Python: Language Basics

Connect with me: Youtube | LinkedIn | WhatsApp Channel | Web | Facebook | Twitter

- Download PDF
- To access the updated handouts, please click on the following link: https://yasirbhutta.github.io/ms-excel/docs/basics.html
- Python Playlist on Youtube
- Download Example Code
- Pyton Resources: Books, Websites, Tutorials
- Python Tools
- Python - Quick Guide for Ultimate Python Beginner's

What is Python

- Python is a high-level, general-purpose programming language.
- It is known for its clear syntax, readability, and versatility.
- Python is widely used for `web development`, `data science`, `machine learning`, and `automation`.

Getting Started

- Install Python: Download and install it from https://www.python.org/downloads/.
- Choose a text editor: A program to write code, like Visual Studio Code, Jupyter Notebook, `PyCharm`, or even a simple text editor like `Notepad`.
- Text editor for Android: Pydroid 3 - IDE for Python 3
  - Video: How to: Install Jupyter Notebook on an Android device
- Interactive mode: Experiment with Python directly in your terminal or command prompt using the python command.

> **Important:** Python source code files always use the `.py` extension.

Lesson 1: Python `print` Function

Objectives

- Understand the basic usage of the `print` function.
- Learn how to print different data types.
- Explore advanced `print` function features like formatting and special characters.
- Practice printing in various tasks.

Introduction to `print`

The `print` function is used to output text or variables to the console. or to a file.

Syntax:

```
print(value1, value2, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

Parameters:

- `value1, value2, ...`: The values to be printed. Multiple values can be separated by commas.
- `sep`: (Optional) Specifies how to separate multiple values. Default is a space `' '`.
- `end`: (Optional) Specifies what to print at the end. Default is a newline character `'\n'`.
- `file`: (Optional) Specifies the file where to print. Default is `sys.stdout` (console).
- `flush`: (Optional) Specifies whether to forcibly flush the stream. Default is `False`.

Task 1: Basic Printing

Instructions:

1. Print a simple message.
2. Print multiple items separated by commas.

Examples

```
# Task 1.1: Print a simple message
print("Hello, world!")

# Task 1.2: Print multiple items
print("Hello", "world", 2024)
```

Task 2: Printing Different Data Types

Instructions

1. Print integers, floats, and strings.

Examples

```
# Task 2.1: Print different data types
print(42)
print(3.14159)
print("This is a string")
```

Task 3: Using `sep` and `end` Parameters

Instructions

1. Change the separator between printed items.

2. Change the ending character of a print statement.

Examples

```python
# Task 3.1: Change the separator
print("apple", "banana", "cherry", sep=", ")

# Task 3.2: Change the ending character
print("Hello", end=" ")
print("world!")

# Task 3.3: Print with a custom ending character:
print("Hello", "World", end="!")
```

Task 4: Print Variables

Instructions

- print variables values using print function

```python
# Task 4.1: print a integer variable
x = 5
print(x)

# Task 4.2: print a string variable
message = 'Python is fun'

# print the string message
print(message)
```

Task 5: String Formatting

Instructions

1. Use f-strings (formatted string literals) for the same purpose.

Examples

```python
# Task 5.1: Use f-strings
name = "Ahmad"
age = 30
print(f"My name is {name} and I am {age} years old.")
```

Task 6: Printing Special Characters

Instructions

1. Print a newline character within a string.
2. Print a tab character within a string.

Examples

```python
# Task 6.1: Print a newline character
print("Hello\nWorld")

# Task 6.2: Print a tab character
print("Hello\tWorld")
```

Task 7: Printing to a File

Instructions

1. Print a message to a text file instead of the console.

Examples

```python
# Task 7.1: Print to a file
with open("output.txt", "w") as file:
    print("Hello, file!", file=file)
```

Practice Exercises

Exercise 1: Print Your Favorite Quote

Print your favorite quote, ensuring proper formatting.

Exercise 2: Create a Simple Receipt

Print a simple receipt with items and prices, properly aligned using tab characters.

Exercise 3: Use Variables in Print Statements

Create variables for your name, age, and favorite hobby, then print a sentence using these variables.

Exercise 4: Output to a File

Write a program that prints a summary of your week (e.g., tasks completed, hours worked) to a text file.

How To Use Print() Function in Python

-

Here is the basic syntax of the print() function:

```
print(object1, object2, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

object1, object2, ...: The objects to be printed. These can be strings, numbers, variables, or any other Python object. sep: The separator to use between objects. The default separator is a space. end: The character or string to print at the end of the output. The default is a newline character (\n). file: The file to write the output to. The default is the console. flush: Whether to flush the output buffer immediately. The default is False.

