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.

Video: Use of print() function in python

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

```
# 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

```
# 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 4.3: print a string variable
message = "How are you?"
print(message)
```

Task 5: String Formatting

Instructions

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

Examples

```
# 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

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

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

```
# Task 6.3: Print a tab character

print("Name\tAge\tCity")
print("Alice\t30\tNew York")
print("Bob\t25\tLos Angeles")
```

Output:

```
Name    Age    City
Alice   30     New York
Bob     25     Los Angeles
```

In this example, `\t` is used to align the columns of text.

In Python, the `\t` character is a special escape sequence that represents a horizontal tab. When used in the `print` function or any other string operation, it inserts a tab space in the output. This can be particularly useful for formatting text to make it more readable.

`\t` can be combined with other string manipulation techniques, such as f-strings

```
# Task 6.4: Print a tab character with f-strings
```

```
name = "Alice"
age = 30
city = "New York"

print(f"{name}\t{age}\t{city}")
```

Task 7: Printing to a File

Instructions

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

Examples

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

When you use the instruction with open("output.txt", "w") as file in Python, the file is created in the current working directory of your program. On an Android system, this could be different depending on the environment where the code is executed (e.g., a specific app's data directory, a shared storage location, etc.).

To determine the exact path, you can use the os module to get the current working directory:

```
import os

print(os.getcwd())
```

Regarding file closure, when you use the `with` statement to open a file, Python automatically takes care of closing the file for you once the block of code under the `with` statement is executed. There is no need to explicitly close the file; it is done automatically when the block is exited. This is one of the benefits of using the with statement for file operations.

Task 8: Understanding Syntax Errors in Python

Objective:

Learn about syntax errors in Python by examining and correcting a sample code snippet.

Instructions:

1. **Review the Code Snippet**: Look at the provided Python code and identify any syntax errors.
2. **Identify the Error**: Understand what a syntax error is and why it occurs.
3. **Correct the Code**: Fix the syntax error in the code snippet.
4. **Explanation**: Write a brief explanation of what the syntax error was and how you corrected it.

Code Snippet:

```
print("Hello World!"
```

Steps:

1. **Review the Code Snippet**: Look carefully at the code snippet above.
2. **Identify the Error**:
    - A syntax error occurs when the Python interpreter finds code that does not conform to the rules of the Python language.
    - The provided code snippet has a syntax error because it has an unmatched parenthesis.
3. **Correct the Code**:
    - To fix the error, ensure all parentheses are properly closed.
    - The corrected code should look like this:

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

4. **Explanation**:
    - **Syntax Error**: The error was due to a missing closing parenthesis.
    - **Correction**: Adding the closing parenthesis at the end of the print statement fixes the syntax error.

Syntax error:

- A syntax error in programming occurs when the code violates the rules of the programming language's syntax.
- This means that the code's structure and commands do not conform to the expected format that the interpreter or compiler requires to successfully read and execute the code.

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.

Exercise 5: 100 times "hello world" without loop

related video: 100 times "hello world" without loop

Exercise 6: How to print multiple lines

Related Video: How to print multiple lines

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 ''').

```
"""
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

In Python, indentation refers to the use of whitespace (spaces or tabs) to denote block-level structure in the code. Python uses indentation to define the scope of code blocks, such as:

- Function definitions
- Loops (for, while)
- Conditional statements (if, elif, else)
- Exception handling (try, except)

In Python, indentation is mandatory and is used to determine the grouping of statements. The number of spaces used for indentation is not fixed, but it's standard to use 4 spaces for each level of indentation. Read

more: Indentation - PEP 8 – Style Guide for Python Code

Here's an example:

```
if True:
    print("Hello")  # This line is indented with 4 spaces
    print("World")  # This line is also indented with 4 spaces
```

In this example, the two print statements are indented with 4 spaces, indicating that they are part of the if block.

Python's indentation system has several benefits, including:

- Improved readability: Indentation makes the code structure clear and easy to read.
- Reduced errors: Indentation helps avoid errors caused by mismatched braces or parentheses.
- Simplified syntax: Python's indentation system eliminates the need for explicit block delimiters like braces or keywords.

Another example, consider the following code snippet:

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

Task: Please correct the following Python code:

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

Error message: `IndentationError: unexpected indent"`

See also:

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

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

1. An indentation error in Python is considered a syntax error.

Answer Key (True/False):

1. True

Multiple Choice (Select the best answer)

print function

> 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?

```
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; "")

What is the output of the following code?

name = "Alice" age = 30 print("My name is", name, "and I am", age, "years old.")

a) My name is Alice and I am 30 years old. (Correct) b) My name is Aliceand I am 30years old. (No separation) c) Alice 30 (Values printed without labels) d) An error (Incorrect syntax)

3. How can you format a string in Python to insert variables directly within it?

a) Using string concatenation with the + operator (Limited control) b) Using the format method (Less readable for complex formatting) c) Using f-strings (Correct) d) All of the above (f-strings are generally preferred)

Which of the following statements is NOT valid for the print function in Python?

a) print("Hello, world!") b) print(5 + 3) (prints the result of the expression) c) print() (prints an empty line) d) print(x, y, sep=",") (prints x and y with a comma separator)

How can you prevent a newline character from being printed at the end of the output in Python?

a) By using a semicolon at the end of the print statement (this is not valid Python syntax) b) Using the end argument within the print function and setting it to an empty string ("") c) Specifying a special flag in the function call d) There's no way to suppress the newline character

What is a syntax error in Python?

A) An error caused by incorrect logic in the code. B) An error detected when the code violates the rules of the Python language. C) An error that occurs during runtime. D) An error caused by a variable not being defined.

Identify the error in the following code snippet:

```
prin("Hello World!")
```

A) Incorrect variable name B) Missing colon C) Misspelled keyword D) Missing parenthesis

Hint NameError: name 'prin' is not defined.

What should be done to correct the syntax error in the following code?

```
print("Hello World!"
```

A) Add a closing quotation mark. B) Add a closing parenthesis. C) Add a colon at the end. D) Indent the line correctly.

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.

Indentation:

> What is the purpose of indentation in Python?

1. ☐ To define code blocks
2. ☐ To define functions
3. ☐ To define variables
4. ☐ To print output

Answer: a) To define code blocks

> Which of the following is a correct way to indent code in Python?

- a) Using tabs
- b) Using spaces
- c) Using both tabs and spaces
- d) Using neither tabs nor spaces

Answer: b) Using spaces (Python recommends using 4 spaces for indentation)

> What happens if you don't indent your code in Python?

- a) It will run correctly
- b) It will throw a syntax error
- c) It will print an error message
- d) It will ignore the code

Answer: b) In Python, improper indentation specifically results in an IndentationError. While a syntax error is a broad category for any error in the syntax, an IndentationError is a specific type of syntax error related to incorrect indentation.

1. What is the indentation error in the following code?

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

- a) Missing indentation
- b) Extra indentation
- c) Incorrect indentation
- d) No error

Answer: b) Extra indentation (the second print statement has extra indentation)

> What is the standard number of spaces used for indentation in Python? a) 2 spaces b) 4 spaces c) 6 spaces d) 8 spaces

Answer: b) 4 spaces

> Which of the following code snippets is correctly indented?

```
a) if True:
  print("True")
    print("False")

b) if True:
    print("True")
    print("False")

c) if True:
     print("True")
  print("False")

d) if True:
  print("True")
      print("False")
```

Answer: b)

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

> In Python, indentation is used to define _____ in the code.

- A) Loops
- B) Functions
- C) Classes
- D) Blocks of code

Exercises

Review Questions

References and Bibliography

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