



Python is a high-level, general-purpose programming language. It's known for its clear syntax, readability, and extensive standard library. Python can be used for a wide range of tasks, including web development, data science, machine learning, and scientific computing.

Advantages of Python include:

- **Readability:** Python's simple syntax makes it easy to read and understand, even for beginners.
- **Versatility:** Python can be used for a wide variety of tasks, from web development to scientific computing.
- **Powerful Libraries:** Python has a large and active community of developers, which has resulted in a wealth of powerful third-party libraries.
- **Ease of Use:** Python is relatively easy to learn, even for beginners.

Here's an example of a simple Python program that prints "Hello, World!":

```
print("Hello World")
```

```
Hello World
```

## Rules of assigning a variable¶

1. It should always start with Alphabets
2. It can contains AlphaNumeric
3. It can contains special characters but only \_
4. We cannot use user define function ie print, input
5. It always take latest value
6. It is case sensitive

```
A = 5
a = 10
print(A)
print(a)

5
10

name = input('Name: ')
age = input('Age: ')
qualification = input('Qualification: ')

Name: Zahid Salim Shaikh
Age: 23
Qualification: BE IT
```

# Operators

## 1. ARITHMETIC OPERATORS(+,-,/,\*,//,%,\*\*)

Arithmetic operators are used to perform mathematical operations on numbers. Python supports a variety of arithmetic operators, including addition, subtraction, multiplication, division, modulus, floor division, and exponentiation.

```
# Addition
a = 10
b = 5
print("Addition:",a+b)

# Substraction
a = 10
b = 3
print("Substraction:",a-b)

# Multiplication
a = 5
b = 7
print("Multiplication:",a*b)

# Division
a = 13
b = 2
print("Division:",a/b)

# Floor Division
a = 13
b = 2
print("Floor Division:",a//b)
```

```

# Modulus
a = 11
b = 3
print("Modulus:", a%b)

#Exponentiation
a = 2
b = 7
print("Exponent:", a**b)

Addition: 15
Substraction: 7
Multiplication: 35
Division: 6.5
Floor Division: 6
Modulus: 2
Exponent: 128

```

## 2. ASSIGNMENT OPERATORS(=)

The assignment operator (=) is one of the most fundamental operators in Python. It is used to assign values to variables. In its simplest form, the assignment operator takes two operands: the variable to be assigned to and the value to be assigned.

variable = value

```

age = 23
name = "Zahid"

```

## 3. COMPARISIONAL OPERATOR(==,=>,<=,>,<)

Comparison operators are used to compare two values and determine if they are equal, not equal, less than, greater than, less than or equal to, or greater than or equal to. It can be used to compare a variety of data types, including numbers, strings, and booleans.

```

x = 10

if x == 10:
    print('x is equal to 10')

x is equal to 10

x = 10
y = 20

if y >= x:
    print('y is greater than or equal to x')

y is greater than or equal to x

```

## 4. LOGICAL OPERATORS(AND,OR,NOT)

Logical operators are used to combine two or more Boolean values and produce a single Boolean value as a result. They are essential for creating complex conditional statements and making decisions based on multiple conditions.

1. AND (&): The AND operator returns True only if both operands are True. If either or both operands are False, it returns False.

```
a = 5
print(a>=7 and a<10)

False
```

2. OR (||): The OR operator returns True if either or both operands are True. It returns False only if both operands are False.

```
a = 5
print(a>=5 or a<10)

True
```

3. NOT (not): The NOT operator reverses the Boolean value of the operand. If the operand is True, it returns False. If the operand is False, it returns True.

```
x = True
y = not x
print(y)

False
```

### Learning Progress:

Variable Assignment: I have successfully grasped the concept of variable assignment in Python. I understand how to assign values to variables using the assignment operator (=).

Arithmetic Operators: I have thoroughly explored the operators in Python, including

1. ARITHMETIC OPERATORS(+,-,/,\*,//,%,\*\*)
2. ASSIGNMENT OPERATORS(=)
3. COMPARISON OPERATOR(==,>,<=,>,<)
4. LOGICAL OPERATORS(AND,OR,NOT)

I can effectively utilize these operators to perform various mathematical operations on numeric data.

