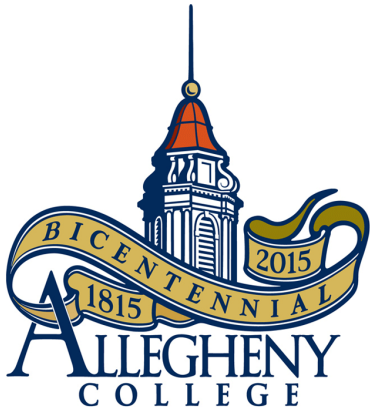
CS101 - Data Abstraction

Python Getting Started

Aravind Mohan

Allegheny College

March 9, 2021



Last Week

- What is a Program?
- What is Data?
- What is a Data Structure?
- How is a program, data, and data structure connected to each other?
- Where is Data coming from?
- How does your Program access Data?

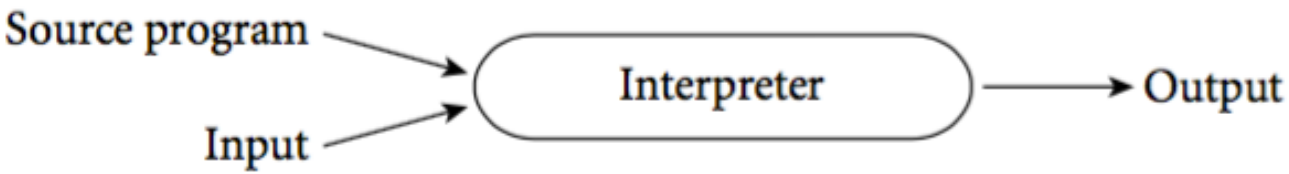
Refer Week1 slides, video, and notes ...

Discussion Based On ...

- GT (Goodrich Textbook) Chapter 01 [1.1,1.2,1.4,1.5,1.6,1.10]

What is an Interpreter?

- An interpreter is a computer program that directly executes instructions written in a programming language, without requiring them previously to have been compiled into a machine language program.
- Interpreter stays around for the execution of the program.
- Interpreter is the point of control during execution.
- Python use an Interpreter.



Interpreter Example

```
$ python3
>>> x = ``Hello, world! ''
>>> y = 4
>>> y*x
`Hello, world! Hello, world! Hello, world
    ! Hello, world! '
>>> x+y
Traceback (most recent call last): File
    ``<stdin>", line 1, in <module>
TypeError: cannot concatenate `str' and `
    int' objects
```

REPL: "Read-Eval-Print-Loop"
User repeatedly types in expressions that are immediately interpreted

Advantages of an Interpreter

- Greater flexibility
- Better diagnostics (error messages)
- E.g., in a REPL, programmer can decide what to do next based on output seen so far

- Some programming languages use Compiler. For example: C, C++, C#, Java
- Some programming languages use Interpreter. For example: **Python**, PHP, Perl, Ruby.
- More on compilers in CMPSC 201, Programming Languages course.

How do you represent Data in a Program?

- Variables
- Lists
- Class and Objects
- Linked List
- Stacks, Queues, Trees, Graphs
- Dictionary
- and so on $\cdots$

Why is Python Dynamically Typed?

- **Data type:** a data type is a classification of data which tells the compiler or interpreter how the programmer intends to use the data.
- Not required to declare the data type of variable. The data type is automatically (dynamic) inferred during the runtime of the program.

```
## assigning a value to a variable
x = [1, 2, 3,4,5,6,7,8,9,10]
print(type(x)) ## x is a list here
## reassigning a value to the 'x'
x = False
print(type(x)) ## x is a bool here
```

Primitive Data Type?

- **Primitive data type:** a primitive data type is the most basic data type available within a programming language.
- There are 4 main primitive data types in Python: integers, floats, boolean, string. These types serve as the building blocks of data manipulation in Python.

Mutable Vs Immutable

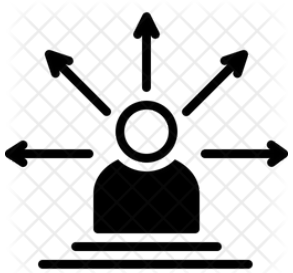
- **Immutable:** The value referenced by an object is a constant and cannot be changed once object created. For example: int, string, boolean, and float are immutable.
- **Mutable:** The value referenced by an object can be changed once object created. For example: list, set, dictionary, and user defined classes are mutable.

Let us talk about this more later ...

Type Conversion

- Converting data from one type to another
- Convert int to string
- Convert string to int
- and so on ....

Conditional Constructs



if, else if, and else

Conditional Constructs

Classify user-provided weather (temperature) as hot, mild, and cold!

```
temp = input("Enter the temperature:")
temp = int(temp)
if (temp > 80):
    print (" It 's hot !")
elif (temp > 60 and temp < 80):
    print (" It 's mild !")
else:
    print (" It 's cold !")
```

PS weather.py in the repo

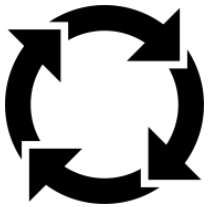
Conditional Constructs

Classify user-provided number as odd or even!

```
number = input("Enter a number (>0):")
number = int(number)
if  (number % 2 == 0):
   print (" It 's even number")
else:
   print (" It 's odd number")
```

PS classify.py in the repo

Iterative Loop Constructs



for and while

Iterative Loop Constructs (for)

Summation of all numbers between 0 and user-provided number!

```
number = input("Enter a number (>0):")
number = int(number)
total  = 0
for  item  in  range(0,number+1):
    total  += item
print ( total )
```

PS summation.py in the repo

Iterative Loop Constructs (while)

Classify user-provided phone number as valid or invalid!

```
number = input("Enter a phone number:")
number = int(number)
digits  = 1
while(True):
   number = int(number/10)
    if  (number <= 0):
      break
   else:
       digits  =  digits +1
if  ( digits  == 10):
   print (" It 's  a  valid  phone number!")
else:
   print (" It 's  an  invalid  phone number!")
```

PS phone.py in the repo

What is a Method?

- A method is a set of code which is referred to by name and can be called (invoked) at any point using its name.
- Methods have a signature.
- Formal argument and Actual argument.
- Global and Local scopes.

An Implementation Of Methods

Check if a given string is a Palindrome!

```
def check(word):
    for item in range(0,int(len(word)/2)):
        if (word[item] != word[len(word)-item-1]):
            return False
    return True
word = input("Enter a word:")
if (check(word) == True):
    print (" It 's a palindrome!")
else:
    print (" It 's not a palindrome!")
```

PS palindrome.py in the repo

Homework - Try Out Yourself

- **Solve GT: R-1.1 and R-1.4; Pg no 51**

Reading Assignment

- GT Chapter 01
  [1.1,1.2,1.4,1.5,1.6,1.10]