COP 3035
Intro Programming in Python

Summer 2024

Lecture 9 – part 1

Homework 3 - Due date: 06/14/2024
Lab 5 - Due Date: 06/17/2024
Exam 2 : 06/21/2024

Lecture 9 – part 2

Review

Review

Conditional statements: `if, elif, else`

For loops

    For loops with strings

    For loops with lists

    For loops with dictionaries

    Tuple unpacking

    Enumerate()

    Range()

for loops

- **Syntax of a for loop:**

```python
my_iterable = [1,2,3]
for item in my_iterable:
        print(item)
```

Lecture 9 – part 3

range(), enumerate(), zip()

The range function

Built-in function that generates a sequence of numbers.

Syntax:

range(stop)

range(start, stop)

range(start, stop, step)

Examples:

range(5) → 0, 1, 2, 3, 4

range(2, 5) → 2, 3, 4

range(2, 9, 2) → 2, 4, 6, 8

- **Question:** How do you "cast" a range object to a list in Python?

The enumerate function

Built-in function that returns **an iterator** yielding pairs (index, element) for each element in a sequence.

Syntax:

```
enumerate(iterable, start=0)
```

Example:

```
for index, value in enumerate(['a', 'b', 'c']):
    print(index, value)
```

The zip function

Built-in function that **aggregates items** from two or more iterables.

Syntax:
```
zip(*iterables)
```

Example:
```
names = ["Alice", "Bob", "Charlie"]
scores = [85, 92, 88]
for name, score in zip(names, scores):
    print(name, score)
```

Lecture 9 – part 4

While loops

while loops

- While loops continue to execute a block of code **while** some condition remains **True**.

Syntax of the while loop:

```python
while some_condition:
    # Do something
else:
    # Do something different
```

break, continue, pass

break – Breaks out the current closest enclosing loop.

continue – Goes to the top of the closest enclosing loop.

pass – Does nothing at all.

Lecture 9 – part 5

Files

Files

Opening and closing files

```
myFile = open('myFilename','w+')
myFile.close()
# open file arguments: 'r', 'w', 'w+', 'a+'
```

Reading and writing

```
myFile.read():  Read the entire file at offset  position.
myFile.seek(offset):  Change files current position
myFile.readlines():  To read the file line by line
myFile.write('text'):  To write a string to the file
```

A safer way to open the file

To automatically close a file when done:

```
With open('myFilename', mode='r') as myfile:
     # Perform file operations
```

File open modes

Mode	Description
'r'	Open for reading (default). Error if the file does not exist.
'w'	Open for writing. Truncates to zero length or creates a new file.
'x'	Open for exclusive creation. Fails if the file already exists.
'a'	Open for writing, appending at the end if the file exists.
'b'	Binary mode. Use with other modes for binary files (e.g., **'rb'**, **'wb'**).
't'	Text mode (default). Use with other modes for text files (e.g., **'rt'**, **'wt'**).
'+'	Update mode, for reading and writing (e.g., **'r+'**, **'w+'**, **'a+'**).

- **'r+'** : Open for reading and writing without truncating the file. The pointer is at the beginning.
- **'w+'** : Truncates the file to zero length or creates a new one for reading and writing.
- **'a+'** : Open for reading and writing. Appends at the end. Creates the file if it doesn't exist.

Lecture 9 – part 6

Integration Exercise

Exercise: Favorite songs report

```
MY FAVORITE SONGS
-----------------------------------------------------------------
Genre               Artist              Song Title          Duration
-----------------------------------------------------------------
Pop                 Adele               Lady Antebellum     3:48
Electro-pop/dance   LMFAO               *Party Rock Anthem* 4:32
Hip hop/pop         Nicki Minaj         Super Bass          3:20
Indie pop           Foster The People   Pumped Up Kicks     4:00
Country             Lady Antebellum     Just A Kiss         3:38
-----------------------------------------------------------------
The total album duration is: 19 minutes and 18 seconds
-----------------------------------------------------------------
```

Plan:

1. Create a dictionary for every song.

2. Organize the songs in a list (playlist).

3. Set a variable for the favorite song.

4. Iterate over the playlist, print each song, flag the favorite and do computations.

5. Compute the total duration and the final report.