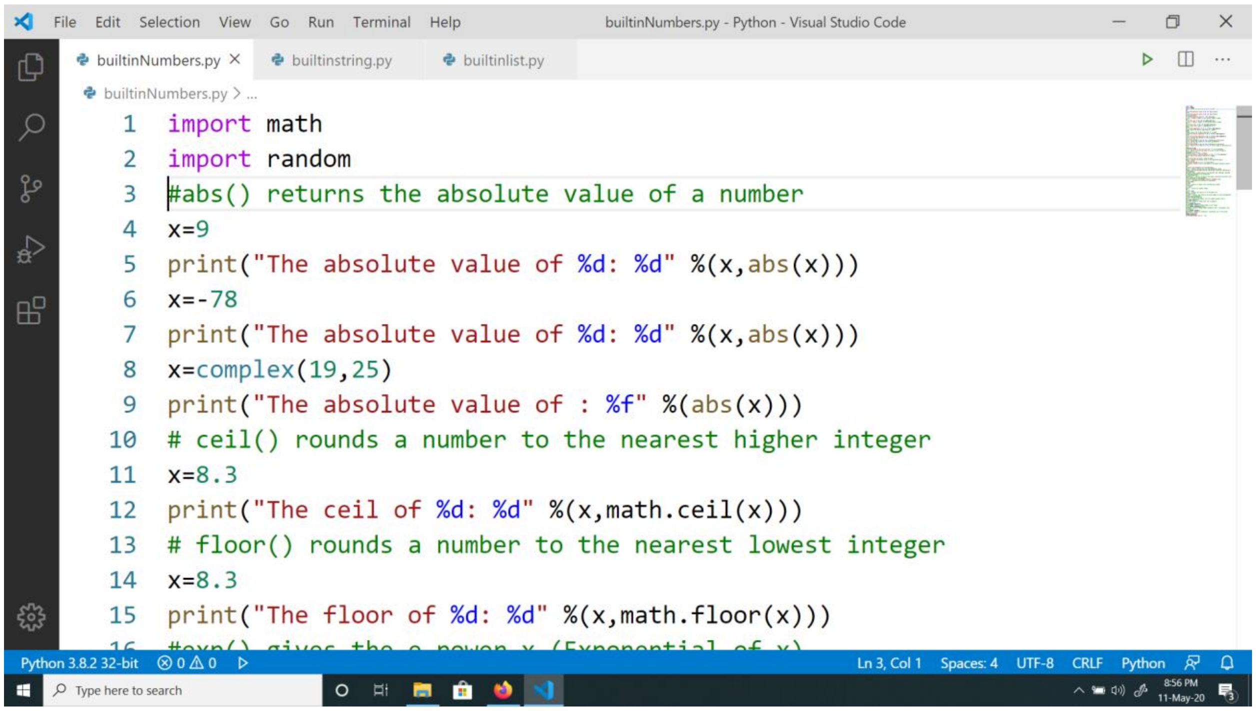
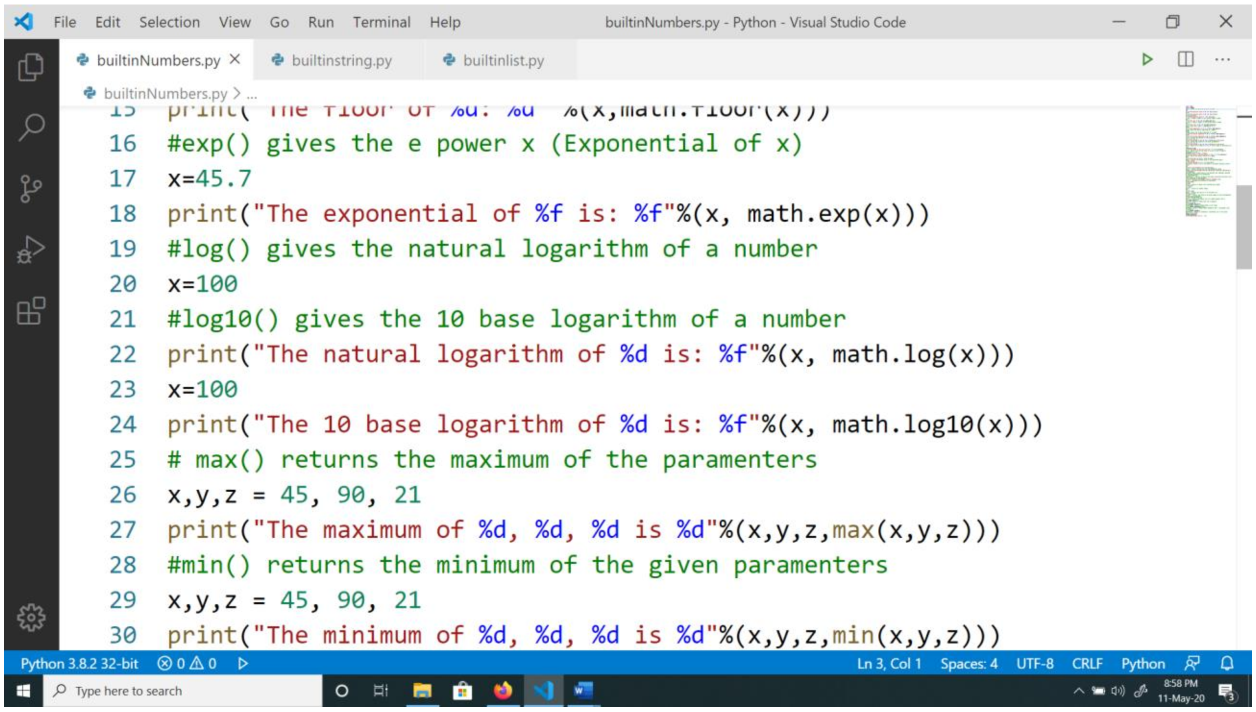


