Week 2 - Data Types and Fucntions

Name: Keh Yong Hao

Matric No: 23053543

Integer

In [1]:
```
x = 1
y = 3
print(type(x))
print(type(y))
```

```
<class 'int'>
<class 'int'>
```

Float

In [2]:
```
a = 1.2
b = 2.3333
print(type(a))
print(type(b))
```

```
<class 'float'>
<class 'float'>
```

String

In [3]:
```
print("Hello World")
```

```
Hello World
```

In [5]:
```
a = "pizza"
print(a)
```

```
pizza
```

In [7]:
```
a = """ HI
lmao"""
print (a)
```

```
 HI
lmao
```

Boolean

In [9]:
```
#10, 5
print(10>5)
print(10==5) # equals -> '=='
print(10<5)
```

```
True
False
False
```

List

In [13]:
```
thelist = ["apple", "banana", "cherry"] # []
print(thelist)
print(type(thelist))
```

```
['apple', 'banana', 'cherry']
<class 'list'>
```

Tuple

In [12]:
```
thistuple = ("apple", "banana", "cherry") #()
print(thistuple)
print(type(thistuple))
```

```
('apple', 'banana', 'cherry')
<class 'tuple'>
```

Range

In [14]:
```
x = range(6)
for n in x:
    print(n)
```

```
0
1
2
3
4
5
```

Sets

In [15]:
```
thisset = ("apple", "banana", "cherry")
print(thisset)
```

```
('apple', 'banana', 'cherry')
```

Frozenset

In [16]:
```
thisset1 = ("apple", "banana", "cherry")
x = frozenset(thisset)
x[1] = "strawberry"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[16], line 3
      1 thisset1 = ("apple", "banana", "cherry")
      2 x = frozenset(thisset)
----> 3 x[1] = "strawberry"

TypeError: 'frozenset' object does not support item assignment
```

Dictionaries

In [17]:
```
# John is 25 yo. He is not a student
```

In [21]:
```
dict1 = {"name": "John" ,
        "age" : 25,
        "is_not_student": True}
print(dict1)
print(dict1["name"])
```

```
{'name': 'John', 'age': 25, 'is_not_student': True}
John
```

Byte

In [23]:
```
a = b"hello"
print(a)
for byte in a:
    print(byte)
```

```
b'hello'
104
101
108
108
111
```

In [25]:
```
bytearraytext = bytearray([104, 101, 108, 108, 111])
print(bytearraytext)
```

```
bytearray(b'hello')
```

Memoryview

In [26]:
```
memviewText = memoryview(a)
print(memviewText)
```

```
<memory at 0x000002B4B9583C40>
```

Functions

In [30]:
```
def sum(a,b):
    return a + b
```

In [31]:
```
c = 3
d = 6
add = sum(c,d)
print(add)
```

```
9
```

Exercise 1

In [33]:
```
def sum(a,b):
    return a + b

def sub(a,b):
    return a - b

def mult(a,b):
    return a * b

def div(a,b):
    return a / b

print(mult(50, 10))
print(div(100,20))
print(sum(5,2))
```

```
500
5.0
7
```

Exercise 2

In [37]:
```
subject = ["Programming", "Chemistry", "Maths", "English"]

def sub_availability(subject_name):
    if(is_available(subject_name)):
        return "the subject exist"
    else:
        return "the subject does not exist"
def is_available(subject_name):
    return subject_name in subject
```

```
input_sub = input("Subject name?")
print(sub_availability(input_sub))
```

```
the subject exist
```

In [ ]: