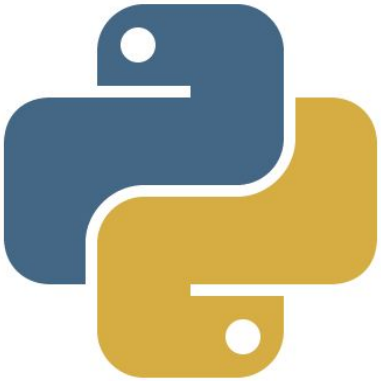

python

TOT - Python Programming Essentials

Day 7

Day 7 Act 1

1. Create a Name Generator App
2. Create 3 lists for first name, middle name, and last name with 10 items per list
3. The application will ask the user to generate a new name.
4. If yes, use a random number between 0 - 9 to randomly select items in the lists
5. Display the generated name “Congratulations! Your new name is _____”
6. If No, display the word “Thank you! ” and display all the names that user generated.

What are Exceptions?

- Events or errors that disrupt the normal flow of execution of a program.
- It prevents the program from reaching a normal end.
- These events or errors usually occur during runtime.
- Example of exceptions:
  - Division by zero
  - Invalid input
  - File not found

Kinds of Exceptions

- **Checked Exception**
  - All exceptions, except for Runtime Exception, are checked exceptions.
  - Exceptions are “checked” because they are subject to the Catch or Specify Requirement. Otherwise, the program code will not compile.
  - Checked exceptions are errors that the program can deal with.

Kinds of Exceptions

- **Errors**
  - Errors are generally beyond the control of the program. These are situations that cannot be anticipated and for which the program cannot recover from.
  - Example:
    - Unreadable file
    - Hardware malfunction
  - Errors are not subject to the Catch or Specify requirement, and are often referred to as unchecked exceptions.

Kinds of Exceptions

- **Runtime Exception**
  - Like errors, Runtime Exceptions are not subject to the Catch or Specify Requirement and are, also, unchecked exceptions.
  - These exceptions are the result of programming flaws such as:
    - dividing by zero,
    - using null pointers or references,
    - or going beyond an array’s boundaries.

Handling Exceptions

Try:

    pass

Except ExceptionName

    pass

Else:

    pass

Finally:

    pass

Try Block

The try block lets you test a block of code for errors.

```
try:
    file = open(‘filename.txt’)
```

Except Block

The Except block will be executed if the try block raises an error.

```
try:
    file = open('filename.txt')
except Exceptions:
    print('Error')
```

Else Block

You can use the else keyword to define a block of code to be executed if no errors were raised:

```
try:
    file = open(‘filename.txt’)
except Exceptions:
    print(‘Error’)
else:
    print(‘No Error’)
```

Finally Block

The finally block, if specified, will be executed regardless if the try block raises an error or not.

```
try:
    file = open('filename.txt')
except Exceptions:
    print('Error')
else:
    print('No Error')
finally:
    file.close()
```

Day 7 Act 2

1. Create a calculator app
2. The user will choose between the 4 math operations (Add, Subtract, Multiply and Divide)
3. The application will ask for 2 numbers
4. Display the result
5. The application will ask again if the user wants to try again
6. Use the appropriate Exception (ex: Invalid input such as text and zero division)

Raise Keyword

Use to execute an exception depending on the condition

```
try:
        a = 15
        If a <=17:
                raise Exception
except Exceptions:
        print('Error')
else:
        print('No Error')
finally:
        file.close()
```