# Python

Section 2

# Python Intro

- Python is a popular programming language.
- **It is used for:**
  - web development (server-side),
  - software development,
  - mathematics,
  - system scripting.
- **Why Python?**
  - works on different platforms (Windows, Mac, Linux, etc).
  - simple syntax similar to the English language.
  - syntax that allows developers to write programs with fewer lines .
  - runs on an interpreter system, meaning that code can be executed as soon as it is written.

# Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

# Examples

- print("Hello, World!")
- print("Five is greater than two!")
- print ("Five is greater than two! ")
  print(“Two is less than five!")

  Syntax Error: (Indentation)

# Comments

- **Creating a Comment**
- Comments starts with a #, and Python will ignore them:
- **Example**
  - #This is a comment
    print("Hello, World!")
  - print("Hello, World!") #This is a comment
- **Multi Line Comments**
- Python does not really have a syntax for multi line comments.
- To add a multiline comment you could insert a # for each line:
  - #This is a comment
    #written in
    #more than just one line
    print("Hello, World!")

# Python Variables

- **Variables**
  - Variables are containers for storing data values.
- **Creating Variables**
  - Python has no command for declaring a variable.
  - A variable is created the moment you first assign a value to it.
- **Example**
- ```
  x = 5
  y = "John"
  print(x)
  print(y)
  ```
  ```

- Variables do not need to be declared with any particular *type*, and can even change type after they have been set.
- **Example**
- x = 4 # x is of type int
  x = "Sally" # x is now of type str
  print(x)
- **Case-Sensitive**
- Variable names are case-sensitive.
- **Example**
- This will create two variables:
- a = 4
  A = "Sally"
  #A will not overwrite a

# Simple data types

- Numbers
  - Integer, floating-point, complex!
- Strings
  - characters are strings of length 1
- Booleans are `False` or `True`

# Examples

```
i = 10
d = 3.1415926
s = "I am a string!"
print “newline\n"
print "no newline"
```

# Operators

- `+ - * / %`
- `+= -=` etc. (no `++` or `--`)
- Assignment using `=`
  - but semantics are different!

```
a = 1
a = "foo"   # OK
```

- Can also use `+` to concatenate strings

# Strings

- "hello"+"world" "helloworld" # concatenation
- "hello"*3 "hellohellohello" # repetition
- "hello"[0] "h" # indexing
- "hello"[-1] "o" # (from end)
- "hello"[1:4] "ell" # slicing
- len("hello") 5 # size
- "hello" < "jello" 1 # comparison
- "e" in "hello" 1 # search
- New line: "escapes: \n "
- Line continuation: triple quotes ”’
- Quotes: ‘single quotes’, "raw strings"

# Casting

- If you want to specify the data type of a variable, this can be done with casting.
- **Example**
- x = str(3)  # x will be '3'
  y = int(3)  # y will be 3
  z = float(3)  # z will be 3.0

# Get the Type

- You can get the data type of a variable with the type() function.
- **Example**
- x = 5
  y = "John"
  print(type(x))
  print(type(y))

- **Illegal variable names:**
- 2myvar = "John"
  my-var = "John"
  my var = "John"
- **Multi Words Variable Names**
- difficult to read.
- There are several techniques you can use to make them more readable:
  - myVariableName = "John"
  - MyVariableName = "John"
  - my_variable_name = "John"

- **Many Values to Multiple Variables**
- Python allows you to assign values to multiple variables in one line:
- **Example**
- x, y, z = "Orange", "Banana", "Cherry"
  print(x)
  print(y)
  print(z)
- **One Value to Multiple Variables**
- And you can assign the *same* value to multiple variables in one line:
- **Example**
- x = y = z = "Orange"
  print(x)
  print(y)
  print(z)

# Output variables

- In the print() function, you output multiple variables, separated by a comma:
- **Example**
- x = "Python"
  y = "is"
  z = "awesome"
  print(x, y, z)
- You can also use the + operator to output multiple variables:
- print(x + y + z)
- when you try to combine a string and a number with the + operator, Python will give you an error:
- **Example**
- x = 5
  y = "John"
  print(x + y) → error

# exercise

- Write a program to Calculate and print the area and circumference of the circle:

```
r = 10          # radius
print("Area =", 3.14159*r**2, "m2")
print("Circumference =", 2*3.14159*r, "m")
```

- Output:
- Area = 314.159 m2
- Circumference = 62.8318 m