In [ ]:

```
# shortcuts
# esc+m ->To convert cell into markdown cell
# esc+ y -> To convert into coding cell
# esc +a -> To create a cell above
# esc+ b -> To create a cell below
# esc +h -> shortcuts
# shift+enter -> To run the cells

```

Type *Markdown* and LaTeX: α^2

In [2]:

```
# ipynb -> interactive python notebook
```

largest Heading

second largest heading

H3

H4

H5

smallest heading

styling text

This is bold text *This is italic text* ***This is bold and italic*** ~~~ This is strike through~~

Unordered list

- City
- vijayawada
- guntur
- Tirupathi

Ordered list

- Department

1. ECE
2. CSE
3. IT
4. EEE

5. MECH
6. CIVIL

Creating a link

- Google.com (https://www.google.com)
- facebook.com (https://www.facebook.com)

In [5]:

```
# comments in python
# single line comment
"""Multi
    Line
        Comment"""
```

Out[5]:

```
'Multi \n    Line\n        Comment'
```

In [6]:

```
# Datatypes in python
# int,float,string,bool

a=90
print(type(a))
```

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

Data structures

- List
- Tuple
- Dictionary
- Sets

List

- list is a collection of elements, created by using square brackets[]
- list elements are seperated by commas ,
- List is mutable i.e changeable.

In [2]:

```
li=[45,"hello",89.7] # mixed datatypes
print(li)
print(type(li))
```

```
[45, 'hello', 89.7]
<class 'list'>
```

In [11]:

```
l=[10,20,30,40,50]
print(l)
print(type(l))
print(max(l))
print(min(l))
print(sum(l))
print(sorted(l))
print(sorted(l,reverse=True))
```

```
[10, 20, 30, 40, 50]
<class 'list'>
50
10
150
[10, 20, 30, 40, 50]
[50, 40, 30, 20, 10]
```

In [5]:

```
# accessing list elements
# 10 20 30 40 50
# 0   1  2  3  4  # forward indexing
# -5  -4 -3 -2 -1 # backward indexing

```

In [9]:

```
print(l[0])
print(l[-5])
print(l[1])
print(l[2])
print(l[3])
print(l[4])
print(l[-1])
```

```
10
10
20
30
40
50
50
```

In [10]:

```
### reverse of a list
l[::-1]
```

Out[10]:

```
[50, 40, 30, 20, 10]
```

In [40]:

```
# slicing [start:stop:step]

l=[10,20,30,40,50]
# [10,20]
print(l[0:2:1])
# [30,40]
print(l[2:4:1])
```

...

In [25]:

```
# concatenation +
l1=[1,2,3]
l2=[4,5,6]
l1+l2
```

...

In [26]:

```
# repetition *
l1*5
```

Out[26]:

```
[1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
```

In [27]:

```
# list is mutable
l=[10,20,30]
print(l)
l[0]=101
print(l)
```

```
[10, 20, 30]
[101, 20, 30]
```

In [28]:

```
print(dir(l),end=" ")
```

```
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__',
'__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__get
attribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '_
_init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__
mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__re
versed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__',
'__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'i
nsert', 'pop', 'remove', 'reverse', 'sort']
```

In [29]:

```
l
```

Out[29]:

```
[101, 20, 30]
```

In [30]:

```
l.append(102)
```

In [31]:

```
l
```

Out[31]:

```
[101, 20, 30, 102]
```

In [32]:

```
l.extend([90,91,92])
print(l)
```

```
[101, 20, 30, 102, 90, 91, 92]
```

In [35]:

```
l=[1,2,3]
l.reverse()
print(l)
```

```
[3, 2, 1]
```

In [36]:

```
l=[10,20,30,10,20]
l.count(10)
```

...

In [37]:

```
l=[10,20,30]
l2=l.copy()
print(l2)
```

...

Tuple

- Tuple is a collection of objects
- tuple uses parenthesis
- tuple is immutable

In [42]:

```
# creating a tuple
t=(10,20,30)
print(t)
print(type(t))
print(len(t))
print(max(t))
print(min(t))
print(sum(t))
```

```
(10, 20, 30)
<class 'tuple'>
3
30
10
60
```

In [43]:

```
# accessing tuple
print(t[0])
print(t[-1])
```

...

In [44]:

```
# slicing

t[::2] # alternate elements
```

Out[44]:

```
(10, 30)
```

In [45]:

```
t1=(1,2,3)
t2=(9,8,7)
t1+t2
```

Out[45]:

```
(1, 2, 3, 9, 8, 7)
```

In [46]:

```
t1*6
```

...

In [47]:

```
# membership operator in

10 in t
```

...

In [48]:

```
# iterating through a tuple
for i in t:
    print(i)
```

...

In [49]:

```
print(dir(t),end=" ")
```

...

In [50]:

```
t=(11,23,2,67,87,23)
t.count(23)
```

...

In [51]:

```
t.index(11)
```

...

Dictionary

- Dictionary is a collection of items or set of key and value pair
- in dictionary keys acts as a index

In [73]:

```
# creating a dictionary

d={1:"rama",2:"geetha",3:"madhu"}
```

In [74]:

```
d.keys() # to get only keys

```

Out[74]:

```
dict_keys([1, 2, 3])
```

In [75]:

```
d.values() # to get only values
```

Out[75]:

```
dict_values(['rama', 'geetha', 'madhu'])
```

In [76]:

```
d.items() # to get items
```

Out[76]:

```
dict_items([(1, 'rama'), (2, 'geetha'), (3, 'madhu')])
```

In [77]:

```
d
```

Out[77]:

```
{1: 'rama', 2: 'geetha', 3: 'madhu'}
```

In [78]:

```
# accessing dictionary values
print(d[1])  # d[key]
print(d[2])
```

```
rama
geetha
```

In [79]:

```
# changing dictionary values

d[1]='Mouni'
print(d)
```

```
{1: 'Mouni', 2: 'geetha', 3: 'madhu'}
```

In [80]:

```
# adding items to dictionary
d[4]="Lakshmi"
d
```

Out[80]:

```
{1: 'Mouni', 2: 'geetha', 3: 'madhu', 4: 'Lakshmi'}
```

In [81]:

```
# iterating through a dictionary

for k,v in d.items():
    print(k,v)
```

```
1 Mouni
2 geetha
3 madhu
4 Lakshmi
```

In [82]:
```
for k in d.keys():
    print(k)
```

```
1
2
3
4
```

In [83]:
```
for v in d.values():
    print(v)
```

```
Mouni
geetha
madhu
Lakshmi
```

In [84]:
```
print(dir(d),end=" ")
```

```
['__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem_
_', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute_
_', '__getitem__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__
ior__', '__iter__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__or
__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__ror__', '_
_setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'cle
ar', 'copy', 'fromkeys', 'get', 'items', 'keys', 'pop', 'popitem', 'setdefaul
t', 'update', 'values']
```

In [85]:
```
d.update({4:"Laxmi"})
print(d)
```

```
{1: 'Mouni', 2: 'geetha', 3: 'madhu', 4: 'Laxmi'}
```

In [86]:
```
d.get(4)  # get(key)
```

Out[86]:
```
'Laxmi'
```

In [68]:
```
d.clear()
```

In [69]:
```
d
```

Out[69]:
```
{}
```

In [70]:
```
del d
```

In [71]:

```
d
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_4184\3161387801.py in <module>
----> 1 d

NameError: name 'd' is not defined
```

In [ ]:

```

```