Department of Computer Technology and Information Systems

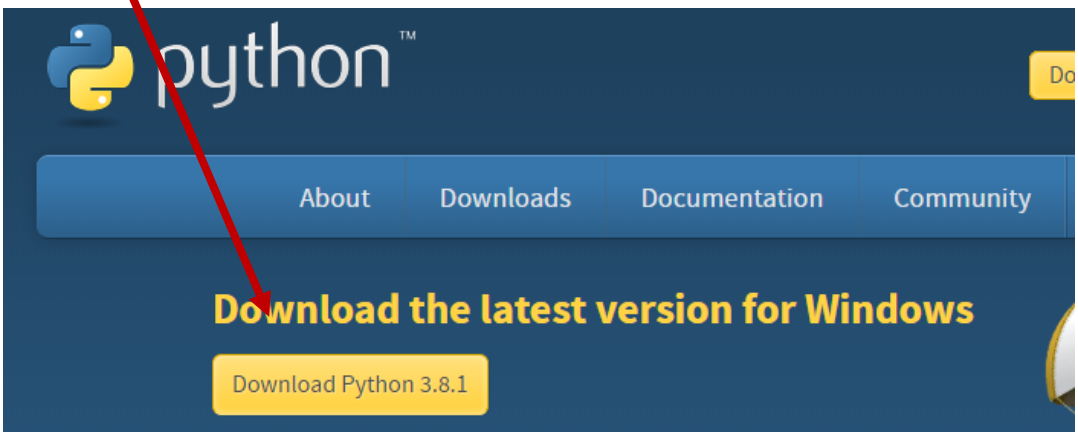
CTIS264 – Computer Algorithms

Spring 2019 - 2020

Lab Guide 1 - Week 2

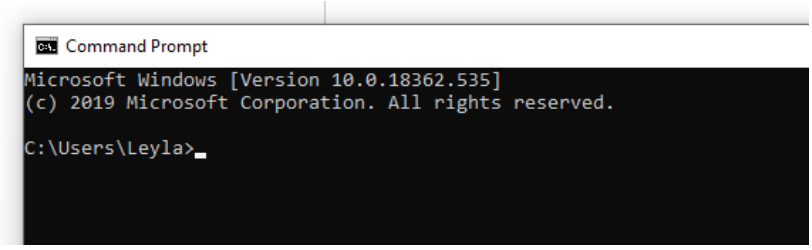
Instructor : Füsun YÜRÜTEN **Assistant** : Leyla SEZER
Objectives: • **Installation of python and packages** • **Python basics** • **Types** • **How to get an input** • **String operations** • **Slicing** • **Arithmetic expressions** • **Built-in functions** • **Math module** • **User-defined functions** • **If statement** • **For and While Loops** • **GCD Algorithms (Euclid's and CIC Algorithm)**

For this course, we will using the **Python Shell IDLE,** which you can download from the web site**;** https://www.python.org/downloads/



Then, you need some python packages**,** these are**; numpy, pandas, matplotlib and xlrd**. Make the following instructions, to install the given packages.

- Open the command prompt, by writing **cmd** to the start menu.



- Write the following commands to install the given packages (numpy, pandas, matplotlib)
  - pip install numpy
  - pip install pandas
  - pip install matplotlib.
  - pip install xlrd

```
C:\Users\Leyla>pip install numpy
Collecting numpy
  Using cached https://files.pythonhosted.org/packages/a9/38/f6d6d8635d496d6b4ed5d8ca4b9f193d0edc59999c3a63779cbc38aa650
f/numpy-1.18.1-cp37-cp37m-win_amd64.whl
Installing collected packages: numpy
  WARNING: The script f2py.exe is installed in 'C:\Users\Leyla\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.
7_qbz5n2kfra8p0\LocalCache\local-packages\Python37\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-1.18.1
WARNING: You are using pip version 19.2.3, however version 19.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\Leyla>pip install pandas
Collecting pandas
  Using cached https://files.pythonhosted.org/packages/02/d0/1e8e60e61e748338e3a40e42f5dfeee63ccdecfc4f0894122b890bfb009
a/pandas-0.25.3-cp37-cp37m-win_amd64.whl
Requirement already satisfied: pytz>=2017.2 in c:\users\leyla\appdata\local\packages\pythonsoftwarefoundation.python.3.7
_qbz5n2kfra8p0\localcache\local-packages\python37\site-packages (from pandas) (2019.3)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\leyla\appdata\local\packages\pythonsoftwarefoundation.
python.3.7_qbz5n2kfra8p0\localcache\local-packages\python37\site-packages (from pandas) (2.8.1)
Requirement already satisfied: numpy>=1.13.3 in c:\users\leyla\appdata\local\packages\pythonsoftwarefoundation.python.3.
7_qbz5n2kfra8p0\localcache\local-packages\python37\site-packages (from pandas) (1.18.1)
Requirement already satisfied: six>=1.5 in c:\users\leyla\appdata\local\packages\pythonsoftwarefoundation.python.3.7_qbz
5n2kfra8p0\localcache\local-packages\python37\site-packages (from python-dateutil>=2.6.1->pandas) (1.13.0)
Installing collected packages: pandas
```

Q1. Write a Python code segment; to make soft introduction to Python**.**

```
>>> print("Hello World")
Hello World
>>> print(2 * 3)
6
>>> print("Welcome to","CTIS264")
Welcome to CTIS264
>>> print(6*3,'is the result')
18 is the result
>>> x=10
>>> print(x)
10
>>> type(x)
<class 'int'>
>>> y='10'
>>> print(y)
10
>>> type(y)
<class 'str'>
>>> z=5.8
>>> type(z)
<class 'float'>
>>> y = int(z)
>>> print(y)
5
>>> type(y)
<class 'int'>
>>>
```

- To get an input from the user, write the following code segment.

