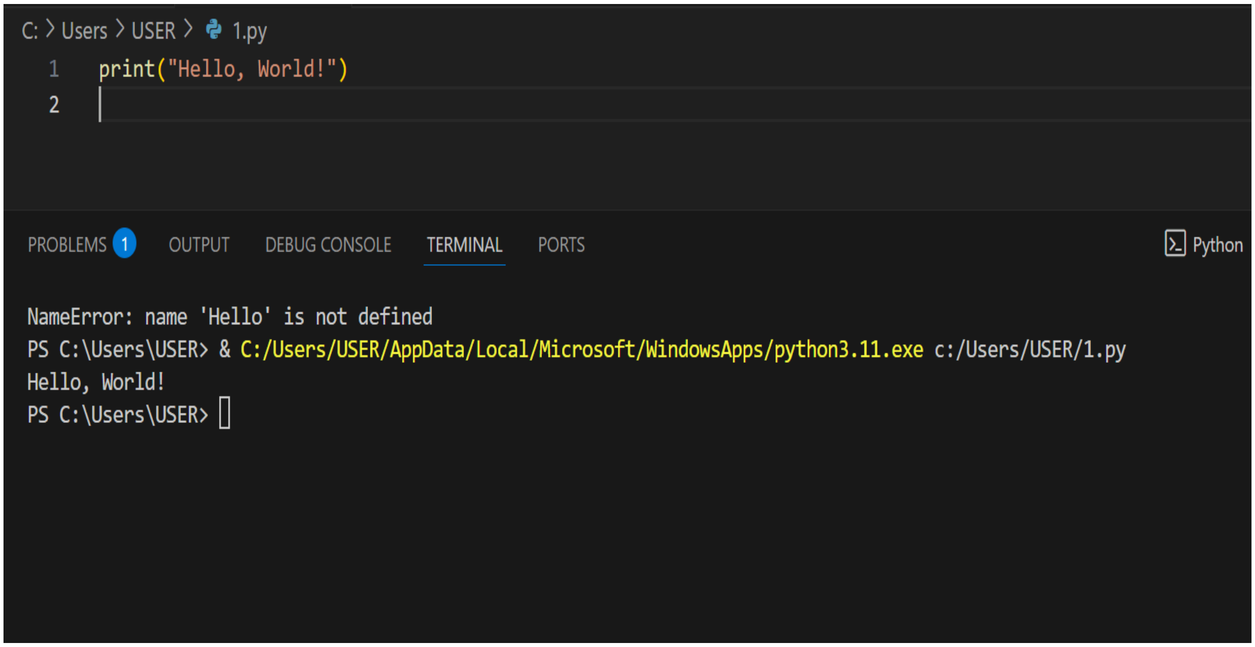
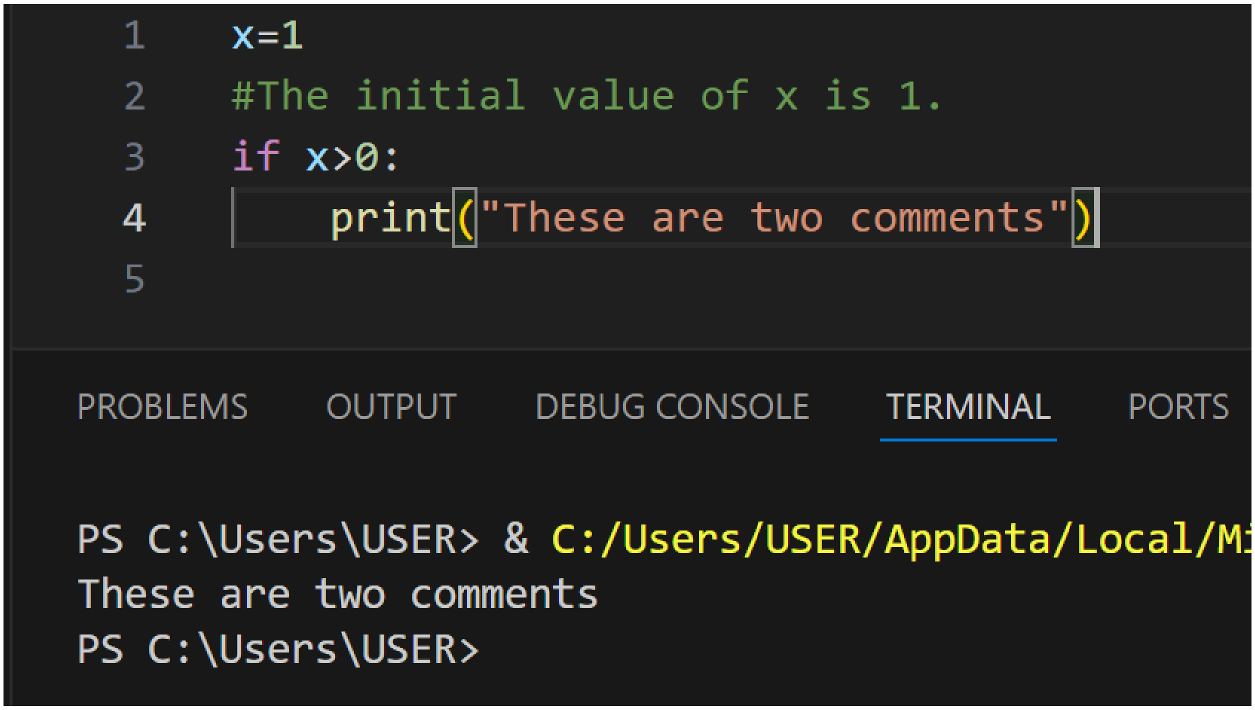
Simple Command:



```python
print("Hello, World!")

