



Hello, World!

Harry Smith



Introduction to Cudio

This is a "coding" demo.

Hello, World! & Running Your First Program

Our First Program

- A new programmer's first program by tradition: Hello, World
- This **Python program** will **print a message** when **executed**.
 - "Python program" → computer code written using the Python programming language
 - "print a message" → display some text on the screen
 - "executed" → run a program so that its instructions are followed by the computer, line by line

Naming the File

To write a program, put all code in a single file. For our first program, we'll choose the filename `hello_world.py`

- Use descriptive names for files
- We use `snake_case` to name things in Python
 - Words start with lowercase letters
 - Words separated by underscore (`_`) characters
- End Python files with the extension `.py`

Hello, World!

Our file, `hello_world.py`:

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

- Just one line! (The beauty of Python: simplicity.)

To run in the terminal:

- write `python hello_world.py`
- hit enter/return:

```
$ python hello_world.py  
Hello, world!
```

Printing: displaying text on screen

- `print()` function
 - Whatever you put inside of the parentheses gets displayed as output.
- Examples:
 - `print("Hello, world!")` prints `Hello, world!`.
 - `print("So long...")` prints `So long...`.
 - `print("🙈🙊🙉")` prints `🙈🙊🙉`.

Text and Strings

- A **string** is a sequence of characters that is interpreted literally.
 - Surround text in quotes to create a **string**
 - Without quotes, text is interpreted as a statement or an expression—that is, as another line of code.

Bug: Forgetting Quotes

If we remove the quotes from `hello_world.py`, we have an example of *buggy code*:

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

Now, when run, we get an *error message*:

```
$ python hello_world.py
File "/python-book/programs/hello_world/hello_world_quoteless.py", line 1
    print(Hello, world!)
           ^
SyntaxError: invalid syntax
```

Structure of a Program

The image features a dark brown background. At the bottom, there is a large, light brown, mountain-like shape that tapers to a point at the top. The title "Structure of a Program" is written in a bold, white, sans-serif font across the upper portion of the image, partially overlapping the dark background and the top of the light brown shape.

Statements & Order of Execution

- Each line of code represents a **statement**, which is a full instruction to be handled by the computer.
- Default program execution: top to bottom, one at a time

Statements & Order of Execution

hello_everybody.py

```
print("Hello to friends.")  
print("Hello to family.")  
print("Hello to fans.")  
print("Hello to you.")
```

Output:

```
$ python hello_everybody.py  
Hello to my friends.  
Hello to my family.  
Hello to my fans.  
Hello to you.
```

Comments

- Use `#` for comments
- Ignored by the computer
 - Doesn't influence the behavior of the program at all!
- For humans: explanations, notes, TODOs

Example: Commenting `hello_world.py`

```
# This is a comment  
print("Hello, world!") # Prints a message
```

Reading User Input

Reading User Input

- `input()` function for getting text interaction
 - Like `print()`, any string placed in the parentheses will be first displayed as a prompt.
 - Whatever the user types is saved for later inside of `name`

Reading User Input: Example

hello_input.py

```
print("Who would you like to say hello to?")  
name = input(">")  
print("Hello, ", name)
```

Running `hello_input.py`

First execution:

```
$ python hello_input.py  
Who would you like to say hello to?  
> Agustina  
Hello, Agustina
```

Next execution, different name:

```
$ python hello_input.py  
Who would you like to say hello to?  
> Carson  
Hello, Carson
```