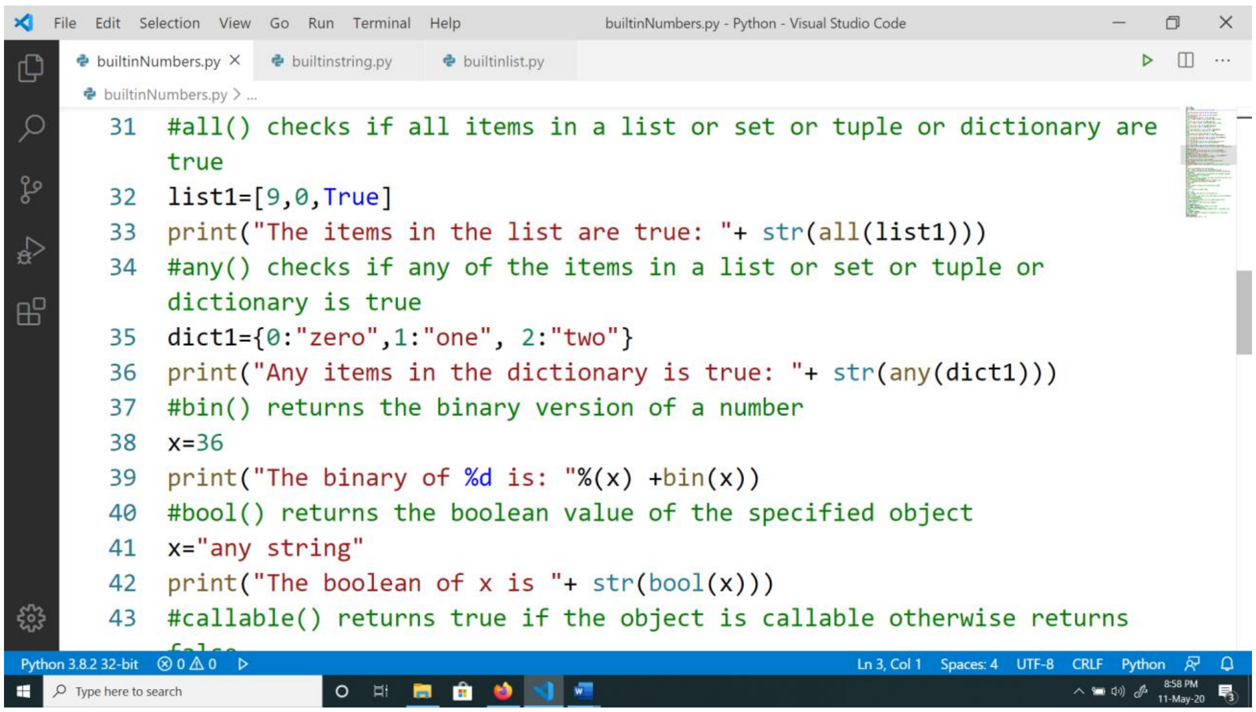
File Edit Selection View Go Run Terminal Help
builtinNumbers.py - Python - Visual Studio Code
builtinNumbers.py
builtinstring.py
builtinlist.py
Python 3.8.2 32-bit
Ln 3, Col 1 Spaces: 4 UTF-8 CRLF Python
Type here to search
8:56 PM 11-May-20

```python
import math
import random
#abs() returns the absolute value of a number
x=9
print("The absolute value of %d: %d" %(x,abs(x)))
x=-78
print("The absolute value of %d: %d" %(x,abs(x)))
x=complex(19,25)
print("The absolute value of : %f" %(abs(x)))
# ceil() rounds a number to the nearest higher integer
x=8.3
print("The ceil of %d: %d" %(x,math.ceil(x)))
# floor() rounds a number to the nearest lowest integer
x=8.3
print("The floor of %d: %d" %(x,math.floor(x)))
```

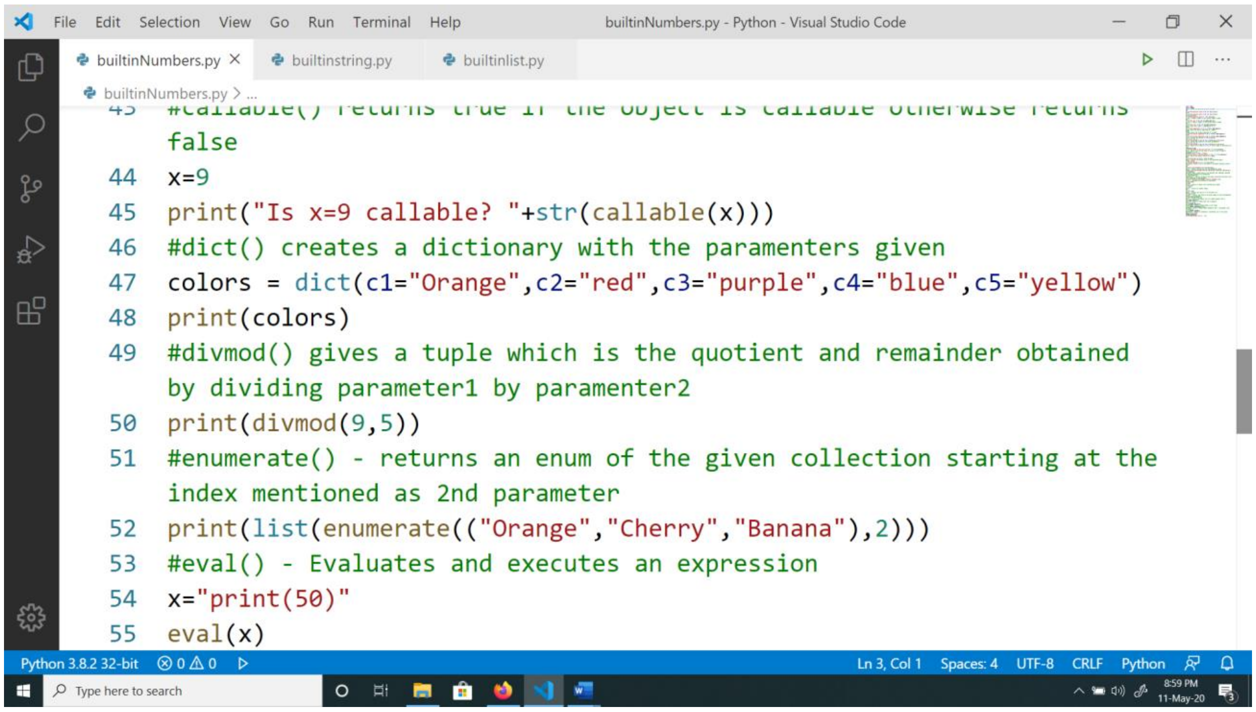


File Edit Selection View Go Run Terminal Help
builtinNumbers.py - Python - Visual Studio Code
builtinNumbers.py
builtinstring.py
builtinlist.py
Python 3.8.2 32-bit
Ln 3, Col 1 Spaces: 4 UTF-8 CRLF Python
Type here to search
8:58 PM 11-May-20