Video: Use of print() function in python

Example #1:

```
message = 'Python is fun'

# print the string message
print(message)
```

Output:

```
Python is fun
```

Example #2:

```
# Print a string:
print("Hello, World!")

# Print a number:
print(10)

# Print a variable:
x = 5
print(x)

# Print multiple objects on the same line:
print("Hello", "World")

# Print multiple objects on separate lines:
print("Hello")
print("World")

# Print with a custom separator:
print("Hello", "World", sep=", ")
```

```
# Print with a custom ending character:
print("Hello", "World", end="!")
```

Task 1: Printing Variables

```
message = "How are you?"
print(message)
```

Task #1: 100 times "hello world" without loop

related video: video: 100 times "hello world" without loop

Task #2: How to print multiple lines

related video: How to print multiple lines

Task: 3 Save text in file

```
#Print to a file:
with open("output.txt", "w") as f:
    print("Hello, World!", file=f)
```

-
- Video: 100 times "hello world" without loop

Comments

- Comments are important for making code more readable and understandable, especially for other programmers who may need to read or modify the code.
- Comments in Python are non-executable lines of code and ignored by the Python interpreter when the code is executed.

There are two main types of comments in Python:

- **Single-line comments:** These comments start with the hash symbol (#) and extend to the end of the line.

```
# This is a single-line comment
print("Hello, World!")
```

- **Multi-line comments:** These comments are enclosed in triple quotes (""" or ''').

```python
"""
This is a multi-line comment.
It can span multiple lines of code.
"""
print("Hello, World!")
```

See also:

- Video: A Comprehensive Guide to Single Line & Multi-Line Comments

Indentation

Indentation is a very important concept in Python. It refers to adding white space before a statement to a particular block of code. In another word, all the statements with the same space to the right, belong to the same code block.

For example, consider the following code snippet:

```python
if True:
    print("True")
else:
    print("False")
```

See also:

- Indentation in Python - geeksforgeeks.org
- Indentation in Python (With Examples) - askpython.com

True/False (Mark T for True and F for False)

Multiple Choice (Select the best answer)

> Which of the following is the correct syntax for the print statement in Python?

1. □ print ("text")
2. □ println ("text")
3. □ echo ("text")

> What will be the output of the following code?

```python
print("Hello, world!")
```

1. □ Hello
2. □ world
3. □ Hello, world!
4. □ There will be no output.

> How can you print multiple values on a single line in Python?

1. □ Use commas to separate the values within the print statement.
2. □ Use semicolons to separate the values within the print statement.
3. □ Use the + operator to concatenate the values before printing.
4. □ Create a list of the values and print the list.

> Which of the following statements will print the value of the variable x?

1. □ print(x)
2. □ print "x"
3. □ println(x)
4. □ echo x

> What is the purpose of the sep argument in the print function?

1. □ To specify the separator between multiple values printed on the same line.
2. □ To specify the end character for the printed line.
3. □ To specify the file to which the output should be printed.
4. □ To specify the format of the output.

> What is the purpose of the end argument in the print function?

1. □ To specify the separator between multiple values printed on the same line.
2. □ To specify the end character for the printed line.
3. □ To specify the file to which the output should be printed.
4. □ To specify the format of the output.

> How can you print a string without a newline character?

1. □ print(string, end="")
2. □ print(string, sep="")
3. □ print(string + "")
4. □ print(string; "")

Comments:

> What is the primary purpose of comments in Python code?

1. □ To execute instructions for the computer
2. □ To temporarily disable lines of code
3. □ To make the code more readable and understandable for humans
4. □ To create errors for debugging

> Which of the following is the correct syntax for a single-line comment in Python?

1. □ // This is a comment
2. □ /* This is a comment */
3. □ # This is a comment
4. □ { This is a comment }

> How can you create a multi-line comment in Python?

1. □ Using triple single quotes (''')
2. □ Using triple double quotes (""")
3. □ Using backslash () at the end of each line
4. □ Using the comment keyword

> What happens when you run code that includes comments?

1. □ The comments are executed along with the code.
2. □ The comments are ignored by the Python interpreter.
3. □ The comments are displayed as output.
4. □ The comments are converted into machine code.

Exercises

Review Questions

References and Bibliography

- Indentation in Python - geeksforgeeks.org
- Indentation in Python (With Examples) - askpython.com