# Tutorial 5 :

1. Familiar with Pycharm

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

HelloWorld.py

Text

Description automatically generated

1. Declaring variables in Python

Text

Description automatically generated

#Float

Text

Description automatically generated

#Boolean

A screenshot of a computer

Description automatically generated

#Strings

potter\_enemies = 9+1

name = 'Harry'

last\_name = 'Potter'

passcode = 'Platform@94/3'

address = 'hogwarts'

level = ''

location = " "

no\_of\_enemies = "10"

print(potter\_enemies)

print(type(potter\_enemies))

print(name)

print(type(name))

print(last\_name)

print(type(last\_name))

print(passcode)

print(type(passcode))

print(address)

print(type(address))

print(level)

print(type(level))

print(location)

print(type(location))

equality = (potter\_enemies == no\_of\_enemies)

print(equality)

print(type(equality))

Output:

Text

Description automatically generated

#None :

name = 'Harry '  
middle\_name = 'James '  
last\_name = 'Potter'  
fullname = name+middle\_name+last\_name  
print(fullname)  
middle\_name = None  
print(middle\_name)  
print(type(middle\_name))

Text

Description automatically generated

Variable conversion :

Int to float

Text

Description automatically generated

Float to int

a\_float = 3.0  
b\_float = 2.0  
answer = int(a\_float+b\_float)  
  
print(answer)  
print(type(answer))

Shape

Description automatically generated with medium confidence

Boolean to string

Text

Description automatically generated

Int and float to string

Text

Description automatically generated

String to int and float

Text

Description automatically generated

Collections in python :

Text

Description automatically generated

Output:

Text

Description automatically generated

Operations on collection

my\_list = [1, 2, 3, 4, 5]  
print(my\_list[0])  
print(my\_list[1:4])  
my\_list.append(6)  
print(my\_list)  
my\_list.remove(3)  
print(my\_list)  
  
my\_tuple = (1, 2, 3, 4, 5)  
print(my\_tuple[0])  
print(my\_tuple[1:4])  
  
my\_set = {1, 2, 3, 4, 5}  
my\_set.add(6)  
print(my\_set)  
my\_set.remove(3)  
print(my\_set)  
print(3 in my\_set)  
  
my\_dict = {'apple': 3, 'banana': 2, 'orange': 4}  
print(my\_dict['apple'])  
my\_dict['pear'] = 1  
print(my\_dict)  
del my\_dict['banana']  
print(my\_dict)

output :

Text

Description automatically generated

If statement :

Graphical user interface, text

Description automatically generated

While loop :

Small calculator program is used

import math  
while True:  
 print("Welcome! Which operation would you like to perform?")  
 print("+ for addition")  
 print("- for subtraction")  
 print("\* for multiplication")  
 print("/ for division")  
 print("% for modulus")  
 print("\*\* for exponential")  
 print("! for factorial")  
 print("Enter 0 to exit")  
  
 operation = input("Enter operation: ")  
  
 if operation == '0':  
 print("Exiting program...")  
 break  
  
  
 if operation == '+':  
 num1 = float(input("Enter first number: "))  
 num2 = float(input("Enter second number: "))  
 result = num1 + num2  
 print("Result: ", result)  
  
 elif operation == '-':  
 num1 = float(input("Enter first number: "))  
 num2 = float(input("Enter second number: "))  
 result = num1 - num2  
 print("Result: ", result)  
  
 elif operation == '\*':  
 num1 = float(input("Enter first number: "))  
 num2 = float(input("Enter second number: "))  
 result = num1 \* num2  
 print("Result: ", result)  
  
 elif operation == '/':  
 num1 = float(input("Enter first number: "))  
 num2 = float(input("Enter second number: "))  
 result = num1 / num2  
 print("Result: ", result)  
  
 elif operation == '%':  
 num1 = float(input("Enter first number: "))  
 num2 = float(input("Enter second number: "))  
 result = num1 % num2  
 print("Result: ", result)  
  
 elif operation == '\*\*':  
 num1 = float(input("Enter base number: "))  
 num2 = float(input("Enter exponent number: "))  
 result = num1 \*\* num2  
 print("Result: ", result)  
  
 elif operation == '!':  
 num = int(input("Enter a non-negative integer: "))  
 result = math.factorial(num)  
 print("Result: ", result)  
  
 else:  
 print("Invalid input! Please try again.")

sample outputs :

Text

Description automatically generated

For loop :

Small factorial program using for loop

num = int(input("Enter a number: "))  
fact = 1  
  
# loop from 1 to num and multiply the factors  
for i in range(1, num + 1):  
 fact \*= i  
  
print("Factorial of", num, "is", fact)

output :

Graphical user interface, application

Description automatically generated