INFO 450 Spring 2021

Week 5

Dicts, Functions, Args and Kwargs

Agenda

- None
- Dicts
- Functions
  - parameters
  - args, **kwargs

None

The sole value of the type NoneType. None is frequently used to represent the absence of a value, as when default arguments are not passed to a function. Assignments to None are illegal and raise a SyntaxError.

https://docs.python.org/3/library/constants.html

Dictionaries

Python dictionaries are a foundational data type in the language

Dictionaries (dict) can be thought of as a 'map' structure

Dictionaries store other values internally, like a list, but…

Values are assigned to keys within the dictionary.

- Can be used to group data together, such as information about a person

Simple Syntax

- Dicts are created with {} as the 'constructor'
- Compare to lists, which are notated as: []

Dicts can store 0 or more 'keys' and their values.

- Keys must be 'hashable' - (strings, numbers, tuple *)
- Values can be 'anything'

Creating a dictionary

syntax:

Keys are paired to their values with colons.

multiple key/values can be set and separated by commas

Modifying data in a dictionary

How do we set key values in a dict?

How do we remove keys from a dict?

Accessing data in a dict

Get a known key (REALLY known, guaranteed, etc. rarely used):

Safely get a key:

Get a key, but if it doesn't exist, get a default value:

Chaining Dictionaries

(not in book that I saw)

Common practice can potentially nest dictionaries. Looking at an example from Spotify API:

https://developer.spotify.com/documentation/web-api/reference/artists/get-artists-albums/

- note: I fudged the structure a little to make it fit my desired example

-

Functions/docs for Dictionaries

https://docs.python.org/3/library/stdtypes.html#typesmapping

- list(d) - Returns a list of all keys
- len(d) - Returns the number of items - (CHF) - number of keys
- key in d: Tests whether the key is present in the d
- k not in d: Tests (inverse) of 'key in d'
- iter(d) - Iterator over the keys in the dictionary
- clear(d) - removes all items from the dict
- copy(d) - shallow copy (doesn't copy values)
- items(d) - returns a view of the data in the items, (key, value) - tuples
- keys(d) - returns a list of keys
- values(d) - returns a list of keys
- … and more

Looping on dicts

https://docs.python.org/3/tutorial/datastructures.html#dictionaries

Generating a unique list of values

Wrap the 'values' with a 'set' command, to convert the 'list' datatype to a 'set'

Set: A set object is an unordered collection of distinct hashable objects.

TO READ:

Read the end of Chapter 6.

Using dicts in a list, using lists in a dict

Functions

Let's deep dive into functions so we can understand the homework assignments and online practices.

- A function is an executable statement.

Defining a function

- The keyword   introduces a function definition.
- It must be followed by the function name and the parenthesized list of formal parameters. The statements that form the body of the function start at the next line, and must be indented.

The first statement of the function body can optionally be a string literal; this string literal is the function’s documentation string, or docstring. (More about docstrings can be found in the section Documentation Strings.)

- There are tools which use docstrings to automatically produce online or printed documentation, or to let the user interactively browse through code; it’s good practice to include docstrings in code that you write, so make a habit of it.

Function to print a help menu

help.py

Yes! You can assign a function 'name' to a variable if you want.

Function with a parameter

- Required parameters are listed in the paranthesis.
- Multiple parameters must be comma separated.

Output of fib

Multiple parameters

Comma separated to allow multiple values to be passed into a function

add_them.py

Optional Parameters

hello_world.py

Mix and Match

parrot.py

Execute parrot.py

Returning values from a function

add_them.py

Empty return statements are essentially None

return_none.py

Returning multiple values

multiple_return.py

Errors and Exceptions

- Syntax Errors
- Exceptions

Syntax Errors

Syntax errors, also known as parsing errors, are perhaps the most common kind of complaint you get while you are still learning Python:

The parser repeats the offending line and displays a little ‘arrow’ pointing at the earliest point in the line where the error was detected.

The error is caused by (or at least detected at) the token preceding the arrow: in the example, the error is detected at the function print(), since a colon (':') is missing before it.

File name and line number are printed so you know where to look in case the input came from a script.

Exceptions

Even if a statement or expression is syntactically correct, it may cause an error when an attempt is made to execute it.

Errors detected during execution are called exceptions and are not unconditionally fatal: you will soon learn how to handle them in Python programs.

Most exceptions are not handled by programs, however, and result in error messages as shown here:

Handling Exceptions

It is possible to write programs that handle selected exceptions.

Look at the following example, which asks the user for input until a valid integer has been entered.

fake_number.py

Can handle multiple Exception types

Handle 'other' Exceptions

Exception information

Clean up actions

The try statement has another optional clause which is intended to define clean-up actions that must be executed under all circumstances.

Command Line Arguments

Common utility scripts often need to process command line arguments.

These arguments are stored in the sys module’s argv attribute as a list.

For instance the following output results from running

command_line.py

at the command line:

is a representing the command line.

== python file name

Homework

Same as last week, but a new project!

https://github.com/vcu-chfauerbach/week5homework

Due March 1st, by 11:59:59 PM

Same rules apply. Make it green!

(Don't check it out yet, it's not working. I'll send an announcement later tonight when it works.)