(result) # Output: 8
```

```
# Example 2: Using variables in expressions  
a = 10  
b = 20  
sum_value = a + b  
print(sum_value) # Output: 30
```

Using Expression in Variable:

- Variables can hold results of any valid expression.
- Expressions make Python programs dynamic and flexible.
- Reusing variables in expressions allows for concise and maintainable code.

Thank You!

