Python Basics - Your First Python Program

Python Basics - Your First Python Program

In this course you'll:

- Write your first Python script
- Learn what happens when you run a script with an error
- Learn how to declare a variable and inspect its value
- Learn how to write comments

Python Basics -Your First Python Program

Along the way you'll also learn about:

- The differences between Interactive and Script windows in IDLE
- The Read-Evaluate-Print Loop (REPL)
- How to create a script and run it
- The Assignment Operator
- Rules for valid variable names
- Some of the standards outlined in PEP 8

Table of Contents

1. **Overview**
2. Writing a Python Script
3. Let's Make Mistakes!
4. Creating Variables
5. Leaving Yourself Helpful Notes
6. Summary and Additional Resources

Table of Contents

1. Overview
2. **Writing a Python Script**
3. Let's Make Mistakes!
4. Creating Variables
5. Leaving Yourself Helpful Notes
6. Summary and Additional Resources

Writing a Python Script

- IDLE - Python's built-in
  Integrated **D**evelopment and **L**earning **E**nvironment
- Two main windows
  - The Interactive Window
  - The Editor Window

The Interactive Window

Executing Python interactively works as a **loop** with three steps:

1. Python **reads** the code entered at the prompt
2. Python **evaluates** the code
3. Python **prints** the result and waits for more input

- This is commonly referred to as a **r**ead-**e**valuate-**p**rint **l**oop
- Abbreviated to REPL
- Python programmers often refer to this Python shell as the **Python REPL**
- or just “the REPL” for short

The Editor Window

You'll write your Python files using IDLE's editor window

- Open the editor window by selecting File > New File

The Editor Window

Before you run your program, you need to save it:

- Select File > Save from the menu
- Save the file as `hello_world.py`

Table of Contents

1. Overview
2. Writing a Python Script
3. **Let's Make Mistakes!**
4. Creating Variables
5. Leaving Yourself Helpful Notes
6. Summary and Additional Resources

Let’s Make Mistakes!

Syntax Errors:

- A syntax error occurs when you write code that isn’t allowed in the Python language

Let’s Make Mistakes!

Runtime errors:

- In the interactive window inside IDLE it catches syntax errors before a program starts running
- In contrast, **runtime errors** only occur while a program is running

Let’s Make Mistakes!

Tracebacks:

- When an error occurs, Python stops executing the program and displays several lines of text called a traceback
- The traceback shows useful information about the error
- Tracebacks are best read from the bottom up

Let’s Make Mistakes!

Review Exercises:

1. Write a program that IDLE won’t run because it has a syntax error.
2. Write a program that crashes only while it’s running because it has a runtime error.

Table of Contents

1. Overview
2. Writing a Python Script
3. Let's Make Mistakes!
4. **Creating Variables**
5. Leaving Yourself Helpful Notes
6. Summary and Additional Resources

Creating Variables

In Python, Variables are names that can be assigned a value and then used to refer to that value throughout your code.

Creating Variables

Variables are fundamental to programming for two reasons:

- Variables keep values accessible
- Variables give values context

Creating Variables

The Assignment Operator:

- An operator is a symbol, such as `+`, that performs an operation on one or more values
- Values are assigned to variable names using a special symbol called the assignment operator `=`
- The `=` operator takes the value to the right of the operator and assigns it to the name on the left

Creating Variables

Rules for Valid Variable Names:

- May contain uppercase and lowercase letters (`A-Z` , `a-z`)
- Digits (`0-9`)
- Underscores (`_`)
- Cannot begin with a digit

Creating Variables

Valid Python names:

- `string1`
- `_a1p4a`
- `list_of_names`

Invalid Python names:

- `9lives`
- `99_balloons`
- `2beOrNot2Be`

Creating Variables

Python variable names may contain Unicode characters:

- **Unicode** is a standard for digitally representing characters used in most of the world's writing systems.

