

```
python.py
1  #print Hello World
2
3  print("Hello World")
```

```
python.py > ...
1  x=1
2  # initial value of x is 1
3  if x > 0:
4      print("These are two comments")
5  # printing a string
```

python.py

```
1 print("Statement1")
2 print("Statement2")
3
4 # You can write these two statemnets in following way
5
6 print("Statement1");print("Statement2")
```

python.py > ...

```
1 x=1
2
3 # Syntax error expected indentation
4
5 if x>0:
6 print("This statement has no indentation")
7 print("This statement has no indentation")
```

python.py > ...

1 x=1

2

3 # Single space indentation

4

5 if x>0:

6 | print("This statement has a single space indentation")

7 | print("This statement has a single space indentation")

python.py > ...

1 x=1


2

3 # Single tab indentation

4

5 if x>0:

6 | print("This statement has a single tab indentation")

7 |  print("This statement has a single tab indentation")

python.py > ...

1 x=1

2

3 # Single space+tab indentation

4

5 if x>0:

6 print("This statement has a single space+tab indentation")

7 print("This statement has a single space+tab indentation")

```
python.py > ...
1  # Type of the variable
2
3  a=1452
4  print(type(a))
5
6  # Output: <class 'int'>
7
8  b=(-4587)
9  print(type(b))
10
11 # Output: <class 'int'>
12
13 c=0
14 print(type(c))
15
16 # Output: <class 'int'>
17
18 g=1.03
19 print(type(g))
20
21 # Output: <class 'float'>
22
23 i=.34
24 print(type(i))
25
26 # Output: <class 'float'>
27
28 j=2.21e-10
29 print(type(j))
30
31 # Output: <class 'float'>
32
33 k=5E220
34 print(type(k))
35
36 # Output: <class 'float'>
```

python.py > ...

```
1  # Complex Numbers
2
3  x = complex(1,2)
4  print(type(x))
5  print(x)
6
7  # Output: <class 'complex'>
8  # Output: (1+2j)
9
10 y=1+2j
11 print(type(y))
12
13 # Output: <class 'complex'>
14
15 z=1+2j
16 print(type(z))
17
18 # Output: <class 'complex'>
```

```
python.py > ...  
1  # Boolean  
2  
3  x = True  
4  print(type (x))  
5  
6  # Output: <class 'bool'>  
7  
8  y = False  
9  print(type (y))  
10  
11  #Output: <class 'bool'>
```

python.py > ...

```
1  # String
2
3  #Strings start and end with double quotes
4  str1= "String"
5  print (str1)
6
7  # Output: String
8
9  #Strings start and end with single quotes
10 str2 = 'String'
11 print (str2)
12
13 #Output: String
14
15 #Strings start with double quote and end with single quote
16 str3 = "String'
17 # SyntaxError
18
19 #Stringa start with single quote and end with double quote
20 str4 = 'String"
21 # SyntaxError
22
23 #Single quote within double quotes
24 str2 = "Day's"
25 print (str2)
26
27 #Output: Day's
28
29 #Double quote within single quotes
30 str2 = 'Day"s'
31 print (str2)
32
33 #Output: Days"s
```


python.py

```
1  # Special character in string
2
3  print("This is a backslash (\\) mark.")
4
5  # Output: This is a backslash (\) mark.
6
7  print("This is tab \t key")
8
9  # Output: This is tab      key
10
11 print("These are \'single qoutes\'")
12
13 # Output: These are 'single qoutes'
14
15 print("These are \"doule qoutes\"")
16
17 # Output: These are "doule qoutes"
18
19 print("This is new line \nNew line")
20
21 # Output: This is new line
22 #      New line
```

python.py > ...

```
1  # Accessing elements in string
2  string1 = "PYTHON TUTORIAL"
3
4  # Print first character
5  print(string1[0])
6
7  # Output: P
8
9  # Print first character
10 print(string1[-15])
11
12 # Output: P
13
14 # Print last character
15 print(string1[14])
16
17 # Output: L
18
19 # Print last character
20 print(string1[-1])
21
22 # Output: L
23
24 # Print 4th character
25 print(string1[3])
26
27 # Output: H
28
29 # Print first character
30 print(string1[-12])
31
32 # Output: H
33
34 print(string1[16])
35
36 # Output: Traceback (most recent call last)
37 # IndexError: string index out of range
```

python.py > ...

```
1  # Creating list
2
3  # list contain all itegers
4  my_list1 = [5,12,13,14]
5  print(my_list1)
6
7  # Output: [5, 12, 13, 14]
8
9  # list contain all string
10 my_list2 = ['red','blue','black','white']
11 print(my_list2)
12
13 # Output: ['red', 'blue', 'black', 'white']
14
15 # list contain a string, an itegers and a float
16 my_list3 = ['red',12,112.12]
17 print(my_list3)
18
19 # Output: ['red', 12, 112.12]
```

python.py > ...

```
1  # Creating list
2
3  color_list = ["RED", "Blue", "Green", "Black"]
4  print(color_list[0])
5
6  # Output: RED
7
8  print(color_list[0],color_list[3])
9
10 # Output: RED Black
11
12 print(color_list[-1])
13
14 # Output: Black
15
16 print(color_list[4])
17
18 # Output: IndexError: list index out of range
```

python.py > ...

```
1  # List Slicing
2
3  color_list = ["RED", "Blue", "Green", "Black"]
4
5  print(color_list[0:2])
6
7  # Output: ['RED', 'Blue']
8
9  print(color_list[1:2])
10
11 # Output: ['Blue']
12
13 print(color_list[1:-2])
14
15 # Output: ['Blue']
16
17 print(color_list[:3])
18
19 # Output: ['RED', 'Blue', 'Green']
20
21 print(color_list[:])
22
23 # Output: ['RED', 'Blue', 'Green', 'Black']
```