PYTHON

FUNCTIONS II

OUTLINE

- Dictionary Functions
- Range Functions
- Date Functions
- Calendar Functions
- Global Variables
- Modules
- Functions

DICTIONARY

FUNCTION

A dictionary is a data type similar to arrays, but works with keys and values instead of indexes. Each value stored in a dictionary can be accessed using a key, which is any type of object (a string, a number, a list, etc.) instead of using its index to address it.

For example, a database of phone numbers could be stored using a dictionary like this:

DICTIONARY

FUNCTION

phonebook = {}

phonebook["John"] = 2349038477566

phonebook["Mike"] = 2347038377264

phonebook["Joel"] = 234814766278l

print(phonebook)

Alternatetivley, a dictionary can be initialized with the same values in the following notation:

phonebook = { "John" : 2349038477566, “Mike" : 2347038377264, "Joel" : 234814766278lb }

print(phonebook)

ITERATING OVER **DICTIONARIES**

Dictionaries can be iterated over, just like a list. However, a dictionary, unlike a list, does not keep the order of the values stored in it. To iterate over key value pairs, use the following syntax:

```
for name, number in phonebook.items():
    print("Phone number of %s is %d" % (name, number))
```

NB: Use the phonebook dictionary defined earlier.

REMOVING A **VALUE**

To remove a specified index, use either one of the following notations:

Use the phonebook dictionary defined earlier

```
del phonebook["John"]
print(phonebook)
```

or

```
phonebook.pop("John")
print(phonebook)
```

RANGE FUNCTION

The range function generates a list of numbers, which is generally used to iterate over with for loops.

There are many use cases. Often you will want to use this when you want to perform an action X number of times, where you may or may not care about the index. Other times you may want to iterate over a list (or another iterable object), while being able to have the index available.

RANGE FUNCTION
PARAMETERS

The range function provides a sequence of numbers from a start position to an end position.

Eg. Range(0,50)

Start: Starting number of the sequence.

Step: Difference between each number in the sequence.

Stop: Generate numbers up to, but not including this number.

DATE TIME
FUNCTIONS

Python just like other object oriented programming languages have in built system defined time functions which can be used for date time manipulations. Some of the functions include time, asctime, localtime(time). A tuple with the correct parameters can also be manipulated to represent time.

EXAMPLE

from datetime import time

from datetime import date

from datetime import datetime

First import the classes above:

1. Get the current date

   today = date.today()

   print("Today's date is ", today)

   today.weekday() will return the current day of the week.

EXAMPLE

Assuming we need to get only time from the datetime class

```
tm = datetime.time(datetime.now())

print(tm)
```

CALENDAR
FUNCTION

In python, the calendar module can be imported to help us find actual dates or display the graphic calendar for a particular year or for a particular month in a year.

EXAMPLE

Step 1: We begin with "import calendar" which will import all the classes of this module.

Step 2: c= calendar.TextCalendar(calendar.SUNDAY) tells the interpreter to create a text calendar. Start of the month will be Sunday. In Python, you can format the calendar as you can change the day of the month to begin with

Step 3: str= c.formatmonth(2025,1) We are creating calendar for the year 2025, Month 1 – January

Step 4: print str will print the output.

USER **DEFINED** MODULES

In python just like other programming languages, a user can define different modules saved to the python folder and imported into the interpreter. These user defined modules contain a set of functions performing different tasks. This helps in reusability of codes.

FUNCTIONS

Functions are a convenient way to divide your code into useful blocks, allowing us to order our code, make it more readable, reuse it and save some time. Also functions are a key way to define interfaces so programmers can share their code.

FUNCTIONS
EXAMPLE

```
def check():
    print("This is a function")
```

The return statement is also used in python to make functions send a value back to when the method was called.

The parameter list can be used defined empty or not depending on what the programmer wants to achieve.

NB: Function makes use of blocks, after the name.

FUNCTIONS
EXAMPLE

Example:

```
def check(a,b):
   return "%s is %b years older than me"%(a,b);
print(check("Okoro",5));
```

ASSIGNMENT

- Write a function to solve a quadratic equation with the almigty formula and return its value using a return statement.

Note: the variables should be passed through the parameter list.