```

```
NameError: name 'Hello' is not defined
PS C:\Users\USER> & C:/Users/USER/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/USER/1.py
Hello, World!
PS C:\Users\USER> 
```

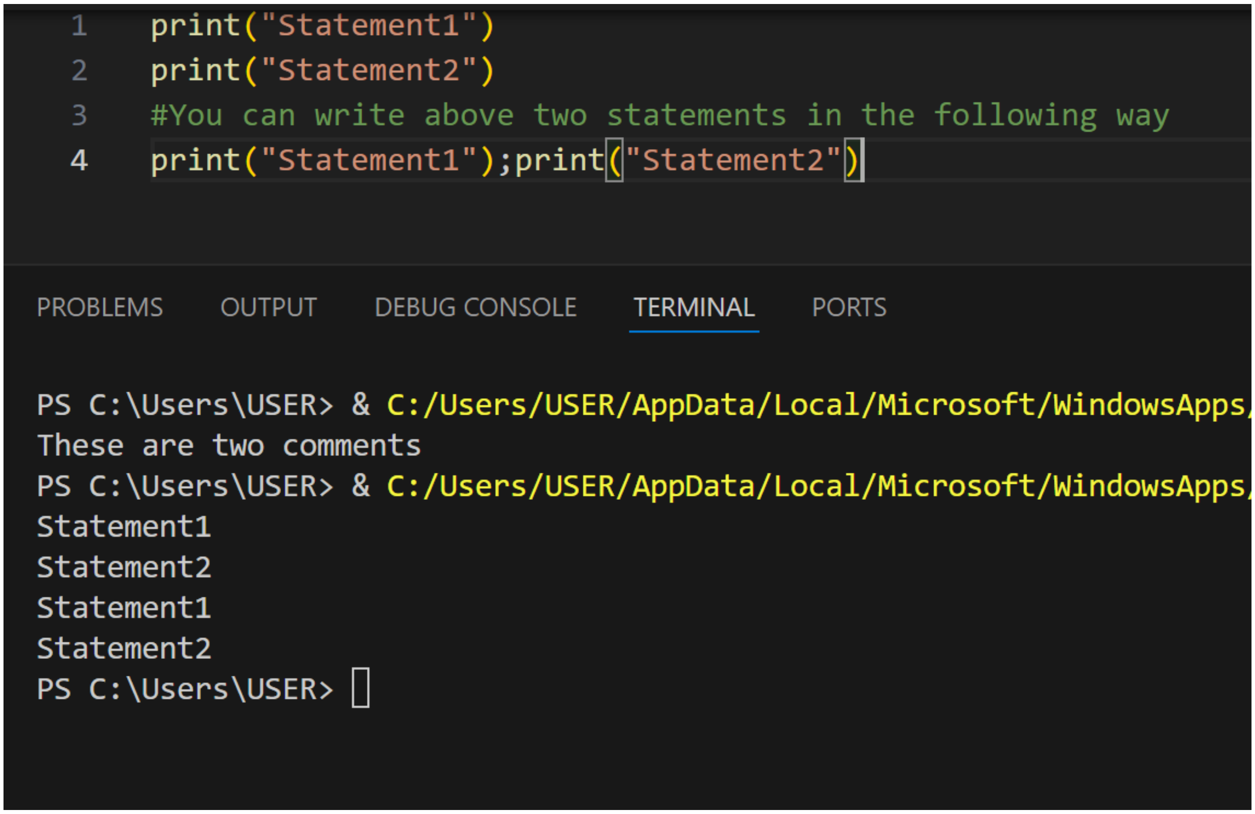
Comments in Python:



```python
x=1
#The initial value of x is 1.
if x>0:
    print("These are two comments")

```

```
PS C:\Users\USER> & C:/Users/USER/AppData/Local/Mi
These are two comments
PS C:\Users\USER>
```

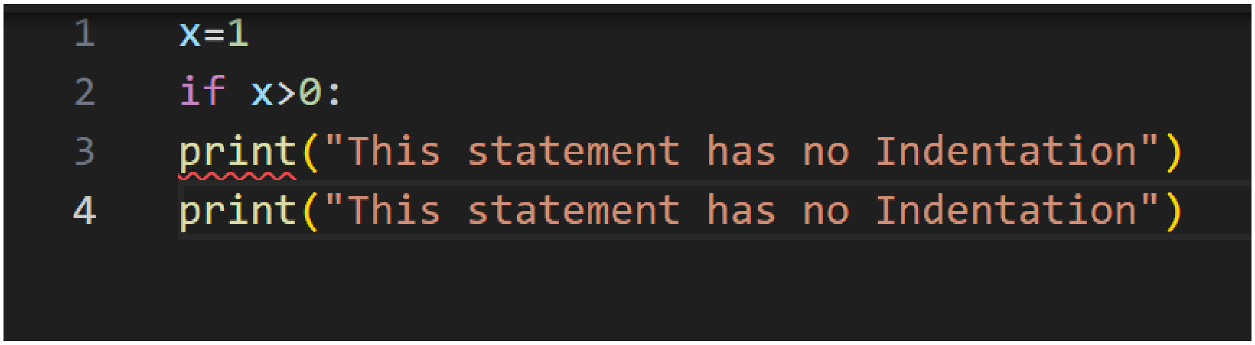
MULTIPLE STATEMENTS ON SINGLE LINE:



```python
print("Statement1")
print("Statement2")
#You can write above two statements in the following way
print("Statement1");print("Statement2")
```

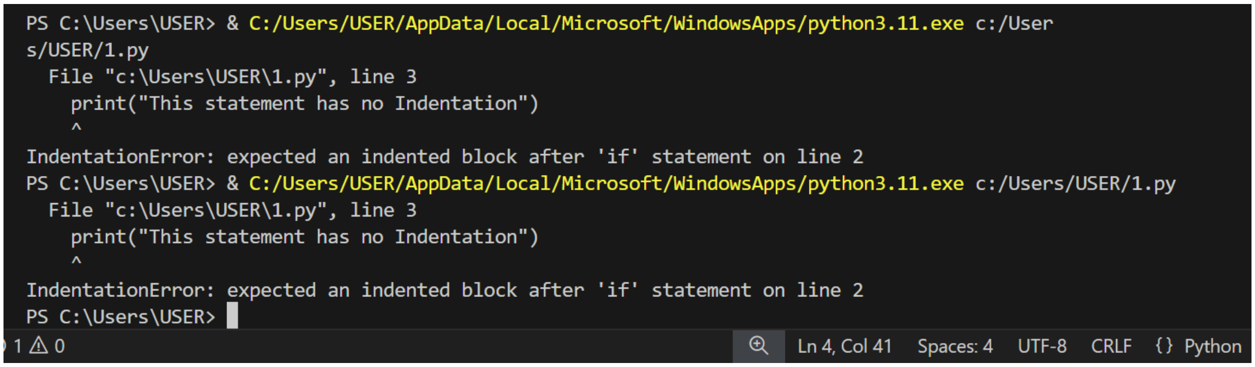
```
PS C:\Users\USER> & C:/Users/USER/AppData/Local/Microsoft/WindowsApps/
These are two comments
PS C:\Users\USER> & C:/Users/USER/AppData/Local/Microsoft/WindowsApps/
Statement1
Statement2
Statement1
Statement2
PS C:\Users\USER> 
```