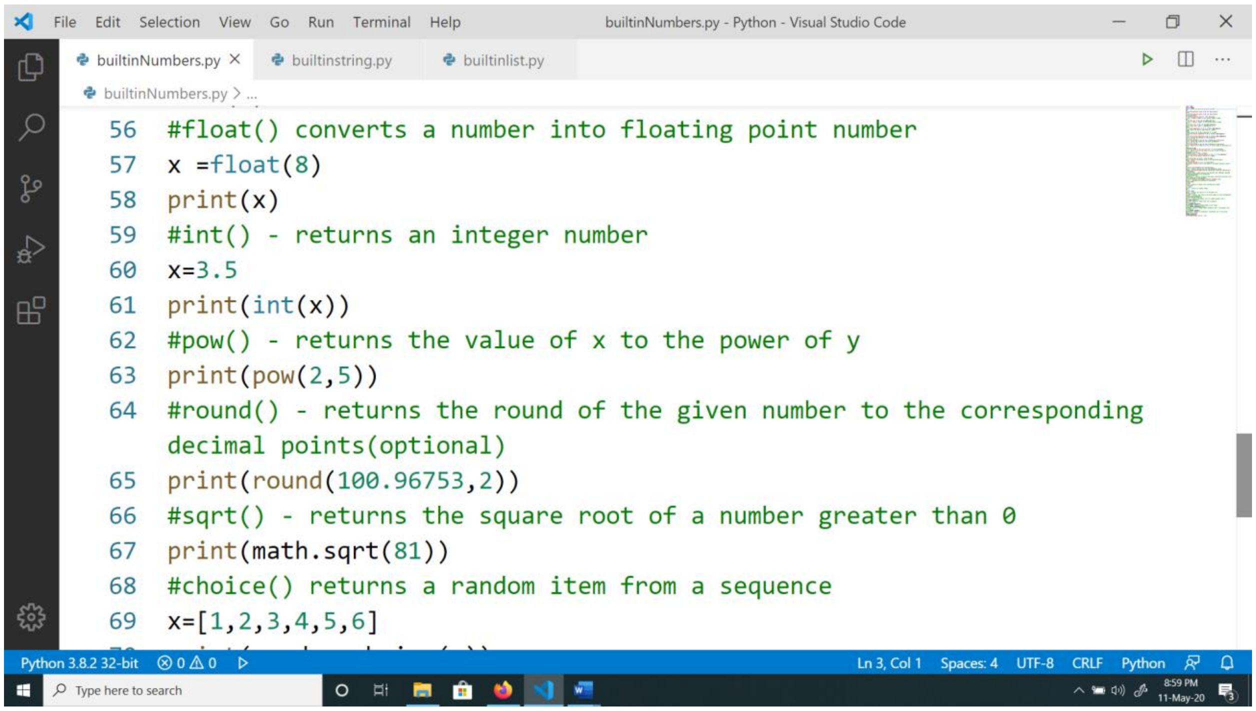
```python
#exp() gives the e power x (Exponential of x)
x=45.7
print("The exponential of %f is: %f"%(x, math.exp(x)))
#log() gives the natural logarithm of a number
x=100
#log10() gives the 10 base logarithm of a number
print("The natural logarithm of %d is: %f"%(x, math.log(x)))
x=100
print("The 10 base logarithm of %d is: %f"%(x, math.log10(x)))
# max() returns the maximum of the paramenters
x,y,z = 45, 90, 21
print("The maximum of %d, %d, %d is %d"%(x,y,z,max(x,y,z)))
#min() returns the minimum of the given paramenters
x,y,z = 45, 90, 21
print("The minimum of %d, %d, %d is %d"%(x,y,z,min(x,y,z)))
```



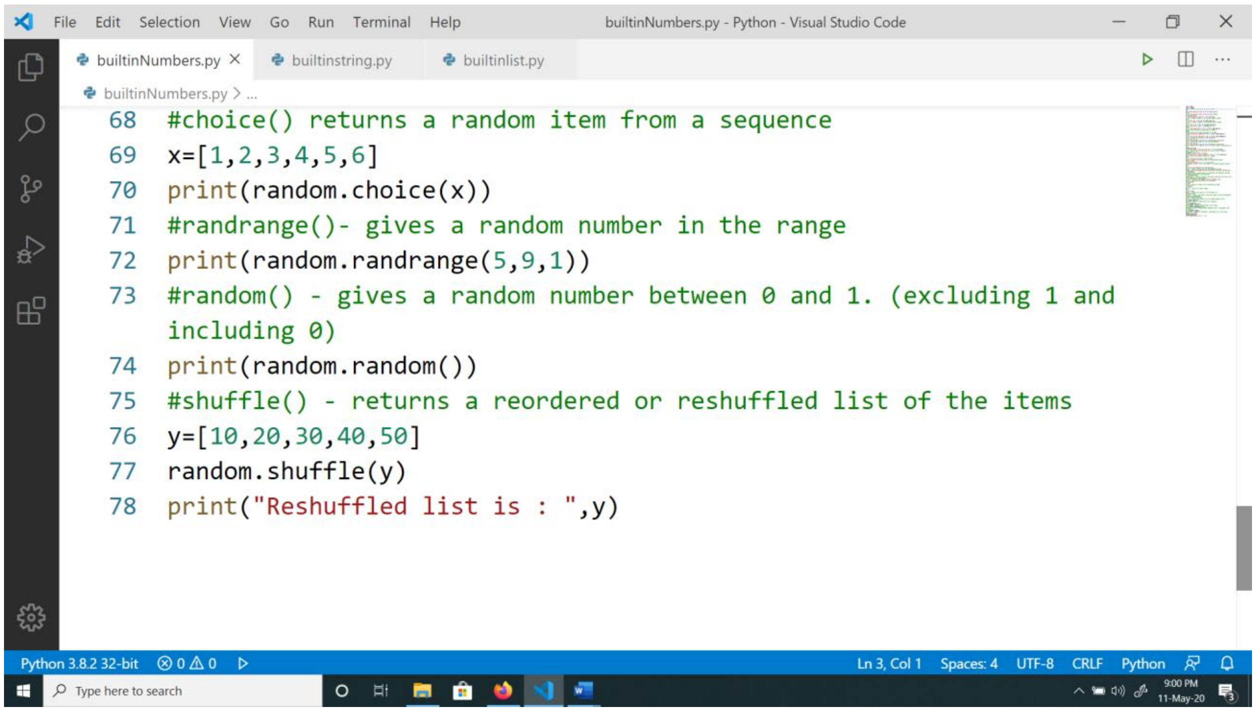
```python
#all() checks if all items in a list or set or tuple or dictionary are true
list1=[9,0,True]
print("The items in the list are true: "+ str(all(list1)))
#any() checks if any of the items in a list or set or tuple or dictionary is true
dict1={0:"zero",1:"one", 2:"two"}
print("Any items in the dictionary is true: "+ str(any(dict1)))
#bin() returns the binary version of a number
x=36
print("The binary of %d is: "%(x) +bin(x))
#bool() returns the boolean value of the specified object
x="any string"
print("The boolean of x is "+ str(bool(x)))
#callable() returns true if the object is callable otherwise returns
```



