In [1]:

```
print("hello world")
```

```
hello world
```

In [2]:

```
print("welcome to my class")
```

```
welcome to my class
```

Data Type

In [7]:

```
a = "python"
type(a)
```

Out[7]:

```
str
```

python is case sensitive

In [8]:

```
b = 5.2
type(b)
```

Out[8]:

```
float
```

In [9]:

```
c = 3
type(c)
```

Out[9]:

```
int
```

In [10]:

```
d = 3+2j
type(d)
```

Out[10]:

```
complex
```

In [11]:

```
x = False
type(x)
```

Out[11]:

```
bool
```

In [12]:

```
x = "True"
type(x)
```

Out[12]:

```
str
```

Rules for naming the variable

In [15]:

```
fruit = "appple","orange"
fruit
```

Out[15]:

```
('appple', 'orange')
```

1. variable name should not start with a number

2 . variable name does not allow spacial char other then underscore(_)

3 . space is not allowed;

4. allow with upper case ;

In [17]:

```
fruit = "appple","orange"
```

DATA STRUCTURE

```
In [18]: x = "apple ", "banana ", " cheery "
         type(x)
```

```
Out[18]: tuple
```

```
In [19]: x = ["apple ", "banana ", " cheery "]
         type (x)
```

```
Out[19]: list
```

```
In [20]: x = {"name": "kesahv" , "age" : 17}
         type(x)
```

```
Out[20]: dict
```

```
In [23]: x = {"apple", "banana ", " cheery "}
         type(x)
```

```
Out[23]: set
```

LIST

1. allow hetrogeneous; (means it allow string , num , float , complex nu) etc

2. index start from 0 ;

3 . Mutable using index;(means we change this using the index)

4 .

```
In [25]: sample_list = [12 , 254 , 65 , 465, "hi"]
         sample_list
```

```
Out[25]: [12, 254, 65, 465, 'hi']
```

```
In [26]: type(sample_list)
```

```
Out[26]: list
```

```
In [28]: sample_list[0]
```

```
Out[28]: 12
```

```
In [29]: sample_list[-1]
```

```
Out[29]: 'hi'
```

```
In [30]: sample_list[2] = 4
```

```
In [31]: sample_list
```

```
Out[31]: [12, 254, 4, 465, 'hi']
```

TUPLES

1 . Allow Heterogenous;

2 , index starts from 0 and Retrieve using index(retrieve means

3 . Cannot change data once which is decleard = immutable

4.()

```
In [38]: sample_tuple = ( 1 , 3.4 , "hii" , 1+2j)
         sample_tuple

Out[38]: (1, 3.4, 'hii', (1+2j))
```

```
In [39]: type(sample_tuple)

Out[39]: tuple
```

```
In [40]: sample_tuple[3]

Out[40]: (1+2j)
```

SET

1. Add heterogenous

2 . Does not allow duplicate

3 . cannot access using index

4 . immutable using but it is mutable

5 . ordered-first place

```
In [2]: sample_set= { 1,334,55,67,32,2114,555,7676,32,21324,43534,345345,43534,"hii","hell
```

```
In [3]: sample_set

Out[3]: {1, 2114, 21324, 32, 334, 345345, 43534, 55, 555, 67, 7676, 'hello', 'hii'}
```

```
In [4]: sample_set[2]### we cannot retrieve the elements using index in a set;

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_5460\2739957020.py in <module>
----> 1 sample_set[2]### we cannot retrieve the elements using index in a set;

TypeError: 'set' object is not subscriptable
```

```
In [5]: sample_set.remove("hii")
```

```
In [6]: sample_set

Out[6]: {1, 2114, 21324, 32, 334, 345345, 43534, 55, 555, 67, 7676, 'hello'}
```

```
In [7]: sample_set.add("keshav")
```

```
In [8]: sample_set

Out[8]: {1, 2114, 21324, 32, 334, 345345, 43534, 55, 555, 67, 7676, 'hello', 'keshav'}
```

DICTIONARY

1 .dictionary has key value pair datastructure

```
###### 2 . key is unidue and values can be duplicate
    ######3. can retrieve the value using key
        ######4. can change the value using key
            ######5. key is immutable
```

```
In [10]: sample_dict = {"a":"alpha" , 1 :"first", 2: "second",3:"third","b" :"beta"}
```

```
In [11]: type(sample_dict)
```

```
Out[11]: dict
```

```
In [12]: sample_dict
```

```
Out[12]: {'a': 'alpha', 1: 'first', 2: 'second', 3: 'third', 'b': 'beta'}
```

```
In [13]: sample_dict["a"]
```

```
Out[13]: 'alpha'
```

```
In [ ]:
```