INDENTATION:



```python
x=1
if x>0:
print("This statement has no Indentation")
print("This statement has no Indentation")
```

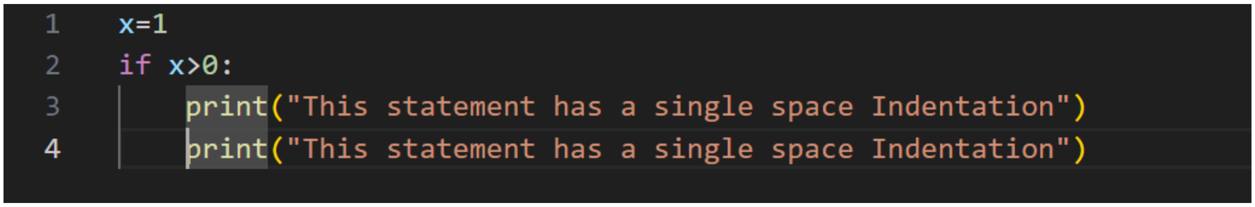
OUTPUT:



```
PS C:\Users\USER> & C:/Users/USER/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/User
s/USER/1.py
  File "c:\Users\USER\1.py", line 3
    print("This statement has no Indentation")
    ^
IndentationError: expected an indented block after 'if' statement on line 2
PS C:\Users\USER> & C:/Users/USER/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/USER/1.py
  File "c:\Users\USER\1.py", line 3
    print("This statement has no Indentation")
    ^
IndentationError: expected an indented block after 'if' statement on line 2
PS C:\Users\USER>
```

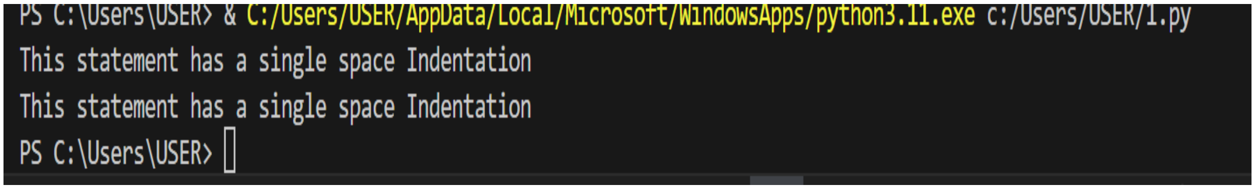
IndentationError: unexpected indent

INDENTATION WITH SINGLE TAB:



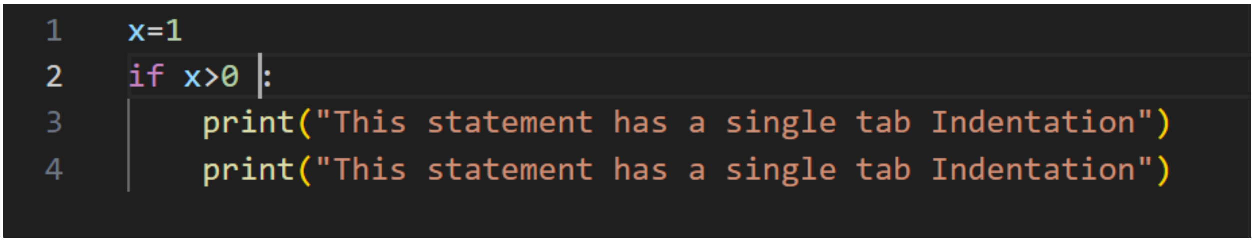
```
x=1
if x>0:
      print("This statement has a single space Indentation")
      print("This statement has a single space Indentation")
```

OUTPUT:



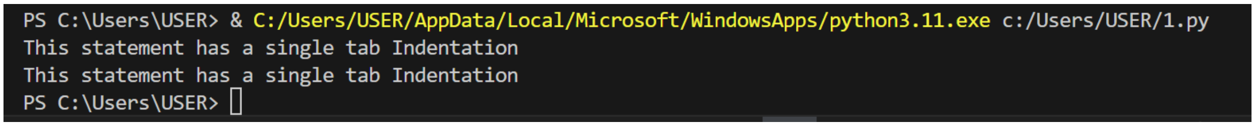
```
PS C:\Users\USER> & C:/Users/USER/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/USER/1.py
This statement has a single space Indentation
This statement has a single space Indentation
PS C:\Users\USER>
```

INDENTATION WITH SINGLE TAB:



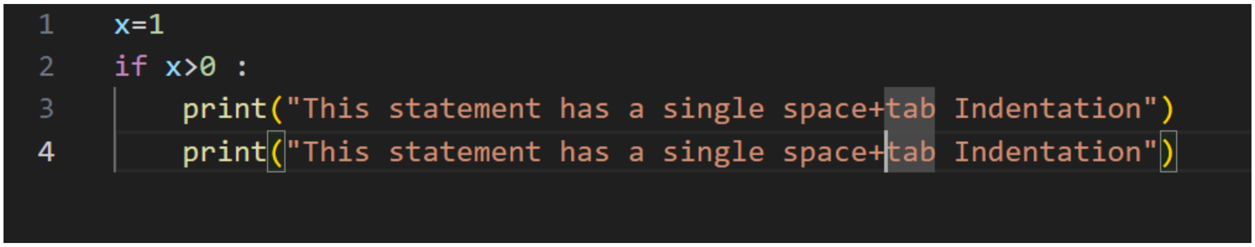
```
x=1
if x>0 :
    print("This statement has a single tab Indentation")
    print("This statement has a single tab Indentation")
```

