Introduction to Python

Chapter 1: Introduction

- Python is widely used, including by a number of big companies like Google, Pinterest, Instagram, Disney, Yahoo!, Nokia, IBM, and many others.

- The Raspberry Pi - which is a mini computer - relies on Python as it's main programming language too.

- Once you learn Python, you'll never have a shortage of ways to utilize the skill. Not to mention, since a lot of big companies rely on the language, you can make good money as a Python developer.

(https://www.pythonforbeginners.com/)

Introduction

Python is:

- strongly typed (i.e. types are enforced)
- dynamically, implicitly typed (i.e. you don’t have to declare variables)
- case sensitive (i.e. var and VAR are two different variables)
- object-oriented (i.e. everything is an object).

(https://www.stavros.io/tutorials/python/)

Syntax

- Python has **no mandatory statement termination characters**
- **blocks are specified by indentation**. Indent to begin a block, de-indent to end one.
- Statements that expect an indentation level end in a colon (:) e.g. if statement
- **Comments** start with the pound (#) sign and are single-line
- **multi-line comments**.

```
'''  comment
    comment
 '''
```

Syntax

- **Values are assigned** (in fact, objects are **bound** to names) with the *equals* sign (“=”)
- **equality testing** is done using two *equals* signs (“==”).
- You can increment/decrement values using the += and -= operators respectively by the right-hand amount. This works on many datatypes, strings included. You can also use multiple variables on one line.

Types

- Python Numbers
  - integers, floating point numbers and complex numbers falls under **Python numbers** category. They are defined as int, float and complex class in Python.
- Python List
- Python Tuple
- Python Strings
- Python Set
- Python Dictionary

Conversion between data types

- We can convert between different data types by using different type conversion functions like int(), float(), str() etc.

  ```
  >>> float(5)
  5.0
  ```

- Conversion from float to int will truncate the value (make it closer to zero).

  ```
  >>> int(10.6)
  10
  >>> int(-10.6)
  -10
  ```

Conversion between data types

Conversion to and from string must contain compatible values.

```
>>> float('2.5')
2.5

>>> str(25)
'25‘



>>> int('1p')   ERROR
```

https://www.programiz.com/python-programming/

COMP2602 Focus

- In this course, you would be mainly concerned with some of the networking aspects of Python, e.g. TCP and UDP programming.

- However, in order to develop an application that uses networking, you would need to know other aspects of the language e.g. Write a program to implement a network quiz game.
  How is client/server communication involved here?

Sample Code - Print

```
#A# Print a message: Welcome to Python
#print statement
print("Welcome to Python")

#Output: Welcome to Python
```

Sample Code - Sequence

```
#B# Sequence statements
#B1# Find the sum of x and y
x=5
y=10
print ("Sum of " + str(x) + " and " + str(y) + " is " + str(x + y))
#str() converts int to string
#Output: Sum of 5 and 10 is 15
```

Sample Code - Sequence

```
#B2# Read a value of x and find f(x)=x^2-4
x=input("Please enter integer x: ")
result=pow(int(x),2) - 4 #x is string, so convert to
                                        integer #using int() function
print(result)
#Output: Please enter integer x: 5
#        21
```

Sample Code - Sequence

#B3# Read 2 positive integers x and y and print x % y

```
x, y = input("Please enter 2 positive integers: ").split()
#split - split input by whitespace
print(int(x) % int(y))
#Output: Please enter 2 positive integers: 13 5
#       3
```

Sample Code – Selection

#C# Selection statements

```
if expression:
   statement(s)
```

Note the indentation

#C1.1 #if-Read an int value and print "LT" if it is less than 10

```
x=input("Please enter int: ")
if int(x)<10:
   print ("LT")
#Output:
'''
Please enter int: 5
LT
'''
```

