ICS-104

Chapter 2 Notes

By:Naif Alqahtani

twitter : @NaifAlqahtani

Youtube.com/MWNLLT

twitter: @NaifAlqahtani
Youtube.com/MWNLLT

• Variables

- Why?
  - To store data
  - To be able to use the same data somewhere else
- Definition:
  - A variable is a storage location in the memory
  - Each variable can be assigned a name
  - Variables have: lables and Values
- How?
  - We use an assignment statement to store data into a variable
  - Say we want to store the value 6 into a variable called six.
    - We do:

```
Six = 6
```

- Assignment Statement:
  - Use an equal sign " = "
  - anything on the right of " = " is the data to be stored
  - anything on the left of " = " is the name given to the location.
- Output:

`print(Six)` will output 6.

- Note:
  - Assignment is NOT equality like in math

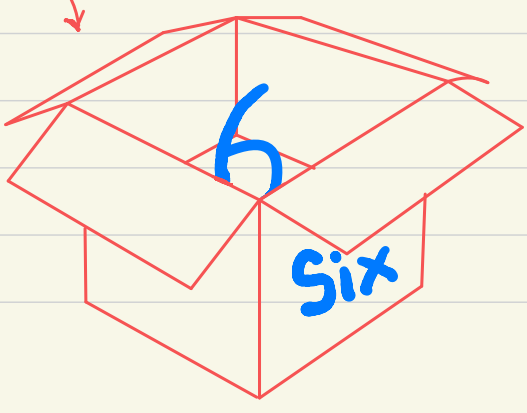
twitter: @NaifAlqahtani
Youtube.com/MWNLLT

twitter: @NaifAlqahtani
Youtube.com/MWNLLT

• Data Types:

- What?
  - Data is stored in different types
- Why?
  - Data types determine the allowed operations on a variable.
  - Determine how the data will be presented.
- Types:
  - Primitive Data types:
    - Integers "int":
      - whole numbers only, positive & negative
    - Strings "str":
      - anything inside of quotation marks " ".
      - can be letters, numbers and symbols.
    - Floats "float":
      - All numbers whole or fraction
      - Decimals only
      - 2.0 is a float, 2 is an integer.
    - Boolean "bool":
      - can be either True or False only
      - eg. `opened = False`
  - User-Defined Data types:
    - will study in Chapter 9

A Variable →



twitter: @NaifAlqahtani
Youtube.com/MWNLLT

twitter: @NaifAlqahtani
Youtube.com/MWNLLT

• Variable names: The Rules!

- Names can only start with english letters or _
- Remaing characters must be:
  - letters
  - Numbers
  - _ ← under-score
- No spaces!
- No reserved words. eg:
  - if x
  - class x
  - for x

• Variable names: Recommendations

- Use descriptive names.
- Use Camel Case Capitalization: Capitalize first letter of each word
  - `numberOfHoursInDay = 24`
- Use under-scores as spacers
  - `number_of_hours_in_day = 24`

Note: these are also considered constants and should be all caps. But just for demonstration's sake :)

- When defining costants, use ALL CAPS
  - `MAX_LENGTH_OF_WORD = 45`

• Comments

- Use comments as much as possible
- Make sure comments are understandable
- Very helpful to understand code.

using comments is important


```
MAX_LENGTH_OF_WORD = 45  # this is the max number of letters allowed in a single word
```

comment starts

twitter: @NaifAlqahtani
Youtube.com/MWNLLT

twitter: @NaifAlqahtani
Youtube.com/MWNLLT

• Arithmetics

- Math stuff but in code
- Python supports these arithmetics operators.
  - + , addition
  - − , subtraction
  - * , multiplication
  - / , division
  - ** , exponents
- a combination of operators is called Arithmetic Expressions

• Order of Arithmetic operations

1. () , brackets
2. ** , exponents
3. * , / , Multiplication & Division
4. + , − , Addition & Subtraction

• Other Arithmetic Operations

1. floor operator :
   - // ← floor Operator
   - Similar to greatest integer function in math $[\![x]\!]$
   - returs greatest integer less than givin value
   - e.g: 7//4 → 1
   - 7/4 is 1.75 and 1 is the greatest integer that is less than 1.75
2. Remainder operator (A.K.A: Modulus)
   - % ← Modulus operator.
   - returns remainder after division
   - e.g: 7 % 4 → 3
   - becaus 7/4 is $1\frac{3}{4}$ ← so 3 will be the output

twitter: @NaifAlqahtani
Youtube.com/MWNLLT

• Functions

- Functions usually return a value.
- Eg. Functions:
  - abs(-3) will return 3
  - round(7.6) will return 8
  - max(7,9,3,5,1) will return 9
- As you can see, some functions take one argument and others can take multiple arguments.

• Libraries

- code already written by other programmers.
- Used to share functions to make coding easier
- eg library: math
- a standard library is a library that is built into the language
- Importing a library:
  - if we want to find $\sqrt{4}$, we can use the sqrt() function from the math library.

```
from math import sqrt
print(sqrt(4))
```

  - this will print 2

twitter: @NaifAlqahtani
Youtube.com/MWNLLT

• Strings

- As I mentioned earlier:
  - Strings "str":
    - anything inside of quotation marks " ".
    - can be letters, numbers and symbols.
- use either single-quotation or double-qoutations.
- the len() function returns the number of characters in a string
- Concatinating (sticking together) two string.
- two concatinate two strings, we use + between 2 strings only

```
firstName = "Naif "
lastName = "Alqahtani"
name = firstName + lastName
print(name)
# output -> Naif Alqahtani
```

- we can repeat a string by multiplying it

```
print("Naif\n" * 50, end="") # prints Naif 50 times
```

- we can use str(), int(), float() to convert between data types
- we can access specific parts of a sting by indexing into the string.
- String[3] will print the 4th character because indexing starts from 0.
- indexing to a character that is greater than the length of the string will produce an error!

twitter: @NaifAlqahtani
Youtube.com/MWNLLT

Inputs

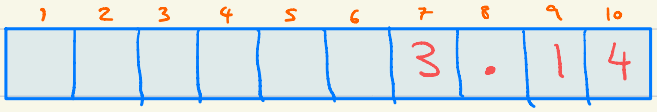
- we can ask the user to input by using the input() function.
- Input always returns the value as string, even if a number is inserted.
- to do math with an input we must convert it into a number using either of these function: int(), float()

Outputs

- controlling the appearance of output
- uses string format operator: %
- %.2f will print 2 decimals only

```
PI = 3.14159
print("%.2f" % PI)
# output -> 3.14
```

- using % 10.2f will print 10 spaces and 2 decimals.



- Use %d for integers
- Use %s for strings