

17/3/2024

PROGRAMMING FOR AI

Lab-12

Name:

ABDUL MOIZ ARSALAN

Reg No.:

B22F1086AI072 (AI22-BLUE)

Submitted To:

MAM ANEELA

Department:

IT AND CS

Program:

BS ARTIFICIAL INTELLIGENCE

Programming for AI Lab

Lab 01

Difference between Assignment Operator and Equality Operator:

The assignment operator and the equality operator serve different purposes in programming:

Assignment Operator:

The assignment operator, denoted by "=", is used to assign a value to a variable. It is used to store a value in a variable.

Example: `x = 5` assigns the value 5 to the variable x.

Equality Operator:

The equality operator, denoted by "==", is used to compare two values to check if they are equal. It returns True if the values are equal, and False otherwise. It is used to compare values in conditions and expressions.

Example: `x == 5` checks if the value of the variable x is equal to 5.

Variables in Python:

There are different types of variables in python. Some of them are given below. First four have one, one example.

- **Integer Variable:** An integer variable holds integer values, which are whole numbers without any decimal point.

```
a = 10
print(a)
print("Type of a:", type(a))
```

- **Floating Point Variable:** A floating-point variable holds numerical values with a decimal point.

```
b = 3.14
print(b)
print("Type of b:", type(b))
```

- **String Variable:** A string variable holds sequences of characters enclosed within single, double, or triple quotes.

```
c = "Hello, World!"
print(c)
print("Type of c:", type(c))
```

- **Boolean Variable:** A Boolean variable holds either True or False values.

```
d = True
print(d)
print("Type of d:", type(True))
```

- **List Variable:** A list variable holds an ordered collection of items. Items can be of any data type, and the list itself is mutable.

- **Tuple Variable:** A tuple variable holds an ordered collection of items, similar to a list. However, tuples are immutable.
- **Dictionary Variable:** A dictionary variable holds key-value pairs where each key is associated with a value. Keys are unique, and the dictionary is mutable.

Single Variable with Multiple Values:

```
a, b, c = 1, 2, 3
```

```
x, y, z = "apple", "banana", "cherry"
```

```
print(a, b, c)
```

```
print(x + y + z) #In case of positive sign there will be no space in between words, letter or numbers
```

Multiple Variables with Single Value:

```
d = e = f = 10
```

```
g = h = i = "hello"
```

```
print(d, e, f)
```

```
print(g)
```

Rules for Variable Name:

In Python, variable names must follow these rules:

- Variable names can contain letters (both uppercase and lowercase), digits, and underscores.
- Variable names must begin with a letter (a-z, A-Z) or an underscore (_).
- Variable names are case-sensitive.
- Variable names cannot be Python keywords or reserved words.
- Variable names should be descriptive and meaningful.
- Variable names should not start with a digit.
- Variable names cannot contain spaces or special characters except for underscore (_).

Here are some of the example of correct and wrong declarations variable names:

Correct Variable Name:

```
my_variable = 10
```

```
variable_2 = 3.14
```

```
_this_is_a_variable = "Hello"
```

```
var123 = True
```

Wrong Variable Name:

```
3variable = 10 # Variable name starts with a digit
```

```
my variable = 5 # Contains space
```

```
@variable = 7 # Contains special character
```

Arithmetic Operation Using Integer Data type and Float Data Type:

Integer Data Type:

```
num1 = 10
```

```
num2 = 5
```

```
addition = num1 + num2
```

```
subtraction = num1 - num2
multiplication = num1 * num2
division = num1 / num2
print("Addition:", addition)
print("Subtraction:", subtraction)
print("Multiplication:", multiplication)
print("Division:", division)
```

Float Data Type:

```
float1 = 3.5
float2 = 2.0
addition_float = float1 + float2
subtraction_float = float1 - float2
multiplication_float = float1 * float2
division_float = float1 / float2
print("Addition:", addition_float)
print("Subtraction:", subtraction_float)
print("Multiplication:", multiplication_float)
print("Division:", division_float)
```

String Literals:

Using single quotes

```
x = 'Hello, World!'
```

Using double quotes

```
y = "Hello, World!"
```

Using double quotes when the string contains a single quote

```
z = "It's a beautiful day."
```

Using single quotes when the string contains double quotes

```
w = 'She said, "Hello!"'
```

```
print(x)
```

```
print(y)
```

```
print(z)
```

```
print(w)
```

Variable Shadowing:

Variable shadowing occurs when a variable declared within a certain scope has the same name as a variable declared in an outer scope. This can lead to confusion and unexpected behavior. It's generally good practice to avoid variable shadowing to maintain code clarity and prevent potential bugs.

Conclusion:

In this lab, we learned about different types of variables in Python, like numbers, text, and true/false statements. We also saw how to give names to these variables and how to do math with

them. We practiced using different symbols for assigning values and checking if things are the same. Finally, we saw how to put words together to make sentences using strings. Overall, we learned how to use variables to store information and do tasks in Python.