Sample Code – Selection

If/else

```
if expression:
   statement(s)
else:
   statement(s)
```

(Note the indentation)

```
#if-else Valid or invalid mark. Valid: 0 <=mark<=100

print("if..else")
mark=input("Please enter mark ")
mark=int(mark)
if mark <0 or mark > 100:
    print ("Invalid mark")
else:
    print ("Valid mark")
```

#Output:

if mark is -5, output is : “Invalid mark”. Mark=50, “Valid mark”

Sample Code – if/elif

```
#C1.3 if-elif grades A 90-100, B 60-89, C 50-59, F 0-49

print("if..elif..else grades A 90-100, B 60-89, C 50-59, F 0-49")
mark=input("Please enter mark: ")
mark=int(mark)
if mark < 0 or mark > 100:
    print ("Invalid mark")
elif mark >=90:                              #Note indentation
    print ("A")
elif mark >=60:
    print ("B")
elif mark >=50:
    print ("C")
else:
    print ("F")
```

Sample Code – if/elif

```
#Output
'''
if..elif..else grades A 90-100, B 60-89, C 50-59, F 0-49
Please enter mark: 50
C

if..elif..else grades A 90-100, B 60-89, C 50-59, F 0-49
Please enter mark: -9
Invalid mark
'''
```

Sample Code – Repetition-while

```
while expression:
      statement(s)
```

```
#D1.1 while: Print i while it is less than 5
i = 1
while i < 5:
  print(i)
  i += 1
'''Output:
1
2
3
4
'''
```

Sample Code – Repetition-while

```
#D1.2 while:
#Find the sum of the integers 2, 4, 6..10
x=2
sum=0
while x <= 10:
    sum=sum+x
    x=x+2
print("Sum is " + str(sum))

#Output: Sum is 30
```

Sample Code –while with break statement

```
#D1.3 while:
#Illustrate break statement in a while 1 infinite loop
x=2
sum=0
while 1:
    sum=sum+x
    x=x+2
    if x==6: #sum is 0 + 2 + 4, x is 6, but does not add 6 to sum
        break; #exit while loop
print("Sum is " + str(sum))

#Output: Sum is 6
```

Sample Code – Repetition - For Loop

The for loop is used to iterate over elements of a sequence

It is often used when you have a piece of code which you want to repeat "n" number of time.

It works like this: " for all elements in a list, do this “

https://www.pythonforbeginners.com/

Sample Code – For loop

```
#D2.1 for: Print all ints from 1 to 4, and print "Done" when loop ends
print("\nPrinting the integers from 1 to 4:")
for x1 in range(4): #range (4) is 0..3
  print(x1+1)
else: #loop has ended
  print("Done.")
#Output:
'''

Printing the integers from 1 to 4:
1
2
3
4
Done.
'''
```

Sample Code - for

```
#D2.2 for: Print all even ints from 2 to 10  inclusive
print("\nPrinting the even integers from 2 to 10:")
for x1 in range(2,11,2): #range(begin,end,step), end is 11, so loop up to 10
  print(x1)
else: #loop has ended
  print("Done even.")

#Output:
'''Output: Printing the even integers from 2 to 10:
2
4
6
8
10
Done even.
'''
```

Sample Code - For

```
#D2.3 for: Print all even ints from 2 to 10  inclusive (reversed)
print("\nPrinting the even integers from 2 to 10 in reverse:")
for x1 in reversed(range(2,11,2)): #range(begin,end,step), end is 11, so loop up to 10
  print(x1)
else: #loop has ended
  print("Done even.")
```

```
#Output:
'''
Printing the even integers
from 2 to 10 in reverse:
10
8
6
4
2
'''
```

Sample Code – for - list

```
#D2.4 for: Illustration with a list
animals = ["bear", "lion", "tiger"]
for x in animals:
  print(x)
#Output:
'''
bear
lion
tiger
'''
```

Sample Code – For - reversed

```
#D2.5 for: Illustration with a list (print reverse)
animals = ["bear", "lion", "tiger"]
for x in reversed(animals):
  print(x)

#Output:
'''
tiger
lion
bear
'''
```