OUTPUT:



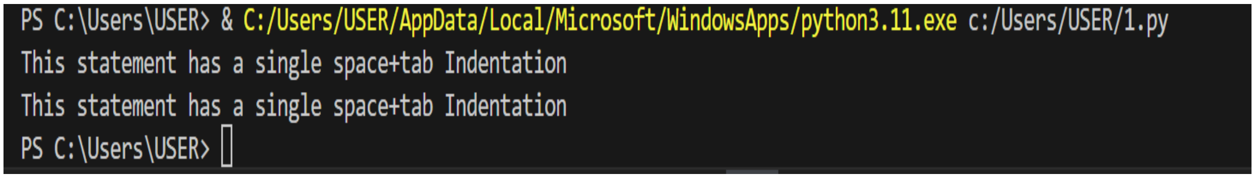
```
PS C:\Users\USER> & C:/Users/USER/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/USER/1.py
This statement has a single tab Indentation
This statement has a single tab Indentation
PS C:\Users\USER> 
```

INDENTATION WITH SINGLE TAB + SPACE:



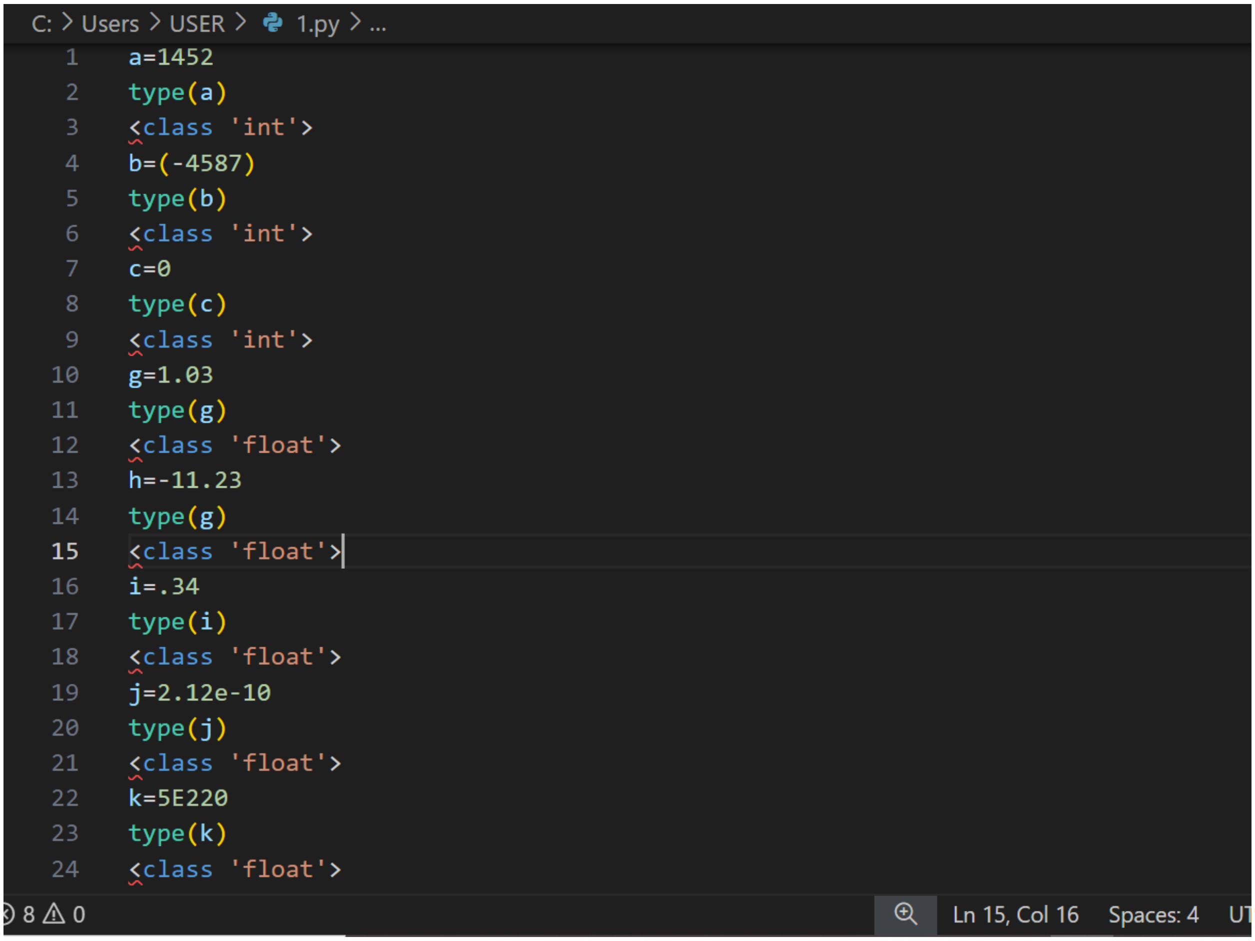
```python
x=1
if x>0 :
    print("This statement has a single space+tab Indentation")
    print("This statement has a single space+tab Indentation")
```

OUTPUT:



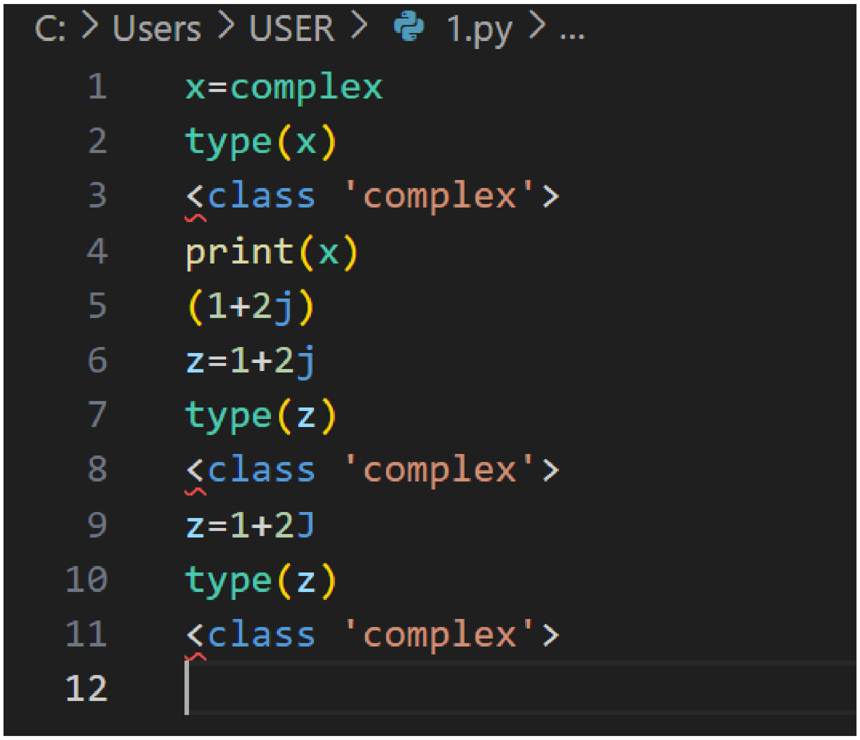
```
PS C:\Users\USER> & C:/Users/USER/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/USER/1.py
This statement has a single space+tab Indentation
This statement has a single space+tab Indentation
PS C:\Users\USER> 
```

DATA TYPES AND TYPE CASTINGS:



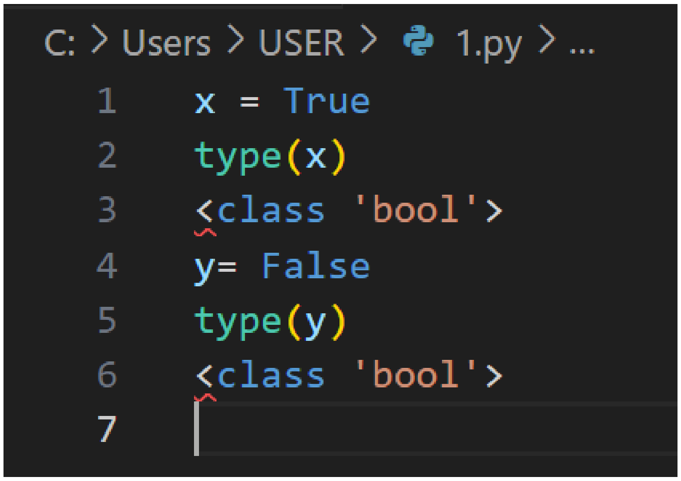
```
a=1452
type(a)
<class 'int'>
b=(-4587)
type(b)
<class 'int'>
c=0
type(c)
<class 'int'>
g=1.03
type(g)
<class 'float'>
h=-11.23
type(g)
<class 'float'>
i=.34
type(i)
<class 'float'>
j=2.12e-10
type(j)
<class 'float'>
k=5E220
type(k)
<class 'float'>
```

DATA TYPES AND TYPE CASTINGS- COMPLEX NUMBERS:



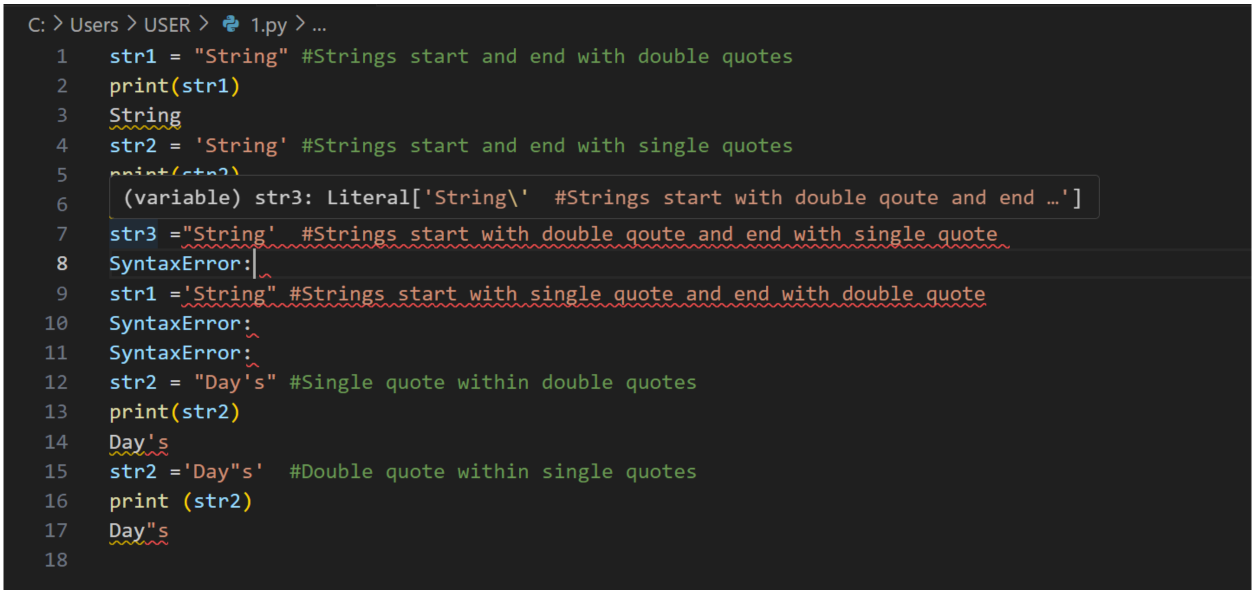
```
x=complex
type(x)
<class 'complex'>
print(x)
(1+2j)
z=1+2j
type(z)
<class 'complex'>
z=1+2J
type(z)
<class 'complex'>

```

DATA TYPES AND TYPE CASTINGS – BOOLEAN:



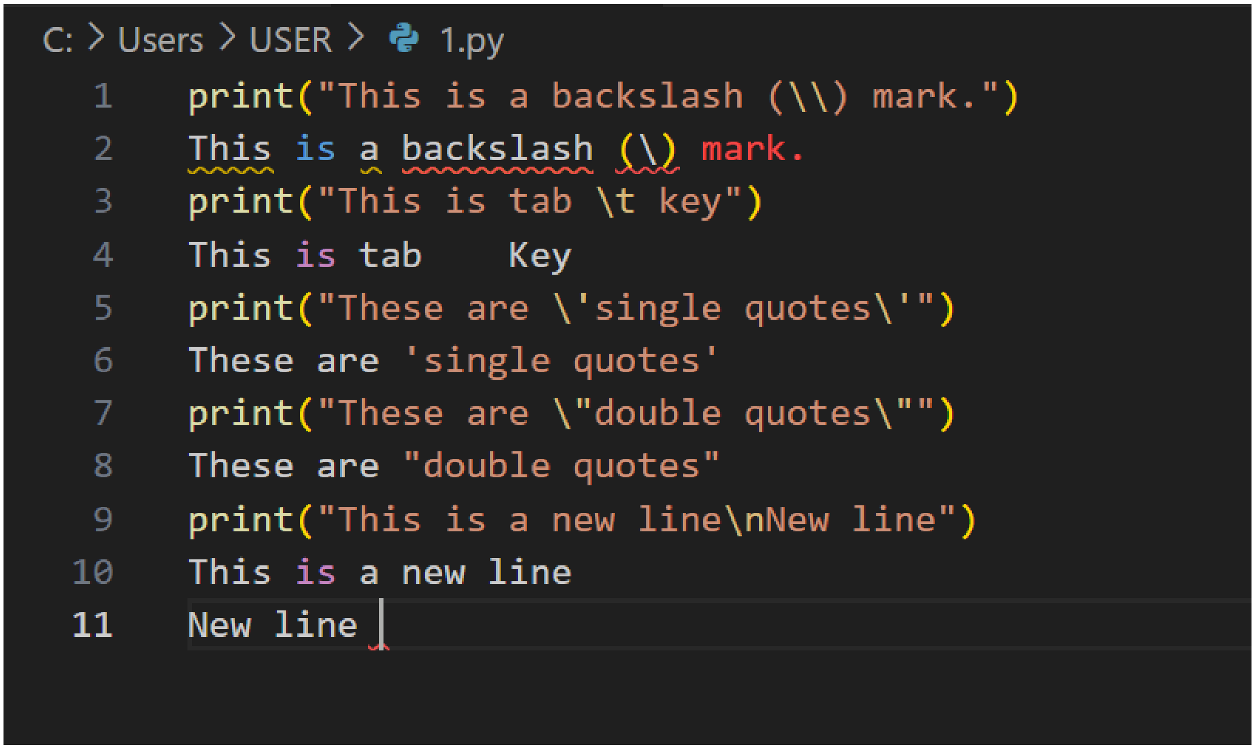
```
x = True
type(x)
<class 'bool'>
y= False
type(y)
<class 'bool'>

```

STRINGS:



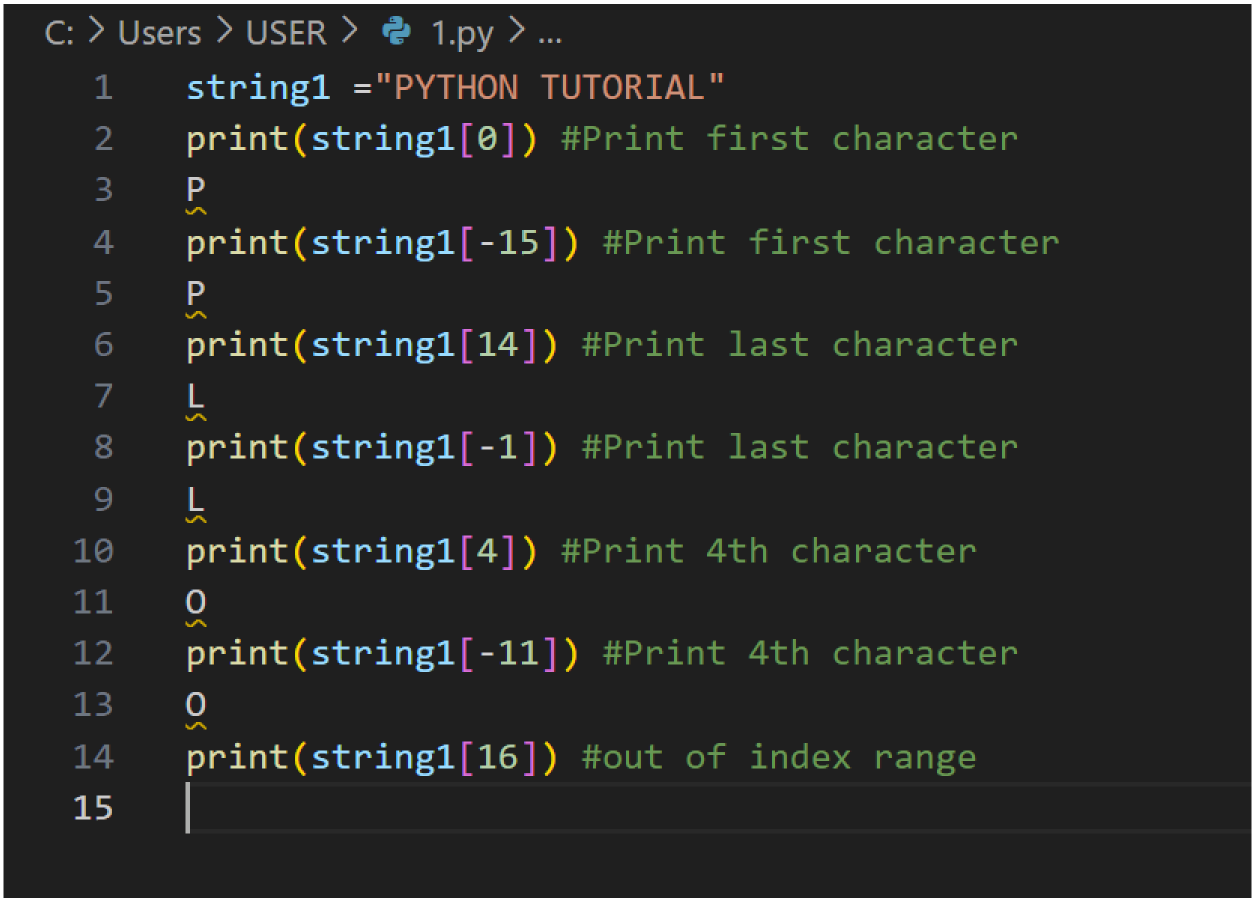
```
str1 = "String" #Strings start and end with double quotes
print(str1)
String
str2 = 'String' #Strings start and end with single quotes
print(str2)
 (variable) str3: Literal['String\'  #Strings start with double qoute and end …']
str3 ="String'  #Strings start with double qoute and end with single quote
SyntaxError:
str1 ='String" #Strings start with single quote and end with double quote
SyntaxError:
SyntaxError:
str2 = "Day's" #Single quote within double quotes
print(str2)
Day's
str2 ='Day"s'  #Double quote within single quotes
print (str2)
Day"s
```

