```
# heading1
## heading2
#### heading4
```

heading1
heading2
heading4

```
**bold**
```

bold

```
*italic*
```

italic

```
***bold and italic***
```

bold and italic

```
* normal text
    * sublist1
    * sublist2
    * sublist3
```

* normal text
    * sublist1
    * sublist2
    * sublist3

comments in python

1. single line comment
2. multi line comment

In [6]:

```
# single line comment

"""
multi
line
comment

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

```
hello
```

In [12]:

```
print("nidhi"*5)
```

```
nidhinidhinidhinidhinidhi
```

In [13]:

```
print("nidhi\n"*3)
```

```
nidhi
nidhi
nidhi
```

In [14]:

```
print("nidhi\t"*2)
```

```
nidhi	nidhi
```

variables

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

In [15]:

```
a = 123
print(a)
```

```
123
```

In [16]:

```
a = 12
print(a)
print("a")
```

```
12
a
```

In [17]:

```
a="apssdc"
print(a)
```

```
apssdc
```

In [20]:

```
x,y,z=12,9,10  # multiple variable assignment
print(x,y,z)
print(x,'\n',y,'\n',z,'\n')
```

```
12 9 10
12
 9
 10
```

python keyword

- keywords are reserved words in python. we cannot use keywords as variable name, function name or any other identifier.

In [21]:

```
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']
```

In [27]:

```
# len = keyword.kwlist  here len = will not be displayed as we are giving len keyword as va
```

In [1]:

```
import keyword
print(keyword.kwlist)
l = keyword.kwlist
print(len(l))
```

```
['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']
35
```

In [ ]:

```

```