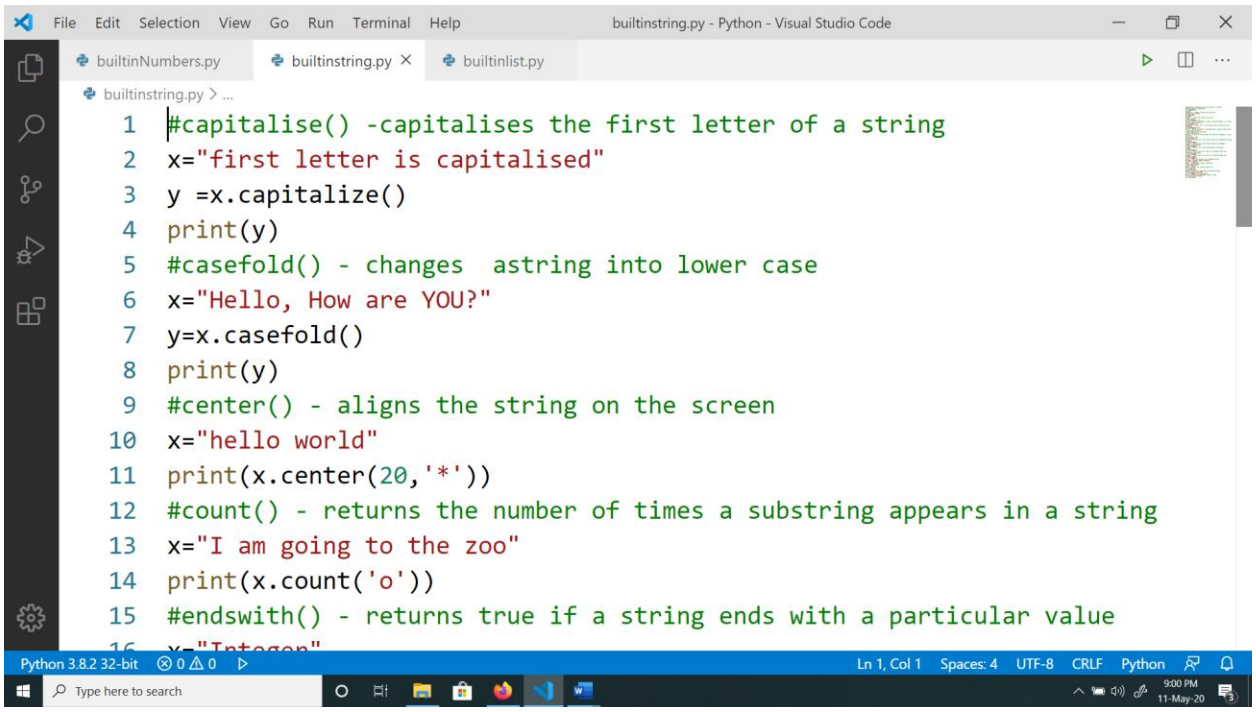
```python
#callable() returns true if the object is callable otherwise returns false
x=9
print("Is x=9 callable? "+str(callable(x)))
#dict() creates a dictionary with the paramenters given
colors = dict(c1="Orange",c2="red",c3="purple",c4="blue",c5="yellow")
print(colors)
#divmod() gives a tuple which is the quotient and remainder obtained by dividing parameter1 by paramenter2
print(divmod(9,5))
#enumerate() - returns an enum of the given collection starting at the index mentioned as 2nd parameter
print(list(enumerate(("Orange","Cherry","Banana"),2)))
#eval() - Evaluates and executes an expression
x="print(50)"
eval(x)
```


File Edit Selection View Go Run Terminal Help
builtinNumbers.py - Python - Visual Studio Code
builtinNumbers.py
builtinstring.py
builtinlist.py
builtinNumbers.py > ...
Python 3.8.2 32-bit
Ln 3, Col 1 Spaces: 4 UTF-8 CRLF Python
Type here to search
8:59 PM 11-May-20

