Python

Lab -1-B

Halo Khalil Sharif

What is Python?

- Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

- It is used for:
- web development (server-side)
- software development
- mathematics
- system scripting.

Python Install

- To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>py --version
```

Install Python

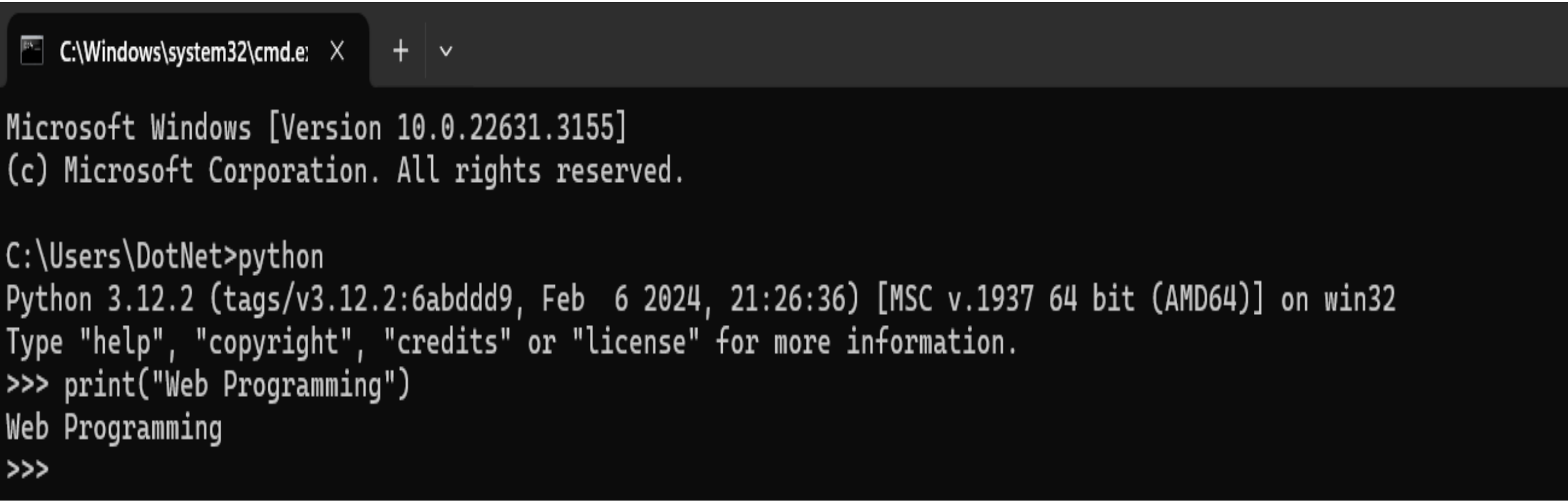
- https://www.python.org/downloads/

Print

- `print("Hello, World!")`

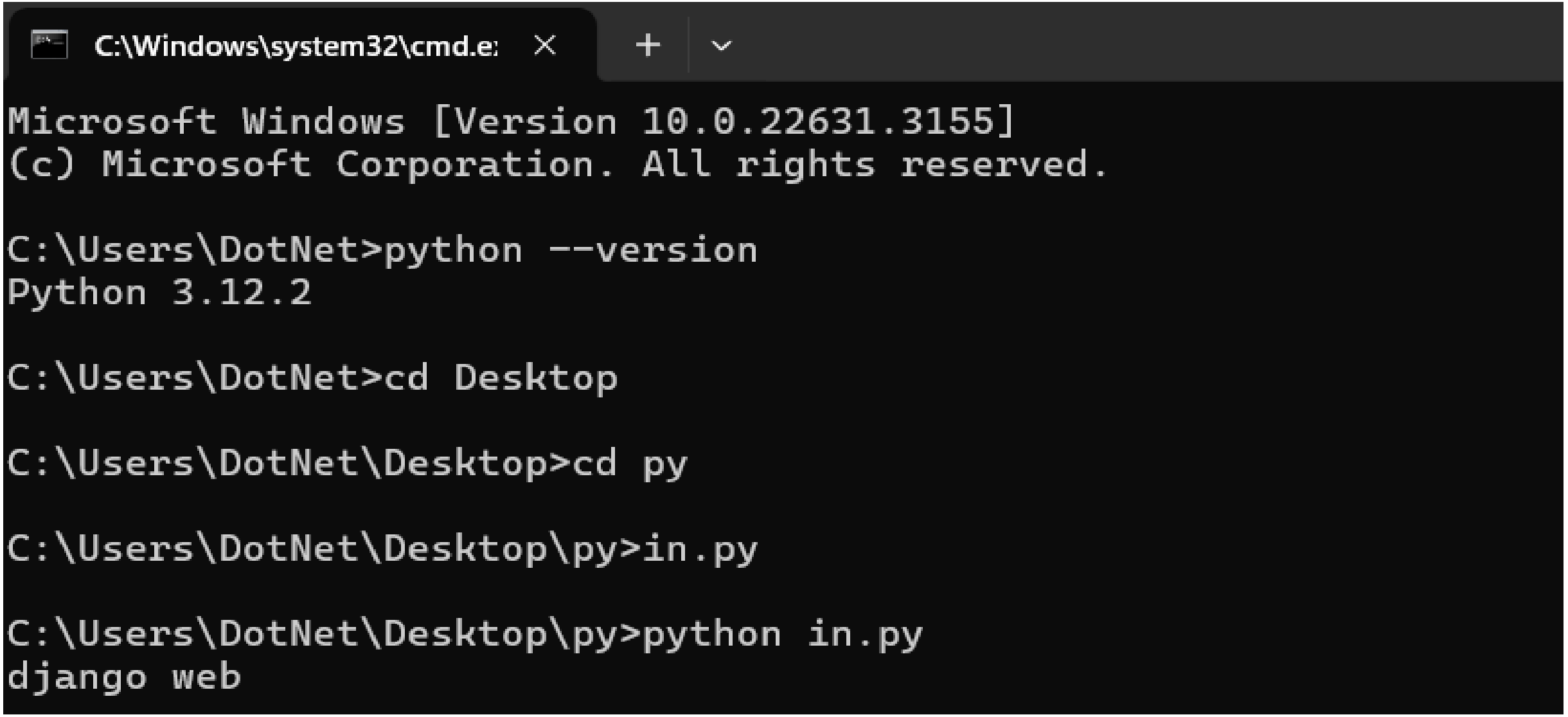
The Python Command Line

