Heading1

Heading2

Heading3

Heading4

Heading5

Heading6

Bold

italic

Bold and italic

- Normal Text
  - sublist1
  - sublist2
  - suvlist3

Comments in python

- Single line comment
- Multi line comment

In [1]:

```
# single line comment
print("Python Programming")
```

```
Python Programming
```

In [2]:

```
"""
Multi
line
comments
"""
print("Hai, everyone")
```

```
Hai, everyone
```

In [4]:

```
print("This is keerthi's meeting")
```

```
This is keerthi's meeting
```

In [5]:

```
print('This is keerthi\'s meeting')
```

```
This is keerthi's meeting
```

In [6]:

```
print('c:docs\navin')
```

```
c:docs
avin
```

In [7]:

```
print('c:docs\\navin')
```

```
c:docs\navin
```

In [10]:

```
print("apssdc "*100)
```

```
apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc
apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc
apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc
apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc
apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc
apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc
apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc
apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc
apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc apssdc
apssdc
```

Variables

- A python variable is a reserved memory location to store values.

In [11]:

```
# single variable assignment
a = 23
print(a)
```

```
23
```

In [12]:

```
# multiple variable assignment with multiple values
x,y,z = 12,14,16
print(x,y,z)
```

```
12 14 16
```

In [13]:

```
print(x, '\n', y, '\n', z)
```

```
12
 14
 16
```

In [14]:

```
# multiple variable assignment
m = 134
a=b=c=m
print(b)
print(a)
```

```
134
134
```

Python keywords

- Keywords are the reserved words. We can not use a keyword as a varibale name, function name, or any other identifiers.

In [15]:

```
import keyword
print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'fo
r', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'no
t', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

Data types and conversions

1. int
2. float

3. string

In [16]:

```
n = 12
print(n)
```

```
12
```

In [17]:

```
type(n)
```

Out[17]:

```
int
```

In [18]:

```
f = 9.5
print(type(f))
```

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

In [19]:

```
s = 'kits'
print(type(s))
```

```
<class 'str'>
```

In [20]:

```
f = 13.8
print(int(f))
```

```
13
```

In [21]:

```
n = 13
print(float(n))
```

```
13.0
```

In [22]:

```
s = '123'
print(type(s))
```

```
<class 'str'>
```

In [23]:

```
print(int(s))
```

```
123
```

In [24]:

```
s1 = 'apssdc'
print(int(s1))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-24-1701b9f937a8> in <module>
      1 s1 = 'apssdc'
----> 2 print(int(s1))

ValueError: invalid literal for int() with base 10: 'apssdc'
```

In [33]:

```
f = '23.9'
print(type(f))
res = float(f)
print(res)
print(type(res))
print(int(res))
```

```
<class 'str'>
23.9
<class 'float'>
23
```

In [34]:

```
for i in range(1,11):
    print(i,end=" ")
```

```
1 2 3 4 5 6 7 8 9 10
```

In [37]:

```
for i in range(1,11):
print(i,end=" ")
```

```
  File "<ipython-input-37-85dfd068cd98>", line 2
    print(i,end=" ")
    ^
IndentationError: expected an indented block
```

In [ ]:

```

```