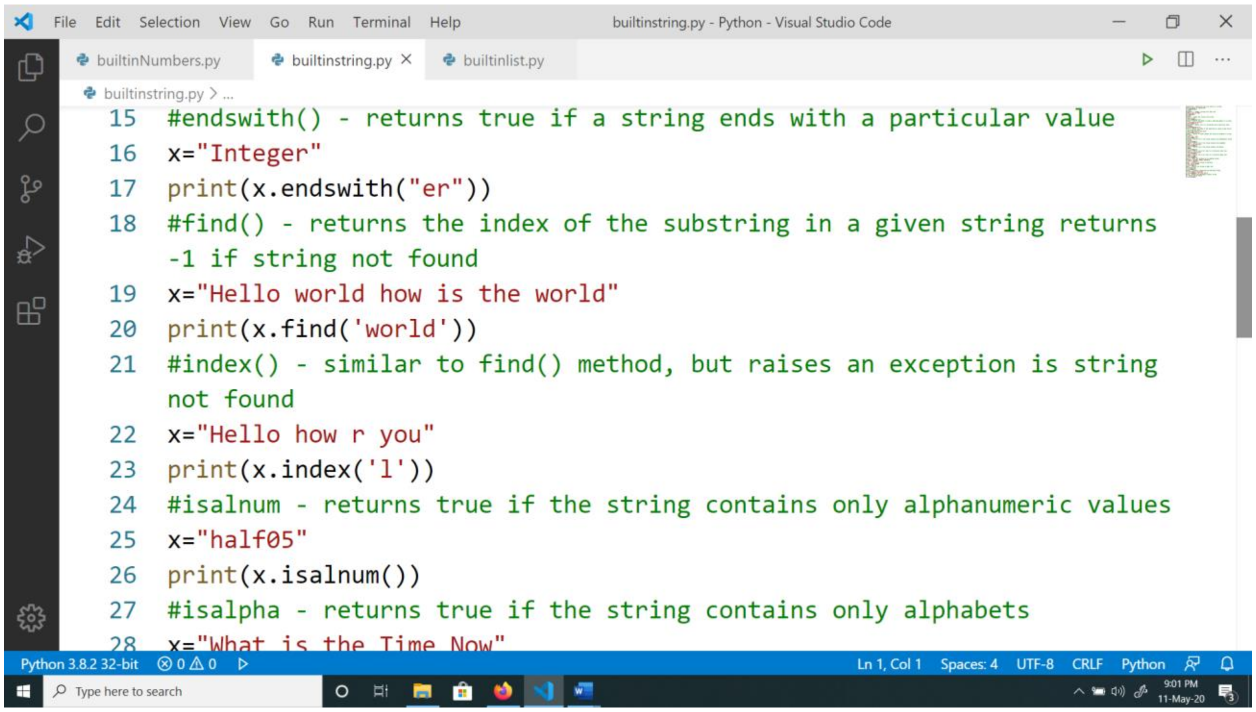
```python
#float() converts a number into floating point number
x =float(8)
print(x)
#int() - returns an integer number
x=3.5
print(int(x))
#pow() - returns the value of x to the power of y
print(pow(2,5))
#round() - returns the round of the given number to the corresponding decimal points(optional)
print(round(100.96753,2))
#sqrt() - returns the square root of a number greater than 0
print(math.sqrt(81))
#choice() returns a random item from a sequence
x=[1,2,3,4,5,6]
```


File Edit Selection View Go Run Terminal Help
builtinNumbers.py - Python - Visual Studio Code
builtinNumbers.py
builtinstring.py
builtinlist.py
builtinNumbers.py > ...
Python 3.8.2 32-bit
Ln 3, Col 1 Spaces: 4 UTF-8 CRLF Python
Type here to search
9:00 PM 11-May-20

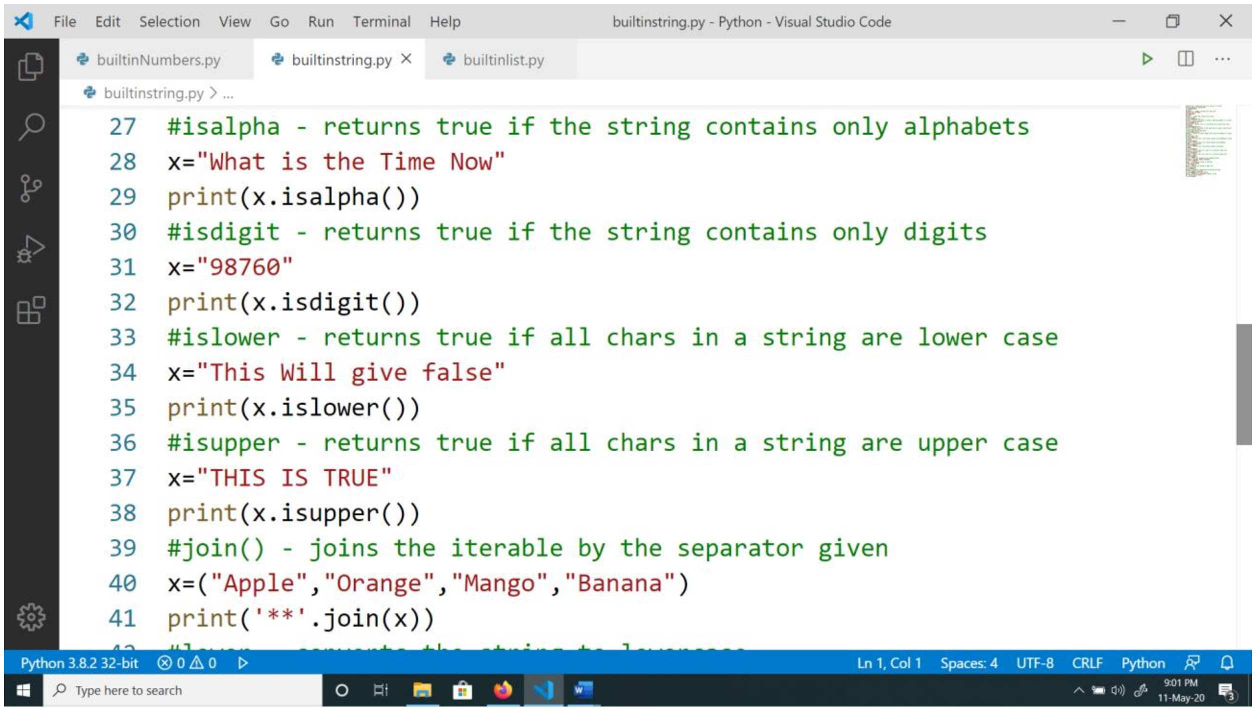
```python
#choice() returns a random item from a sequence
x=[1,2,3,4,5,6]
print(random.choice(x))
#randrange()- gives a random number in the range
print(random.randrange(5,9,1))
#random() - gives a random number between 0 and 1. (excluding 1 and including 0)
print(random.random())
#shuffle() - returns a reordered or reshuffled list of the items
y=[10,20,30,40,50]
random.shuffle(y)
print("Reshuffled list is : ",y)
```



```python
#capitalise() -capitalises the first letter of a string
x="first letter is capitalised"
y =x.capitalize()
print(y)
#casefold() - changes  astring into lower case
x="Hello, How are YOU?"
y=x.casefold()
print(y)
#center() - aligns the string on the screen
x="hello world"
print(x.center(20,'*'))
#count() - returns the number of times a substring appears in a string
x="I am going to the zoo"
print(x.count('o'))
#endswith() - returns true if a string ends with a particular value
x="Integer"
```