Examples:

- Decorated letters like `é` and `ü`
- Chinese, Japanese, and Arabic symbols `民` `ぢ` `ט`
- Not every system can display these characters

Python Variable Naming Conventions

Descriptive Names Are Better Than Short Names

- ```
  s = 3600
  ```
- ```
  seconds = 3600
  ```
- ```
  seconds_per_hour = 3600
  ```
  ```

Python Variable Naming Conventions

What “case” should you use?

- mixedCase
  - numStudents, listOfNames, secondsPerHour
- lower_case_with_underscores (also referred to as snake_case)
  - num_students, list_of_names, seconds_per_hour
  - In Python, it’s more common to write variable names in this style

Python Variable Naming Conventions

PEP 8 - Widely regarded as the official **style guide** for writing Python

- For more information check out: pep8.org
- Following the standards outlined in PEP 8 ensures that your Python code is readable by most Python programmers
- PEP stands for **P**ython **E**nhancement **P**rotocol
- A PEP is a design document used by the Python community to propose new features or changes to the language

Inspecting Values in the Interactive Window

To inspect the value of a variable in the Interactive Window

- Type the name of the variable by itself at the prompt
- You may see different output displayed compared to using `print()`
- You can also learn more about a variable by using `type()`

Creating Variables

Review Exercises:

1. Using the interactive window, display some text using print().
2. Using the interactive window, assign a string literal to a variable. Then print the contents of the variable using the print() function.
3. Repeat the first two exercises using the editor window.

Table of Contents

1. Overview
2. Writing a Python Script
3. Let's Make Mistakes!
4. Creating Variables
5. **Leaving Yourself Helpful Notes**
6. Summary and Additional Resources

How to Write a Comment

Two ways to create a comment in your code:

- Block Comment - Starts with a `#` on a new line

```
# This is a block comment.
greeting = "Hello, World"
```

- Inline Comment - Continue on the same line, with a `#` just after the code

```
print(greeting) # This is an inline comment
```

How to Write a Comment

To **comment out** lines of your code:

- Place a `#` at the beginning of a line of code
- Non-destructive way to test the behavior of your program without specific lines of code

How to Write a Comment

- To comment out a section of code in IDLE, higlight the line(s) and press:
  - Windows: `Alt` + `3`
  - macOS: `Ctrl` + `3`
  - Ubuntu Linux: `Ctrl` + `D`
- To remove comments, higlight the line(s) and press:
  - Windows: `Alt` + `4`
  - macOS: `Ctrl` + `4`
  - Ubuntu Linux: `Ctrl` + `Shift` + `D`

PEP 8 Comment Recommendations

- Comments should always be written in complete sentences
- Have a single space between the `#` and the first word of the comment
  - ```
    # This comment is formatted to PEP 8.

    #this one isn't
    ```
- Inline comments should start with two spaces after the code
  - ```
    phrase = "Hello, World"  # This comment is PEP 8 compliant.
    print(phrase)# This one isn't
    ```

PEP 8 Comment Recommendations

PEP 8 also recommends that comments be used sparingly