Code segment:

```
>>> val = input("Enter your name and surname:")
Enter your name and surname:Leyla Sezer
>>> age = int(input("Enter your age:"))
Enter your age:35
>>> print(val,"you are",age,"years old")
Leyla Sezer you are 35 years old
>>>
```

Q2. Write a Python code segment that;

- Defines a string as shown below, then gets some part of the string and shows in the screen which is named **slicing** in python.

Code segment:

```
>>> s="abcdefgh"
>>> print(s[3:6])
def
>>> print(s[3:6:2])
df
>>> print(s[::])
abcdefgh
>>> print(s[::-1])
hgfedcba
>>> print(s[4:1:-2])
ec
```

Q3. Write a Python code segment that uses the following arithmetic expressions;
Use the given table to remind for the arithmetic expressions**;**

Expression	Meaning
x + *y*	*x* added to *y*, if *x* and *y* are numbers *x* concatenated to *y*, if *x* and *y* are strings
x - *y*	*x* take away *y*, if *x* and *y* are numbers
x * *y*	*x* times *y*, if *x* and *y* are numbers *x* concatenated with itself *y* times, if *x* is a string and *y* is an integer *y* concatenated with itself *x* times, if *y* is a string and *x* is an integer
x / *y*	*x* divided by *y*, if *x* and *y* are numbers
x // *y*	Floor of *x* divided by *y*, if *x* and *y* are numbers
x % *y*	Remainder of *x* divided by *y*, if *x* and *y* are numbers
x ** *y*	*x* raised to *y* power, if *x* and *y* are numbers

Code segment:

```
>>> print(12/5)
2.4
>>> print(12//5)
2
>>> print(10/4.0)
2.5
>>> print(4 /10)
0.4
>>> print(4 // 10)
0
>>> print(12 % 3)
0
>>> print(10 % 3)
1
>>> print(3 % 10)
3
>>> print(2 ** 3)
8
```

Q4. Write a Python code segment that**;**

- Uses the following built-in functions (round, abs, len),
- Then uses the math module to see the functions (sqrt, ceil, floor, pi).

Code segment:

```
>>> round(3.2)
3
>>> abs(-16)
16
>>> len('abcdef')
6
>>> round(-3.9)
-4
>>> round(3.9)
4
```

```
>>> #import the math module to access functionality
>>> import math
>>> math.sqrt(25)
5.0
>>> math.ceil(-2.8)
-2
>>> math.floor(7.9)
7
>>> math.pi
3.141592653589793
```

Q5. Write a Python program that makes following operations, using the string functions.

- Input a full name from the user,
- Count the number of characters in the full name,
- Count the number of ‘a’ in the full name,
- Find the position of blank from the start,
- Find the position of the surname from the beginning,
- Change the full name to uppercase and lowercase format,
- Find the length of name and surname separately,
- Replace all ‘e’ to ‘w’.

Function Name	Purpose
s.count(s1)	Counts how many times the string s1 occurs in s.
s.find(s1)	Returns the index of the first occurrence of the substring s1 in s, and -1 if s1 is not in s.
s.rfind(s1)	Same as find, but starts from the end of s (the 'r' in rfind stands for reverse)
s.replace(old,new)	Replaces all occurrences of the string old in s with the string new.
s.strip()	Removes the leading and trailing whitespace from s.

Output:

```
Enter your name: Beren Saat
Your name has 10 characters
Your name contains 2 'a' characters
Position of blank from start: 5
Surname is Saat
Position of your surname from beginning: 6
Your name in lowercase: beren saat
Your name in uppercase: BEREN SAAT
Length of name is  5 Length of surname is 4
After replace all 'e' to 'w'  Bwrwn Saat
```

Q6. Write a Python program that defines a function that;

- Gets several integers from the user until '0' is entered, finds the sum of odd numbers as in the output.

Output:

```
Enter an integer value except (0):6
Enter an integer value except (0):5
Enter an integer value except (0):3
Enter an integer value except (0):2
Enter an integer value except (0):1
Enter an integer value except (0):7
Enter an integer value except (0):8
Enter an integer value except (0):0
16
```

Q7. Write a Python program that;

- Writes a function that takes two positive integer and returns GCD which is found by ***Consecutive Integer Checking Algorithm***.

  Consecutive integer checking algorithm;

  - Step 1: Assign the value of min(*m,n*) to *t.*
  - Step 2: Divide *m* by *t.* If the remainder is 0, go to Step 3; otherwise, go to Step 4.
  - Step 3: Divide *n* by *t.* If the remainder is 0, return *t* and stop; otherwise, go to Step 4.
  - Step 4: Decrease *t* by 1 and go to Step 2.
- Then program prompts the user to enter two positive integer and displays the GCD of them.

Output:

```
A non-negative integer please: 64
Another non-negative integer please: 12
GCD from consecutive check :  4
```

Q8. Write a Python program that;

- Writes a function that takes two positive integer and returns GCD which is found by ***Euclid's Algorithm***.

  Euclid's algorithm;

  - Step 1: If *n* = 0, return *m* and stop; otherwise go to Step 2.
  - Step 2: Divide *m* by *n* and assign the value for the remainder to r.
  - Step 3: Assign the value of *n* to *m* and the value of *r* to *n then* go to Step 1.
- Then program prompts the user to enter two positive integer and displays the GCD of them.

Output:

```
Enter first number for GCD  : 126
Enter second number for GCD : 46
GCD from euclid's  2
```