**Python for data science**

**Data type in python**

|  |  |
| --- | --- |
| Text Type: | Str “Haroon Rasheed” |
| Numeric Types: | int, float, complex 3,3.0,H3 |
| Sequence Types: | list, tuple, range |
| Mapping Type: | Dict |
| Set Types: | set, frozenset |
| Boolean Type: | Bool |
| Binary Types: | bytes, bytearray, memoryview |
| None Type: | NoneType |

**Variables in python**

Python variables are simply containers for storing data value.

* A variable name must start with a letter or the underscore character
* A variable name cannot start with a number
* A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
* Variable names are case-sensitive (age, Age and AGE are three different variables)
* A variable name cannot be any of the python keywords.

**Operators in python**

* **Python Arithmetic Operators**

**Operator Name Example**

**+** Addition x + y

- Subtraction x – y

\* Multiplication x \* y

/ Division x / y

% Modulus x % y

\*\* Exponentiation / power x \*\* y

// Floor division x // y

* **Python Comparison Operators**

**Operator Name Example**

**==** Equal x == y

!= Not Equal x != y

> Greater than x > y

**<** Less than x < y

>= Greater than or equal to x >= y

<= Less than or equal to x <= y

**Some useful function in python**

**Round( )** function round the input to a specific number of places or to the nearest integer. Only one argument in this function.

print(round(4.55683, 3)) output is 4.557

**divmod(x,y)** output the quotient and the remainder in a tuple (we will see what a tuple is)

print(divmod(34,5)) output is (6 , 4)

**isinstance( )** returns True, if the first argument is an instance of that class. Multiple classes can also be checked at once.

b = 4 print(isinstance(b, int)) output is true

c = 5.4 print(isinstance(b, int)) output is false

d = "Haroon" print(isinstance(b, str)) output is true

**pow(x,y,z)** x raise to the power y and remainder by z

print(pow(4, 5, 5)) output pow is 1024 and then remainder is 4

**input( )** a = input(“Enter something”)

x = input("Enter a number: " ) output this work on prompt but not change data type

print(x)

a = float(input("Enter a real number :")) if run same data type then this procedure is right

**Control Flow**

**Condition statement**

**If condition**

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

**If else condition**

a = 200  
b = 33  
if b > a:  
  print("b is greater than a")  
else:  
  print("b is not greater than a")

**If elif else condition**

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")

**Nested if condition**

x = 41  
  
if x > 10:  
  print("Above ten,")  
  if x > 20:  
    print("and also above 20!")  
  else:  
    print("but not above 20.")

**Indentation**

Indentation refers to the spaces at the beginning of a code line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code.

**Loop in python**

**While Loop**

n = int(input("Max iteration:"))

i = 1

*while* i<n:

    print(i)

    i += 1

print("done")

**else statement in while loop**

n = int(input())

i = 1

*while* (i < n):

*if* i % 2 ==0:

        print("this is iteration number:", i)

        print(i)

*else*:

*pass*

    i += 1

print("loop done")

**Break and continue in while loop**

i = 1

*while* True:

*if* i % 17 == 0:

        print("break")

*break*

*else*:

        i += 1

        print("i am inside the loop")

*continue*

print("this loop is done")

**For Loop in Python**

L = []

for i in range(10):

print(i+1)

L.append(i\*\*2)

print(L)

fruits = ["apple", "banana", "cherry"]

*for* x *in* fruits:

  print(x)

**else in for loop**

*for* x *in* range(6):

  print(x)

*else*:

  print("Finally finished!")

**Exploring Dictionary in for loop**

D = {"a": 32, "b": 43, "c": -21,"d": "asdf"}

*for* x *in* D:

    print(x, D[x])

**Function in python**

Block of statement that perform a specific task.

def my\_function(*fname*):

    print(*fname* + " Rasheed")

my\_function("Haroon")

my\_function("Mamoon")

my\_function("Jawaria")

my\_function("Farah")

**Parameters and argument**

A parameter is the variable listed inside the parentheses in the function definition.

An argument is the value that is sent to the function when it is called.