### Next Steps:

Calculator Development: I am ready to proceed with building a calculator using the fundamental knowledge I have acquired. I will leverage variable assignment to store operands and operator

symbols. I will employ arithmetic operators to perform the desired calculations based on the chosen operator.

## Basic Calculator

```
a = int(input("Enter number 1: "))
b = int(input("Enter number 2: "))

print('Sum of numbers are',a+b)
print('Subtraction of numbers are',a-b)
print('Multiplication of numbers are',a*b)
print('Quotient with decimal',a/b)
print('Quotient without decimal',a//b)
print('Remainder is',a%b)
print('Exponent is',a**b)
```

```
Enter number 1: 20
Enter number 2: 15
Sum of numbers are 35
Subtraction of numbers are 5
Multiplication of numbers are 300
Quotient with decimal 1.3333333333333333
Quotient without decimal 1
Remainder is 5
Exponent is 32768000000000000000
```

## Conditional Statements

In Python, conditional statements, such as 'if' and 'elif', enable programs to make decisions based on specific conditions. They control the flow of execution by directing the program to execute different code blocks depending on the evaluation of the conditions.

### 1. if Statement:

The 'if' statement is the most basic conditional structure, allowing you to execute a block of code if a specified condition is true.

For example, to check if a variable 'age' is greater than 18, you would use:

```
age = int(input("Enter Age: "))

if age > 18:
    print("Adult")
```

```
Enter Age: 23
Adult
```

### 2. if-else Statement:

The 'if-else' statement extends the 'if' statement by providing an alternative block of code to execute if the condition is false.

```
number = int(input("Enter a Number: "))
if number > 0:
    print("The number is positive.")
else:
    print("The number is negative.")

Enter a Number: 7
The number is positive.
```

### 3. elif Statement:

The 'elif' statement allows you to chain multiple conditions within a single 'if' statement.

```
age = int(input("Enter Age: "))
if age <= 12:
    print("You are a child.")
elif age <= 18:
    print("You are a teenager.")
else:
    print("You are an adult.")

Enter Age: 17
You are a teenager.
```

### 1. Nested IF Statement

```
number = 10
if (number >= 0):
    if number == 10:
        print('Number is 10')
    else:
        print('Number is positive')
else:
    print('Number is negative')

Number is 10
```

## Learning Progress:

Conditional Statements: I have thoroughly grasped the concepts of conditional statements in Python, including 'if', 'if-else', and 'elif'. I understand how to use these statements to control the flow of execution based on specified conditions.

Applying Conditional Statements: I have successfully practiced applying conditional statements to solve various programming tasks. I can effectively utilize these statements to make decisions based on user input and different scenarios.

## Next Steps:

- Grade Assignment Program: I am ready to develop a program that assigns grades based on user-entered marks. I will utilize conditional statements and logical operators to evaluate the input marks and determine the corresponding grade.
- Grade Categorization: I will implement the following grade categorization:
  - Marks above 90: A+
  - Marks between 70 and 90: A
  - Marks between 50 and 70: B
  - Marks between 35 and 50: C
  - Marks below 35: F
- User Input Handling: I will incorporate user input handling mechanisms to allow users to enter their marks. I will validate the input to ensure it is a valid numerical value.
- Output Display: I will display the assigned grade to the user.

## Basic Grade Assignment Program

```
marks = int(input("Enter Marks: "))

if marks >= 90 and marks <= 100:
    print("A+ Grade")
elif marks >= 70 and marks < 90:
    print("A Grade")
elif marks >= 50 and marks < 70:
    print("B Grade")
elif marks >= 35 and marks < 50:
    print("C Grade")
elif marks >= 0 and marks < 35:
    print("F Grade")
else:
    print("Invalid Input")
```

```
Enter Marks: 77
A Grade
```