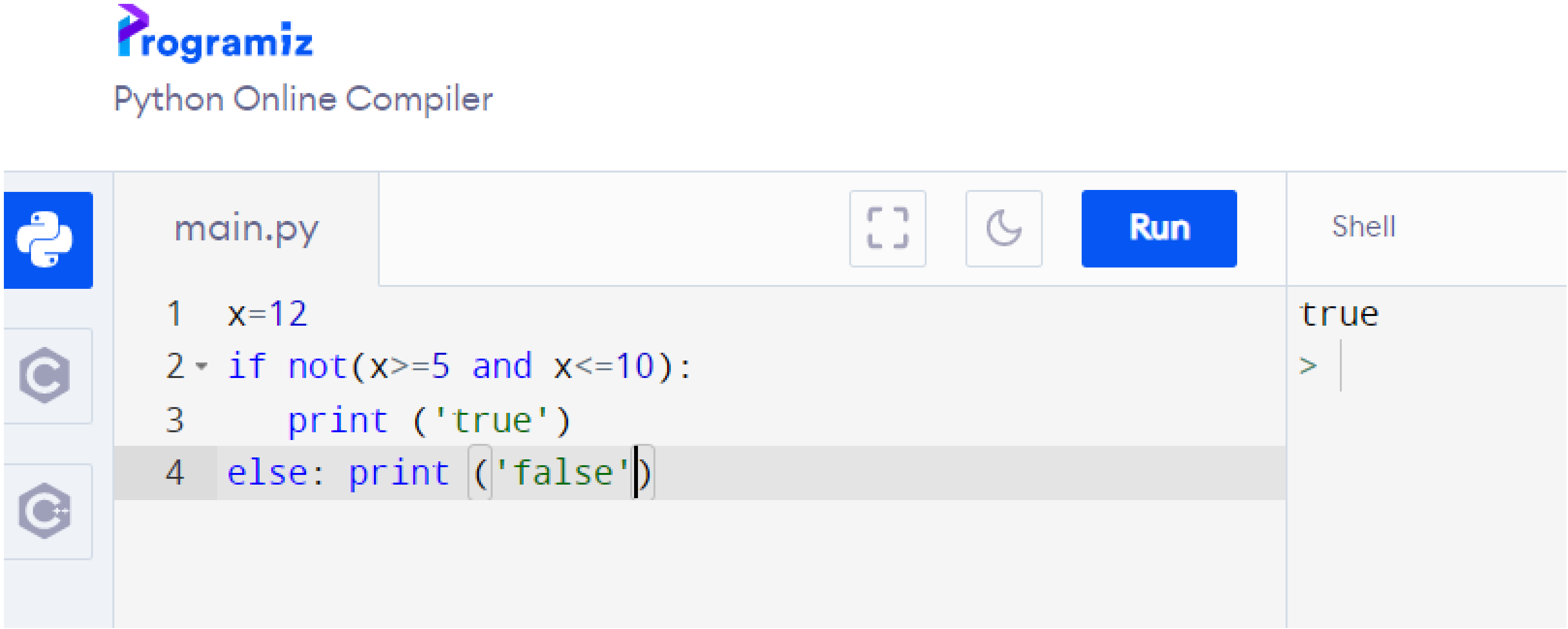
Sample Code – and

```
#E and
mark=input("Please enter a mark to check valid using 'and' statement")
mark=int(mark)
if mark >= 0 and mark <= 100:
    print("Valid")
else:
    print("Invalid“)
```

```
#output
Mark is 50- Valid
Mark is 150- Invalid
```

Sample Code – not

#F not



Programiz

Python Online Compiler

```
x=12
if not(x>=5 and x<=10):
    print ('true')
else: print ('false')
```

```
true
>
```

Sample Code – not

#F not

```
x=12
if not(x>=5 and x<=10):
   print ('true')
else: print ('false’)
```

```
true
```

Next

Python Networking code
(Application Layer Slides)

Python Practice would be done in the lab