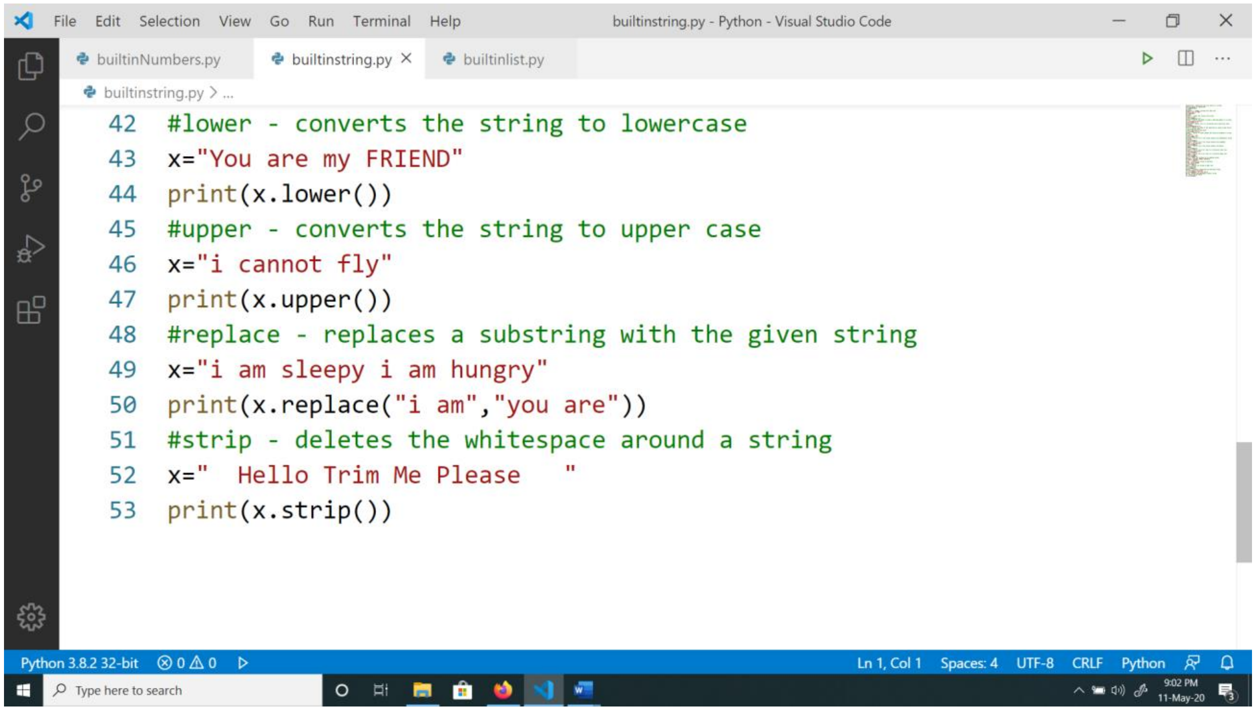


```python
#endswith() - returns true if a string ends with a particular value
x="Integer"
print(x.endswith("er"))
#find() - returns the index of the substring in a given string returns -1 if string not found
x="Hello world how is the world"
print(x.find('world'))
#index() - similar to find() method, but raises an exception is string not found
x="Hello how r you"
print(x.index('l'))
#isalnum - returns true if the string contains only alphanumeric values
x="half05"
print(x.isalnum())
#isalpha - returns true if the string contains only alphabets
x="What is the Time Now"
```



