Unit:2 ; Class:2

Variables and Data Types

Lecturer: Musfique Ahmed

Variables:

- Variables are used to store and manipulate data in Python.
- You can assign a value to a variable using the assignment operator "=".
- Variables can hold different types of data, such as numbers, strings, lists, etc.

Naming and Using Variables

When you're using variables in Python, you need to adhere to a few rules and guidelines. Breaking some of these rules will cause errors; other guidelines just help you write code that's easier to read and understand. Be sure to keep the following variable rules in mind:

- Variable names can contain only letters, numbers, and underscores. They can start with a letter or an underscore, but not with a number. For instance, you can call a variable message_1 but not 1_message.

- Spaces are not allowed in variable names, but underscores can be used to separate words in variable names. For example, greeting_message works, but greeting message will cause errors.

- Avoid using Python keywords and function names as variable names; that is, do not use words that Python has reserved for a particular pro grammatical purpose, such as the word print. (See "Python Keywords and Built-in Functions" on page 489.)

- Variable names should be short but descriptive. For example, name is better than n, student_name is better than s_n, and name_length is better than length_of_persons_name.

- Be careful when using the lowercase letter l and the uppercase letter O because they could be confused with the numbers 1 and 0

Try It Yourself

Write a separate program to accomplish each of these exercises . Save each program with a filename that follows standard Python conventions, using lowercase letters and underscores, such as simple_message.py and simple_messages.py .

1. Simple Message: Store a message in a variable, and then print that message .

2. Simple Messages: Store a message in a variable, and print that message . Then change the value of your variable to a new message, and print the new message

Data Types:

- Python has several built-in data types, including:
  - Text type: str
  - Numeric types: int, float, complex
  - Boolean type: bool
  - Sequence types: list, tuple, range
  - Mapping type: dict

- Each data type has its own characteristics and usage.

Type: Masala

Type: Liquid

Strings

A *string* is simply a series of characters. Anything inside quotes is considered a string in Python, and you can use single or double quotes around your strings like this:

```
"This is a string."
'This is also a string.'
```

This flexibility allows you to use quotes and apostrophes within your strings:

```
'I told my friend, "Python is my favorite language!"'
"The language 'Python' is named after Monty Python, not the snake."
"One of Python's strengths is its diverse and supportive community."
```

String Operations

- **Concatenation**: Joining strings with +

```
first_name = "John"
last_name = "Doe"
full_name = first_name + " " + last_name
print(full_name)  # Output: John Doe
```

- **Repetition**: Repeating strings with *

```
print("Hello " * 3)  # Output: Hello Hello Hello
```

- **Methods**: Useful string functions

```
print(word.lower())  # Output: python
print(word.upper())  # Output: PYTHON
print(len(word))     # Output: 6
```

- **Indexing**: Accessing characters by position

```
word = "Python"
print(word[0])  # Output: P
print(word[-1]) # Output: n
```

- **Slicing**: Extracting parts of strings

```
print(word[0:3])  # Output: Pyt
```

Adding Whitespace to Strings with Tabs or Newlines

To add a tab to your text, use the character combination \t as shown at ❶:

```
>>> print("Python")
Python
>>> print("\tPython")
    Python
```

To add a newline in a string, use the character combination \n:

```
>>> print("Languages:\nPython\nC\nJavaScript")
Languages:
Python
C
JavaScript
```

You can also combine tabs and newlines in a single string. The string "\n\t" tells Python to move to a new line, and start the next line with a tab. The following example shows how you can use a one-line string to generate four lines of output:

```
>>> print("Languages:\n\tPython\n\tC\n\tJavaScript")
Languages:
    Python
    C
    JavaScript
```

Stripping Whitespace

You can also strip whitespace from the left side of a string using the `lstrip()` method or strip whitespace from both sides at once using `strip()`:

```
>>> favorite_language = ' python '
>>> favorite_language.rstrip()
' python'
>>> favorite_language.lstrip()
'python '
>>> favorite_language.strip()
'python'
```

Number Operations

You can **add (+), subtract (-), multiply (*), and divide (/)** integers in Python. Python uses two multiplication symbols to represent **exponents(**)**

```
>>> 2 + 3
5
>>> 3 - 2
1
>>> 2 * 3
6
>>> 3 / 2
1.5
```

```
>>> 3 ** 2
9
>>> 3 ** 3
27
>>> 10 ** 6
1000000
```

```
>>> 2 + 3*4
14
>>> (2 + 3) * 4
20
```

```
>>> 0.1 + 0.1
0.2
>>> 0.2 + 0.2
0.4
>>> 2 * 0.1
0.2
>>> 2 * 0.2
0.4
```

Try It Yourself

1. **Number Eight:** Write addition, subtraction, multiplication, and
division operations that each result in the number 8 . Be sure to enclose your operations in print statements to see the results . You should create four lines that **look like this:**

   print(5 + 3)

   Your output should simply be four lines with the number 8 appearing once on each line.

2. **Favorite Number:** Store your favorite number in a variable . Then, using that variable, create a message that reveals your favorite number . Print that message

Assignment

Save each of the following exercises as a separate file with a name like name_cases.py .

1. Personal Message: Store a person's name in a variable, and print a mes sage to that person . Your message should be simple, such as, "Hello Eric, would you like to learn some Python today?"
2. Name Cases: Store a person's name in a variable, and then print that per son's name in lowercase, uppercase, and titlecase .
3. Famous Quote: Find a quote from a famous person you admire . Print the quote and the name of its author . Your output should look something like the following, including the quotation marks:

Albert Einstein once said, "A person who never made a mistake never tried anything new."

4. Famous Quote 2: Store the famous person's name in a variable called famous_person . Then compose your message and store it in a new variable called message . Print your message .
5. Stripping Names: Store a person's name, and include some whitespace characters at the beginning and end of the name . Make sure you use each character combination, "\t" and "\n", at least once .

Print the name once, so the whitespace around the name is displayed . Then print the name using each of the three stripping functions, lstrip(), rstrip(), and strip()

Thank You