SPECIAL CHARACTERS:



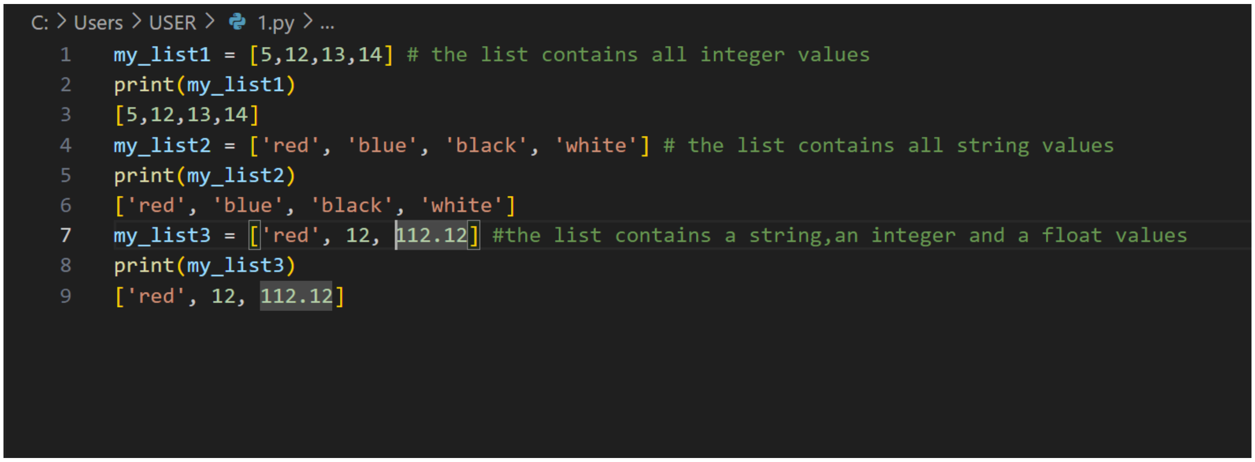
```
print("This is a backslash (\\) mark.")
This is a backslash (\) mark.
print("This is tab \t key")
This is tab     Key
print("These are \'single quotes\'")
These are 'single quotes'
print("These are \"double quotes\"")
These are "double quotes"
print("This is a new line\nNew line")
This is a new line
New line
```

STRING INDICES AND ACCESSING STRING ELEMENTS:



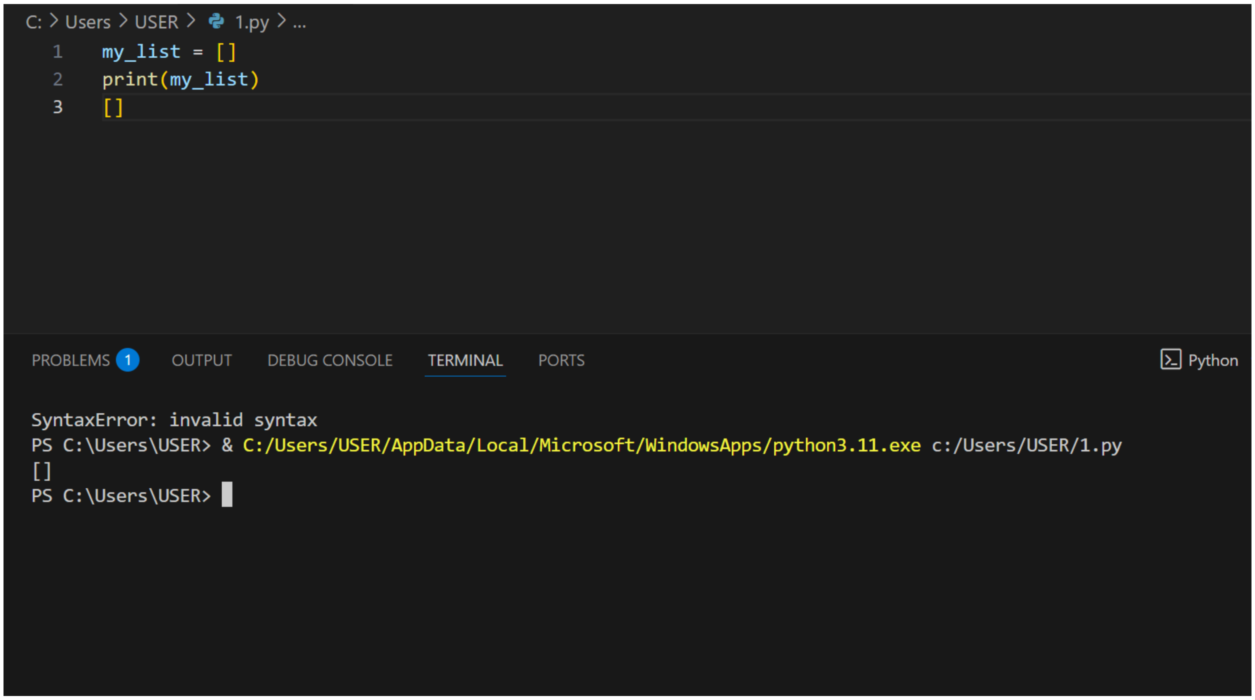
```
string1 ="PYTHON TUTORIAL"
print(string1[0]) #Print first character
P
print(string1[-15]) #Print first character
P
print(string1[14]) #Print last character
L
print(string1[-1]) #Print last character
L
print(string1[4]) #Print 4th character
O
print(string1[-11]) #Print 4th character
O
print(string1[16]) #out of index range