```
# Print "Hello, World"
print("Hello, World")
```

- Comments that describe what is already obvious are unnecessary
- This is a pet peeve of many programmers

Table of Contents

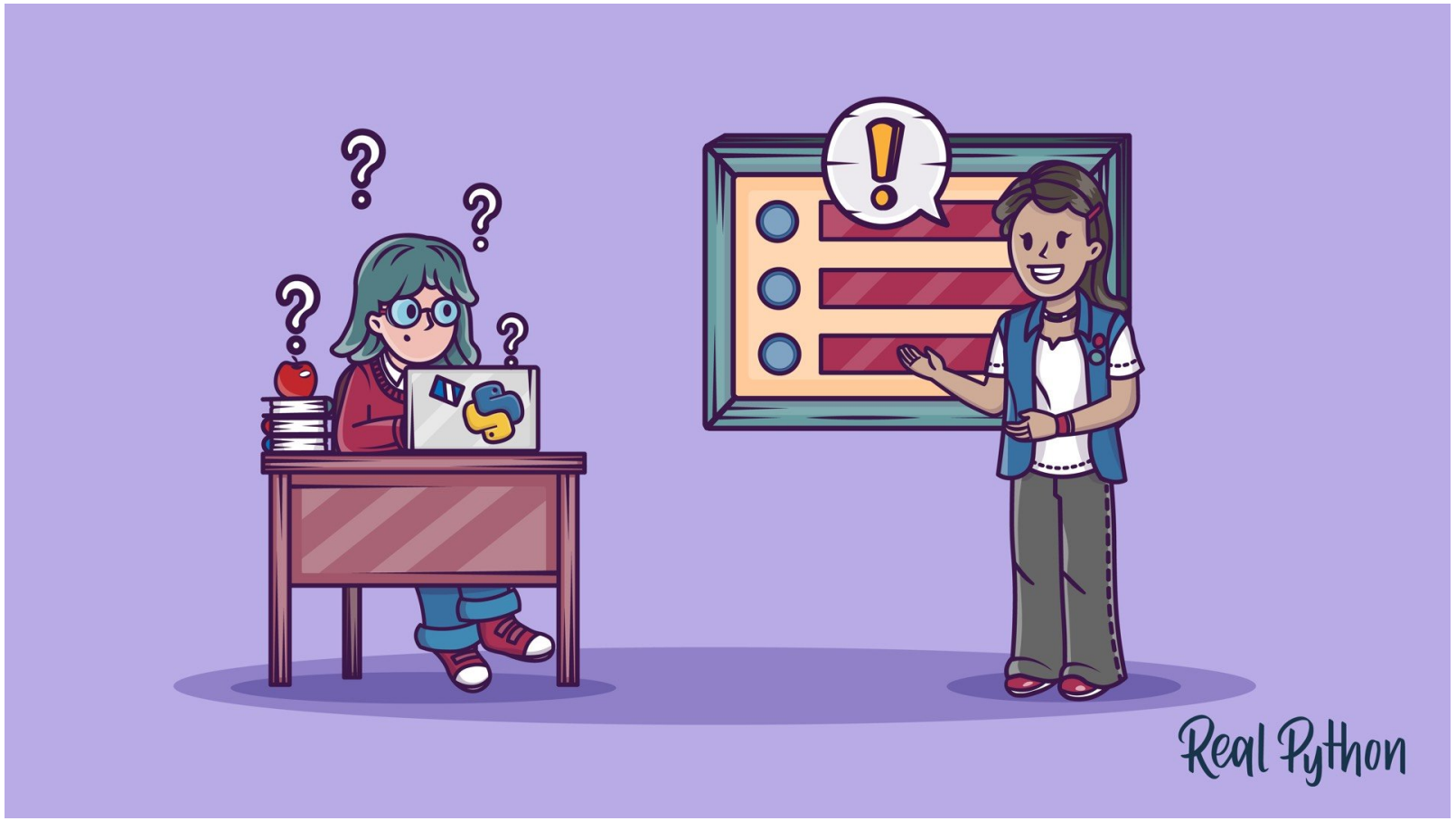
1. Overview
2. Writing a Python Script
3. Let's Make Mistakes!
4. Creating Variables
5. Leaving Yourself Helpful Notes
6. **Summary and Additional Resources**

Summary

You were introduced to several concepts in this course:

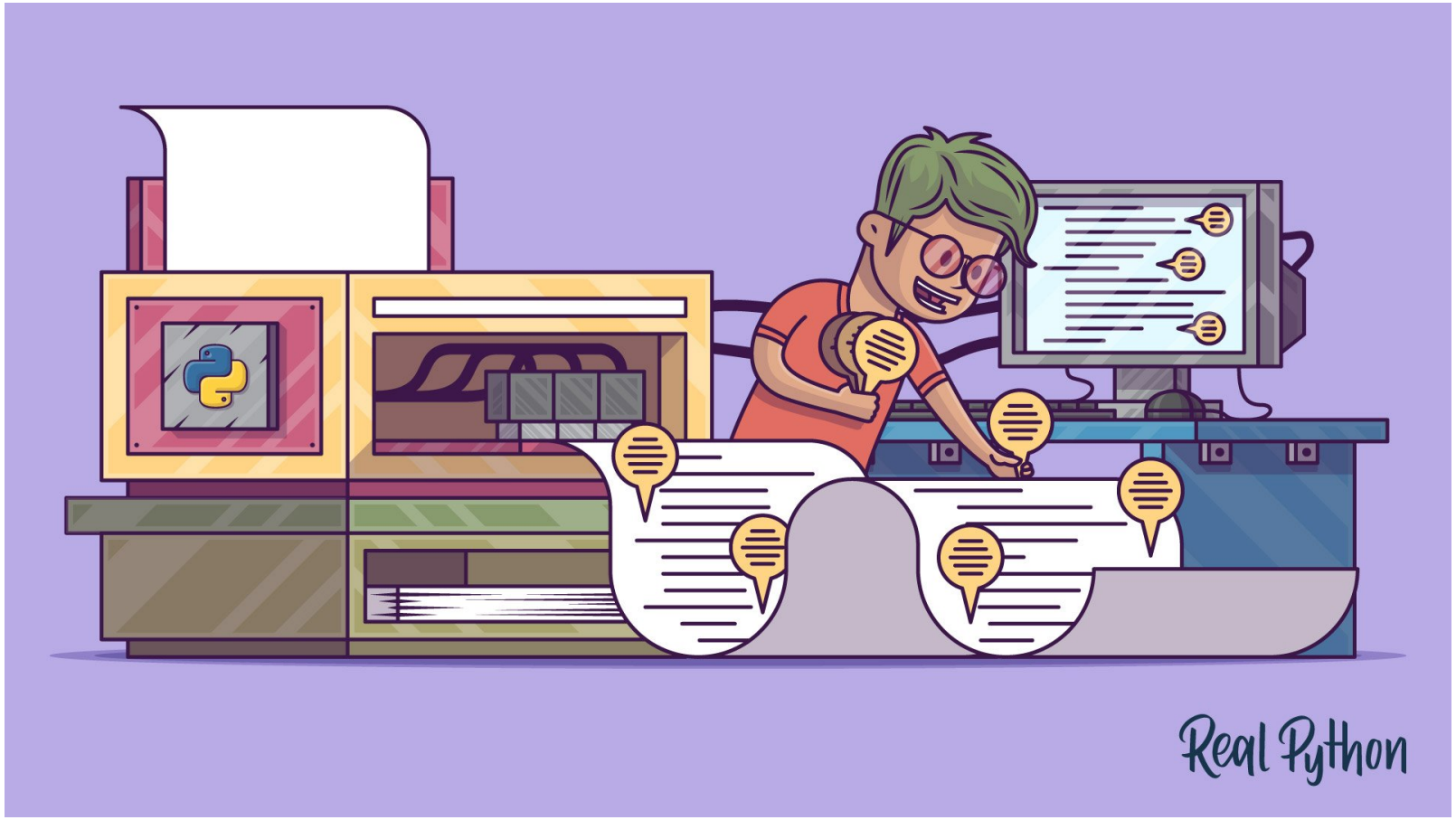
- Variables give names to values in your code
- Use the assignment operator (=) to assign values to a variable
- Syntax errors occur when you write code not allowed in the Python language
- Run-time errors occur while a program is running
- Tracebacks show useful information about an error and are best read from the bottom up
- Comments are lines of code that don't get executed and serve as documentation

