

Python

**Race Condition
Round Robin Method
Thread**



Race condition

- A **race condition** is a phenomenon that occurs in concurrent programming when the outcome of the program depends on the relative timing or interleaving of multiple threads or processes. It arises when two or more threads or processes attempt to modify a shared resource simultaneously, leading to unpredictable or erroneous behavior.
- **Key characteristics of a race condition:**
 - Shared Resource: A race condition typically involves a shared resource such as a variable, memory location, file, or data structure that is accessed and potentially modified by multiple threads or processes concurrently.
 - Unpredictable Behavior: The behavior of the program becomes unpredictable because the outcome depends on the specific timing of each thread's execution. Small differences in the timing or order of thread execution can lead to different results.
 - Interleaved Execution: Concurrent threads or processes execute in an interleaved manner, meaning that their instructions may be executed in an unpredictable order. This interleaving can result in unexpected interactions between threads accessing the shared resource.

Round Robin scheduling method

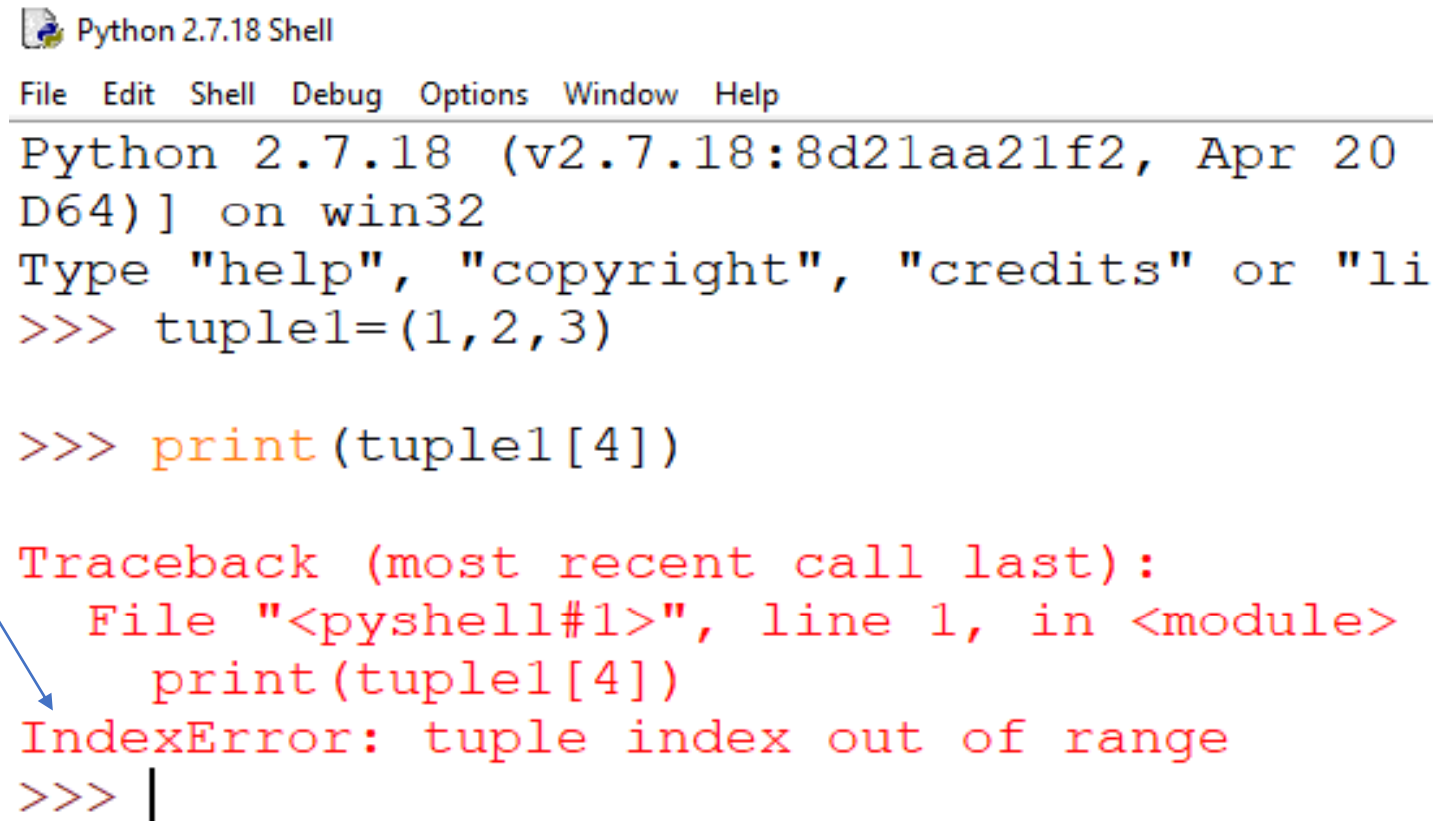
- The term that describes the execution of tasks in a circular manner, such that after completing the last task, the process returns to the first task, is "Round Robin scheduling method." In this scheduling algorithm, each task is assigned a fixed time slice or quantum, and the tasks are executed in a circular order. When a task's time slice expires, it is preempted, and the next task in the queue is given the CPU to execute. This process continues until all tasks are completed or until a specified time limit is reached. Round Robin scheduling is commonly used in operating systems for time-sharing systems and in scheduling algorithms for CPU scheduling.