```

LISTS:



```
my_list1 = [5,12,13,14] # the list contains all integer values
print(my_list1)
[5,12,13,14]
my_list2 = ['red', 'blue', 'black', 'white'] # the list contains all string values
print(my_list2)
['red', 'blue', 'black', 'white']
my_list3 = ['red', 12, 112.12] #the list contains a string,an integer and a float values
print(my_list3)
['red', 12, 112.12]
```

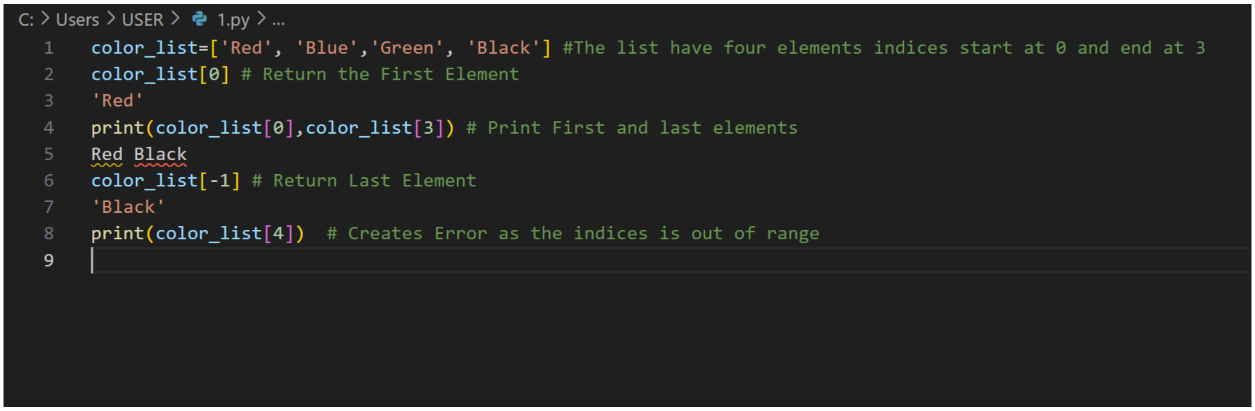
LISTS INDICES:



```
my_list = []
print(my_list)
[]
```

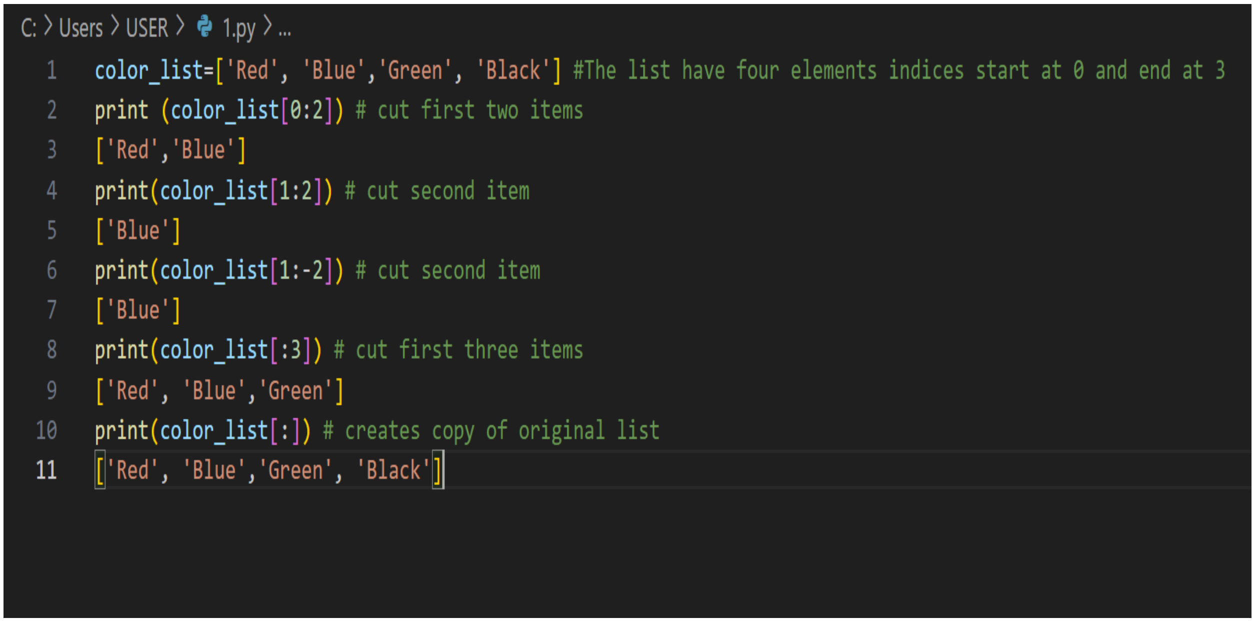
```
SyntaxError: invalid syntax
PS C:\Users\USER> & C:/Users/USER/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/USER/1.py
[]
PS C:\Users\USER> 
```

COLOR LISTS INDICES:



```
color_list=['Red', 'Blue','Green', 'Black'] #The list have four elements indices start at 0 and end at 3
color_list[0] # Return the First Element
'Red'
print(color_list[0],color_list[3]) # Print First and last elements
Red Black
color_list[-1] # Return Last Element
'Black'
print(color_list[4])  # Creates Error as the indices is out of range

```

COLOR LIST INDICES:



```
color_list=['Red', 'Blue','Green', 'Black'] #The list have four elements indices start at 0 and end at 3
print (color_list[0:2]) # cut first two items
['Red','Blue']
print(color_list[1:2]) # cut second item
['Blue']
print(color_list[1:-2]) # cut second item
['Blue']
print(color_list[:3]) # cut first three items
['Red', 'Blue','Green']
print(color_list[:]) # creates copy of original list
['Red', 'Blue','Green', 'Black']
```
