

Business Programming (using Python)

Zhaohu (Jonathan) Fan

Information Technology Management

Scheller College of Business

Georgia Institute of Technology

August 31, 2023

Main topics

- Go over some of the Severance Chapter 2 concepts (plus some)
- Debugging
 - Errors & Exceptions
 - Try & Except

Survey question

Regarding the integration of project-based learning in this course, which of the following assessment structures do you prefer?

- A. **Maintain Current Structure**: Retain the **final project (20% of grade)** and final exam (20% of grade) as separate components.
- B. **Combined Approach**: Merge the final project and final exam into a unified **project presentation (20% of grade)** and final project deliverable (20% of grade).

Survey question

- **Plan A: A final exam** or **Plan B: A project presentation** ?
 - **Plan A** (maintain current structure) or **Plan B** (combined approach)?

Errors & Exceptions

Errors & exceptions

- When facing an error message, and the error type is unclear, you should do some research on it.
 - There are two main types of errors: 1) **syntax errors**, and 2) exceptions.

Syntax errors

Output

```
Hello world
```

(Incorrect) code

```
# if true print('Hello world')
```

Syntax errors

Output



The screenshot shows a Python interpreter window with a red error icon in the top left. The code being executed is `if true print('Hello world')`, which is underlined in red. Below the code, the error message reads: "File "<ipython-input-1-ac8bff3a995f>", line 1" followed by the code snippet `if true print('Hello world')` with a red caret under the `print` keyword. Below this, the text "SyntaxError: invalid syntax" is displayed. At the bottom of the window, there is a button labeled "SEARCH STACK OVERFLOW".

- The parser repeats the offending line and highlights the error.
 - File name and line number are also printed for reference (if multi-line code).
 - In Python, code is generally written **line-by-line**, with **each statement** occupying a single line for readability and simplicity.

what if a statement is too long to fit on a single line?

Code

```
python

# Using backslash
total = item_one + \
        item_two + \
        item_three

# Using parentheses
total = (item_one +
        item_two +
        item_three)
```

- Break the statement into multiple lines using a **backslash** `\` at the end of the if statement line.
 - Alternatively, Python allows line continuation within **parentheses** `()`, **brackets** `[]`, and **braces** `{}` without the backslash.

Syntax errors

Output



The screenshot shows a Python interpreter window with a red error icon and a play button. The code entered is `if true` followed by `print('Hello world')`. The `if true` line is underlined in red. Below the code, the error message reads: `File "<ipython-input-2-e34dab743bad>", line 1`, `if true`, `^`, `SyntaxError: expected ':'`. A "SEARCH STACK OVERFLOW" button is visible at the bottom.

```
if true
print('Hello world')
```

File "<ipython-input-2-e34dab743bad>", line 1
if true
^
SyntaxError: expected ':'

SEARCH STACK OVERFLOW

- It's missing a **colon** `:` at the end of the if statement line.

Syntax errors

Output



The screenshot shows a Python interpreter window with a red error icon in the top left. The code entered is:

```
if True:  
print('Hello world')
```

The error message displayed is:

```
File "<ipython-input-6-10537858c69d>", line 2  
  print('Hello world')  
  ^  
IndentationError: expected an indented block after 'if' statement on line 1
```

Below the error message is a button labeled "SEARCH STACK OVERFLOW".

- In Python, the boolean value `true` should be capitalized as `True`.
 - `print('Hello world')` line is not properly indented to indicate that it belongs to the `if` statement.
 - Make sure to **indent** the print statement properly to **maintain the code structure**.

What does indentation signify in Python?

- **Indentation** *"in Python refers to the leading whitespace—spaces or tabs—at the beginning of a line of code."*
- In Python, **indentation** is used to identify blocks of code that belong together.

Conditional statements if else

- In Python, conditional statements are used for decision-making.
 - The basic structure involves the use of `if`, `elif(else if)`, and `else` keywords

Code `if`

```
# if condition:  
    # Block of code to execute if the condition is true
```

Code `if`, `elif(else if)`, `else`

```
#if condition1:  
    # Execute this block if condition1 is true  
#elif condition2:  
    # Execute this block if condition1 is false and condition2 is true  
#else:  
    # Execute this block if none of the conditions are true
```

Syntax errors

Output

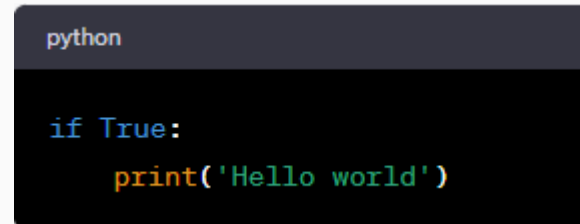


A code execution environment showing a successful run. On the left, there is a green checkmark and a play button icon. Below the play button is a small '0s' indicating execution time. The code being executed is: `if True:
 print('Hello world')`. Below the code, the output 'Hello world' is displayed.

```
if True:  
    print('Hello world')
```

Hello world

The corrected code



A code editor window with a dark background. The title bar says 'python'. The code inside is: `if True:
 print('Hello world')`. The code is color-coded: 'if' is blue, 'True' is blue, ':' is blue, 'print' is orange, and the string is green.

```
python  
  
if True:  
    print('Hello world')
```

In Python, are single quotes (' ') and double quotes (" ") **interchangeable** for defining string literals?

Syntax errors

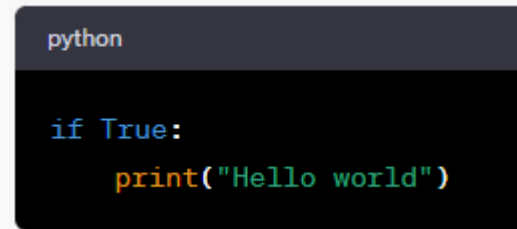
Output



A terminal window showing a successful execution of Python code. On the left, there is a green checkmark and the text '0s'. To the right of this is a play button icon. The code being executed is `if True: print("Hello world")`. Below the code, the output 'Hello world' is displayed.

```
✓ 0s ▶ if True:  
      print("Hello world")  
  
Hello world
```

The corrected code



A terminal window with a dark background. The title bar at the top says 'python'. The code being executed is `if True: print("Hello world")`. The code is color-coded: 'if' is blue, 'True' is blue, ':' is blue, 'print' is orange, and the string is green.

```
python  
  
if True:  
    print("Hello world")
```


Yes, in Python, single quotes (' ') and double quotes (" ") are **interchangeable** for defining string literals.

Exceptions

Exceptions

- Each exception belongs to an Exception type (with a link).
 - TypeError: Raised when an operation or function refers to an object of **inappropriate type**.
 - ValueError: Raised (thrown) when an operation or function refers to an argument that is the right type but has an **inappropriate value**.

Handling Exceptions

- Try & except & finally

Code

```
try:
    result = 10 / 0
except ZeroDivisionError:
    print("Cannot divide by zero!")
finally:
    print("This will run no matter what.")
```

- Code that you suspect may raise an exception is placed inside the `try` block
 - The `finally` block contains code that will always be executed, whether an exception was raised or not. This is often used for cleanup actions, such as closing files.