Thread

- A Threads represent individual units of execution within a process and share the same memory space, making them more lightweight than separate processes.

Exception handling

```
tuple1=(1,2,3)  
print(tuple1[4])
```

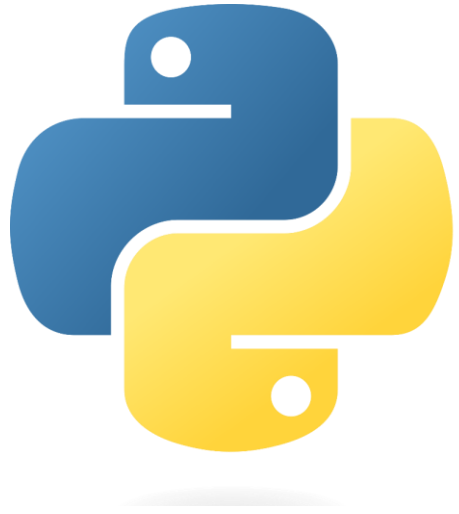


The screenshot shows a Python 2.7.18 Shell window. The title bar reads 'Python 2.7.18 Shell'. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the following content:
Python 2.7.18 (v2.7.18:8d21aa21f2, Apr 20 D64)] on win32
Type "help", "copyright", "credits" or "li
>>> tuple1=(1,2,3)

>>> print(tuple1[4])

Traceback (most recent call last):
 File "<pyshell#1>", line 1, in <module>
 print(tuple1[4])
IndexError: tuple index out of range
>>> |

A blue arrow originates from the `print(tuple1[4])` line in the code block on the left and points to the `print(tuple1[4])` line in the screenshot, indicating the source of the exception.



Python GUI

Tkinter module

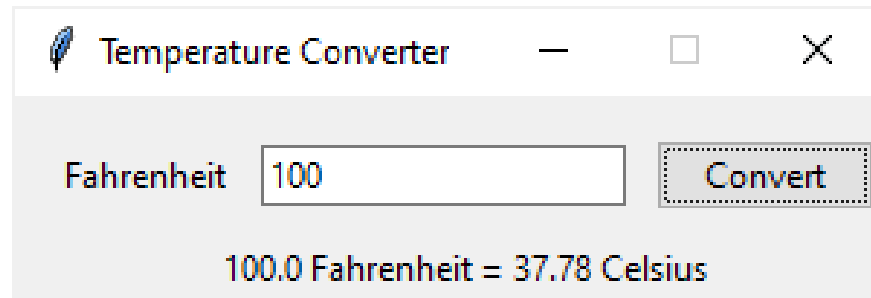


- The primary purpose of using a Canvas-container in Python's Tkinter library
 - To draw shapes like lines, curves, and circles in a GUI.

- Reference to GUI program on LMS!

Temperature converter

temperatureconverter.py on LMS



A screenshot of a graphical user interface for a temperature converter. The window has a title bar with a feather icon, the text "Temperature Converter", and standard window controls (minimize, maximize, close). The main area has a light gray background. It contains a label "Fahrenheit" followed by a text input field containing the number "100". To the right of the input field is a button labeled "Convert". Below these elements, the text "100.0 Fahrenheit = 37.78 Celsius" is displayed.

Temperature Converter

Fahrenheit 100 Convert

100.0 Fahrenheit = 37.78 Celsius

Reference

- <https://www.pythontutorial.net/>
- <https://www.pythontutorial.net/tkinter/tkinter-window/>
- <https://www.pythontutorial.net/tkinter/tkinter-event-binding/>
- <https://www.pythontutorial.net/tkinter/tkinter-command/>
- Data science reference link
- <https://www.datacamp.com/blog/top-python-libraries-for-data-science>