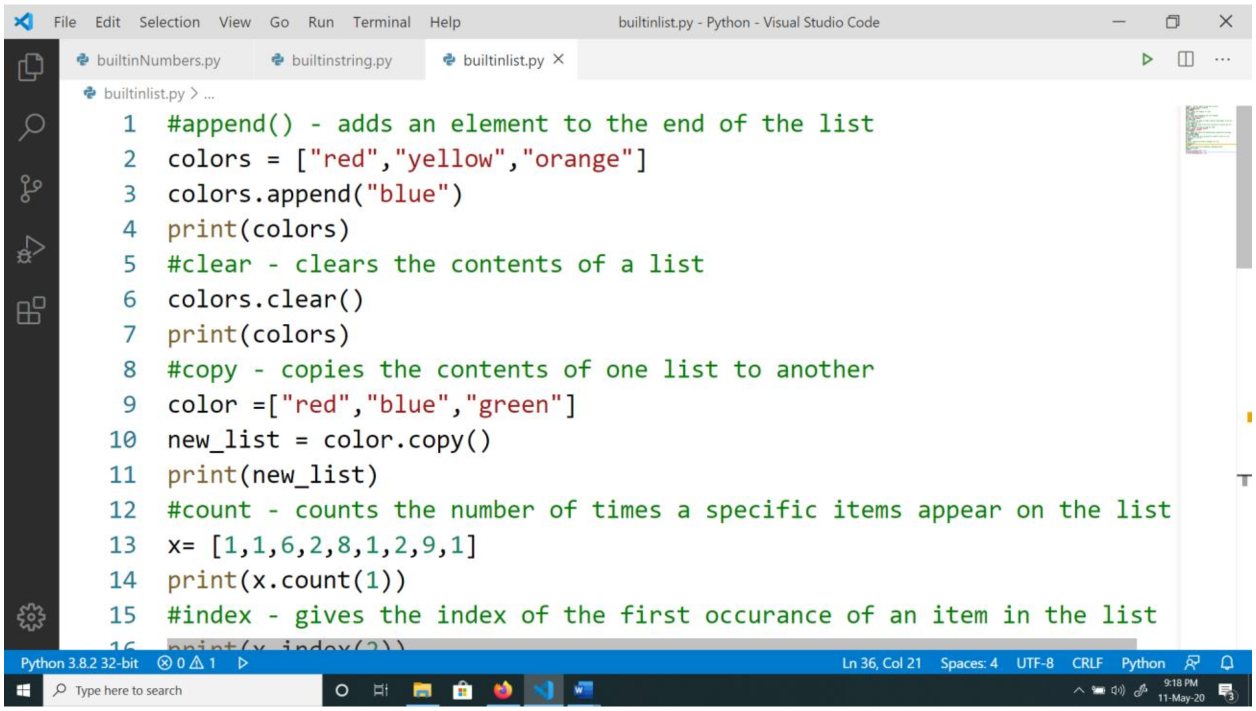
File Edit Selection View Go Run Terminal Help
builtinstring.py - Python - Visual Studio Code
builtinNumbers.py
builtinstring.py
builtinlist.py
Python 3.8.2 32-bit
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python
Type here to search
9:01 PM 11-May-20

```python
#isalpha - returns true if the string contains only alphabets
x="What is the Time Now"
print(x.isalpha())
#isdigit - returns true if the string contains only digits
x="98760"
print(x.isdigit())
#islower - returns true if all chars in a string are lower case
x="This Will give false"
print(x.islower())
#isupper - returns true if all chars in a string are upper case
x="THIS IS TRUE"
print(x.isupper())
#join() - joins the iterable by the separator given
x=("Apple","Orange","Mango","Banana")
print('**'.join(x))
```

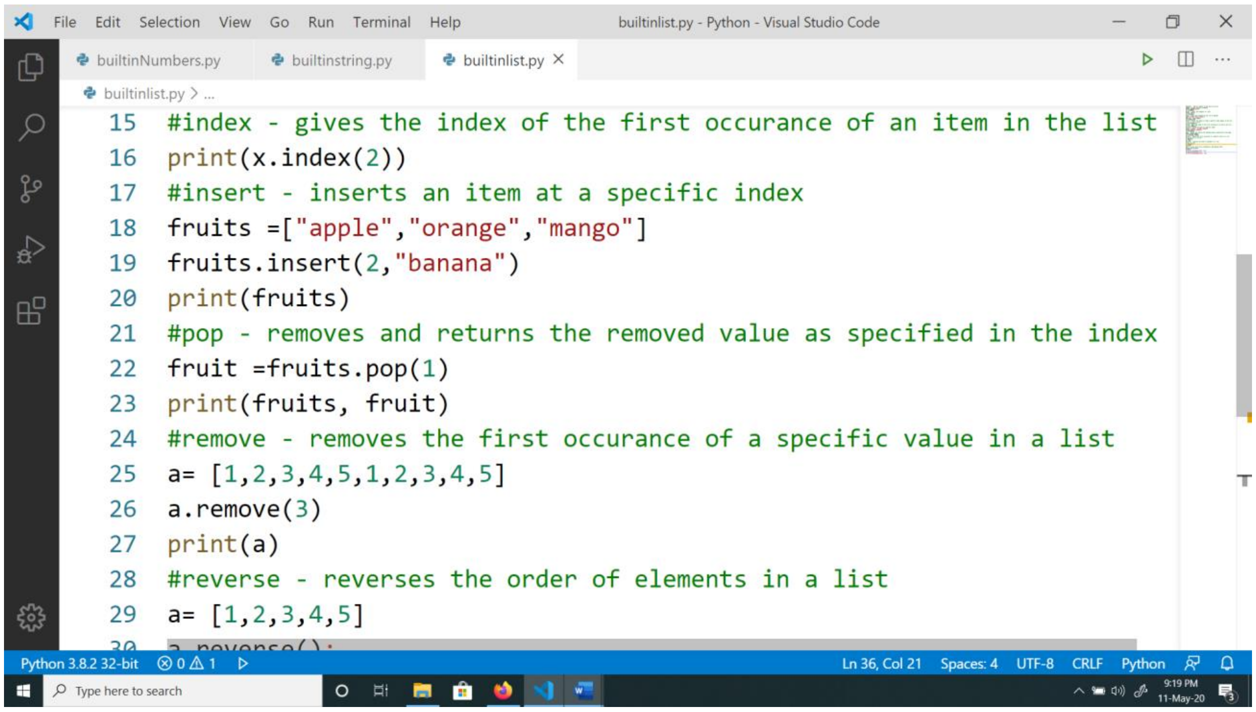


File Edit Selection View Go Run Terminal Help
builtinstring.py - Python - Visual Studio Code
builtinNumbers.py
builtinstring.py
builtinlist.py
Python 3.8.2 32-bit
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python
Type here to search
9:02 PM 11-May-20