```
C:\Users\>python
```




```
>>>exit()
```

Python Quick start



Python Comments

- ```
  #This is a comment
  print("Hello, World!")
  ```
  ```

Python Variables

- Python has no command for declaring a variable.
- A variable is created the moment you first assign a value to it.
- ```
  x = 5
  y = "Web"
  print(x)
  print(y)
  ```
  ```

Assign Multiple Values

- ```
  x, y, z = "A", "B", "C"
  print(x)
  print(y)
  print(z)
  ```
  ```

Output Variables

- ```
  x = "Python is awesome"
  print(x)
  ```

- ```
  x = 5
  y = "A"
  print(x, y)
  ```

Python Indentation

- Indentation refers to the spaces at the beginning of a code line.
- Python uses indentation to indicate a block of code.

```
if 5 > 2:
  print("Five is greater than two!")
```

```
if 5 > 2:
 print("Five is greater than two!")
if 5 > 2:
        print("Five is greater than two!")
```

Python Functions

- A function is a block of code which only runs when it is called.
- You can pass data, known as parameters, into a function.
- A function can return data as a result.

Creating a Function

- In Python a function is defined using the `def` keyword:

```
def my_function():
  print("Hello from a function")
```

Calling a Function

- To call a function, use the function name followed by parenthesis:

```
def my_function():
  print("Hello from a function")

my_function()
```

Arguments in Function

- Information can be passed into functions as arguments.
- Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

```
def my_function(fname):
  print(fname + "Text")

my_function("A")
my_function("B")
my_function("C")
```

If … Else

- Python Conditions and If statements
- Python supports the usual logical conditions from mathematics:
- Equals: `a == b`
- Not Equals: `a != b`
- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`

continue

- An "if statement" is written by using the `if` keyword.

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

Elif

- The `elif` keyword is Python's way of saying "if the previous conditions were not true, then try this condition".

```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

Else

- The `else` keyword catches anything which isn't caught by the preceding conditions.

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

Short Hand If

- If you have only one statement to execute, you can put it on the same line as the if statement.

```
if a > b: print("a is greater than b")
```

Short Hand If … Else

- If you have only one statement to execute, one for if, and one for else, you can put it all on the same line:

```
a = 2

b = 330

print("A") if a > b else print("B")
```

Nested If

- You can have if statements inside if statements, this is called nested if statements.

```
x = 41

if x > 10:
  print("Above ten,")
  if x > 20:
    print("and also above 20!")
  else:
    print("but not above 20.")
```

Loops

- Python programming language provides two types of Python loopshecking time. In this article, we will look at Python loops and understand their working with the help of examp – **For loop** and **While loop** to handle looping requirements.

While Loop in Python

```
count = 0
while (count < 3):
    count = count + 1
    print("Hello")
```

For Loop in Python

```
n = 4
for i in range(0, n):
    print(i)
```

User Input

- Python 3.6

```
username = input("Enter username:")
print("Username is: " + username)
```

- Python 2.7

```
username = raw_input("Enter username:")
print("Username is: " + username)
```