11 Beginner Tips for Learning Python Programming



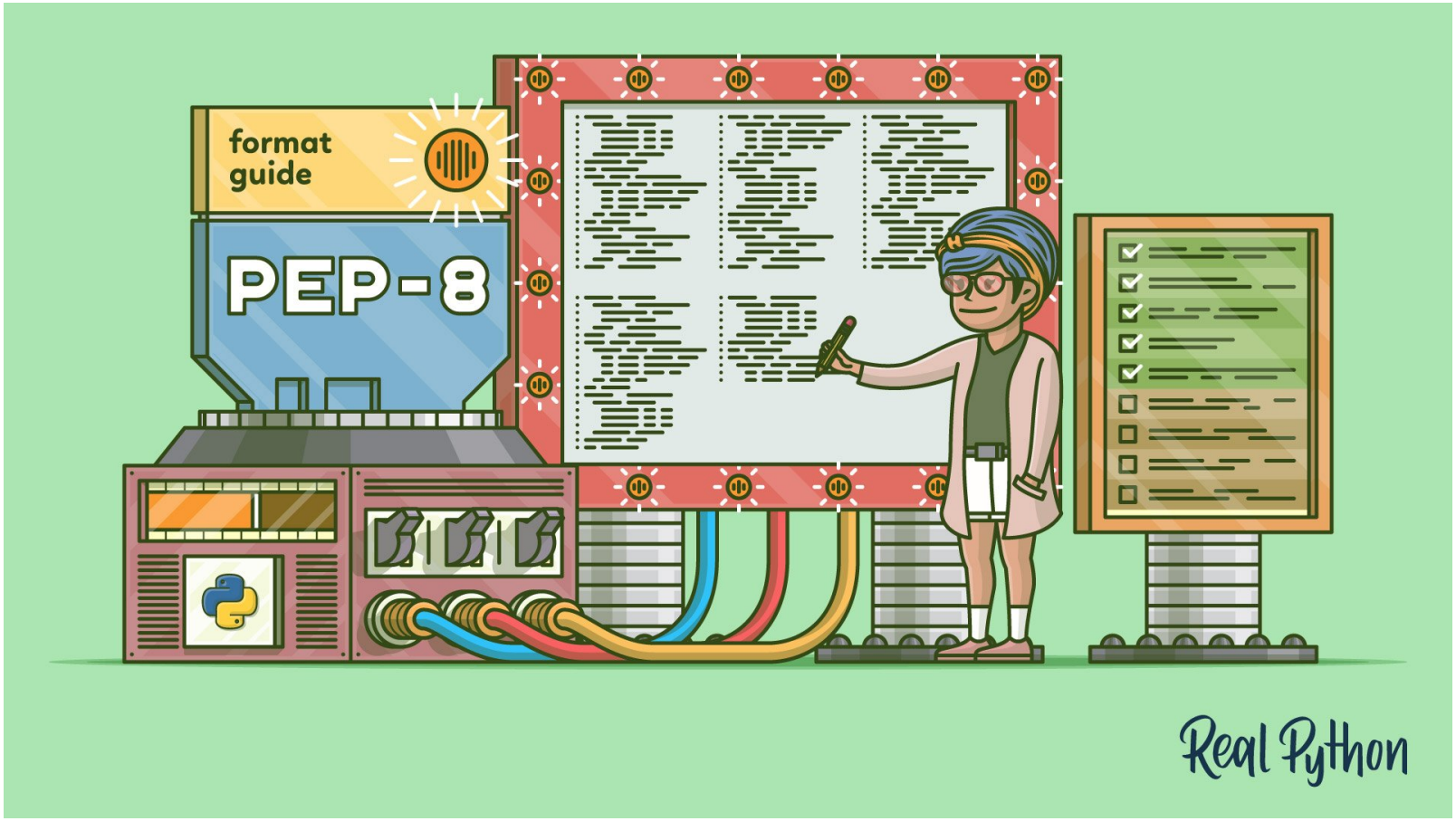
https://realpython.com/python-beginner-tips/

Writing Comments in Python (Guide)



https://realpython.com/python-comments-guide/

How to Write Beautiful Python Code With PEP 8



https://realpython.com/python-pep8/

Python Basics: Chapter 03 - Quiz

https://realpython.com/quizzes/pybasics-first-program/

Congratulations!