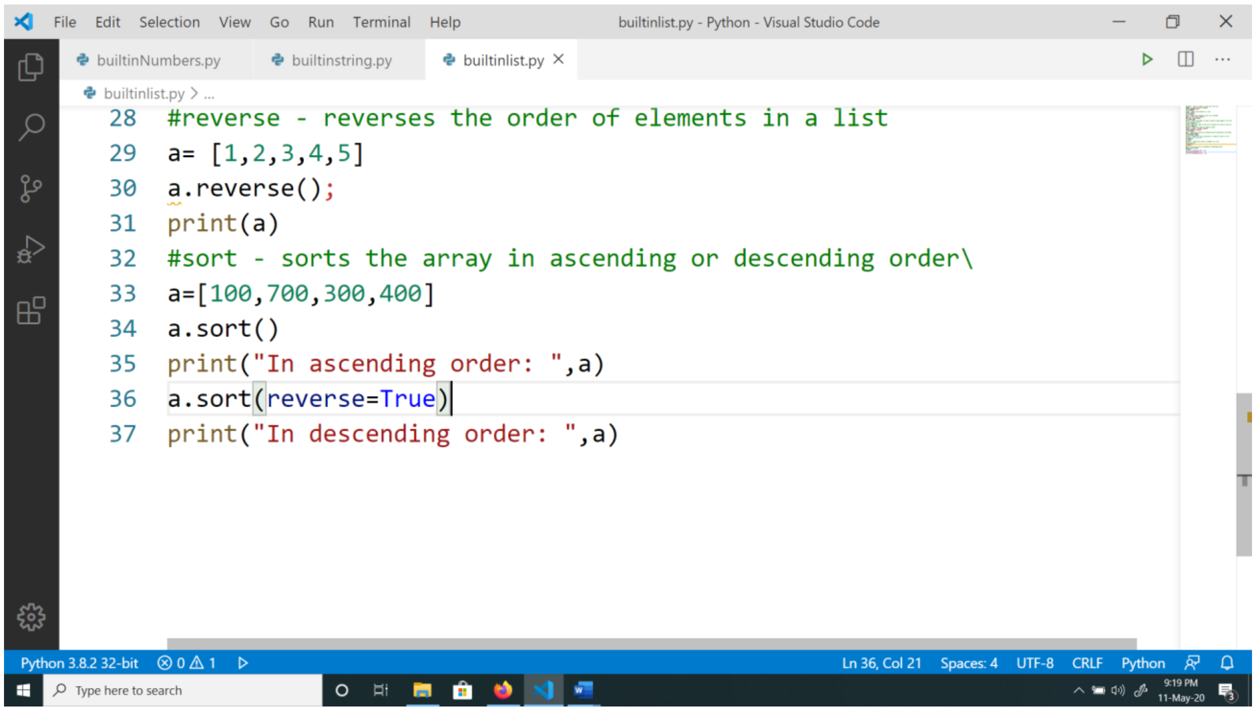
```python
#lower - converts the string to lowercase
x="You are my FRIEND"
print(x.lower())
#upper - converts the string to upper case
x="i cannot fly"
print(x.upper())
#replace - replaces a substring with the given string
x="i am sleepy i am hungry"
print(x.replace("i am","you are"))
#strip - deletes the whitespace around a string
x="  Hello Trim Me Please    "
print(x.strip())
```



```
#append() - adds an element to the end of the list
colors = ["red","yellow","orange"]
colors.append("blue")
print(colors)
#clear - clears the contents of a list
colors.clear()
print(colors)
#copy - copies the contents of one list to another
color =["red","blue","green"]
new_list = color.copy()
print(new_list)
#count - counts the number of times a specific items appear on the list
x= [1,1,6,2,8,1,2,9,1]
print(x.count(1))
#index - gives the index of the first occurance of an item in the list
```

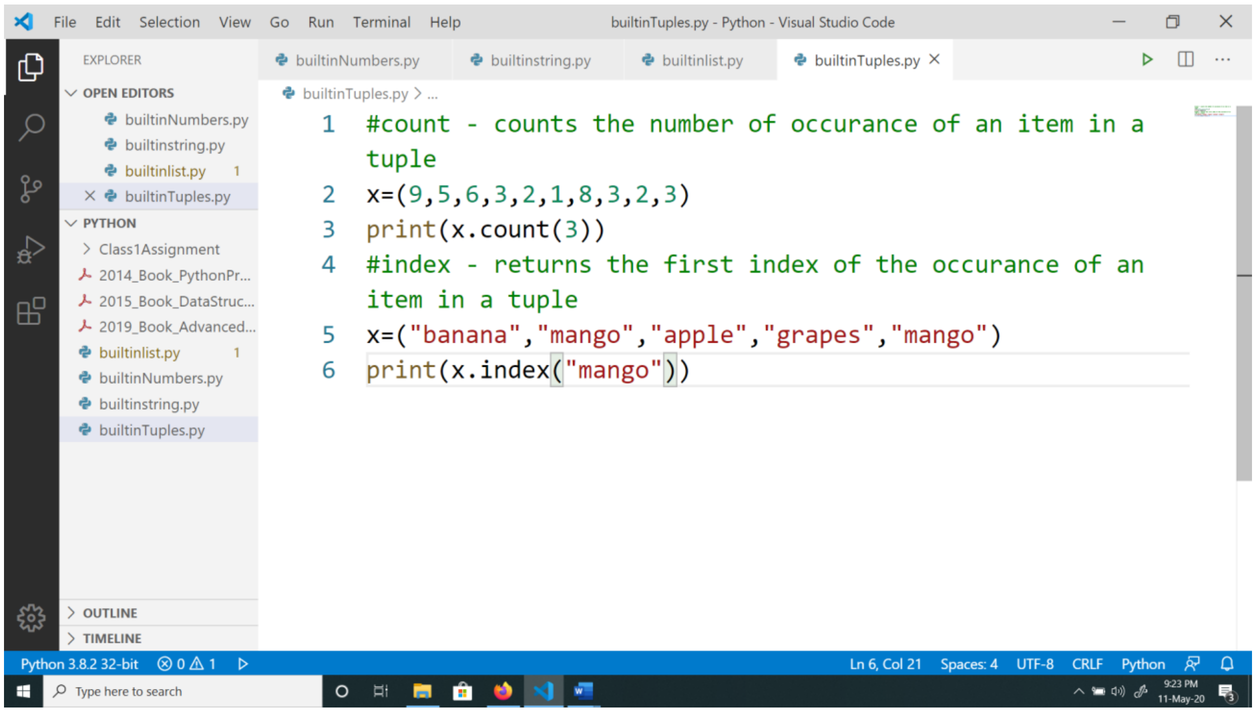


```
#index - gives the index of the first occurance of an item in the list
print(x.index(2))
#insert - inserts an item at a specific index
fruits =["apple","orange","mango"]
fruits.insert(2,"banana")
print(fruits)
#pop - removes and returns the removed value as specified in the index
fruit =fruits.pop(1)
print(fruits, fruit)
#remove - removes the first occurance of a specific value in a list
a= [1,2,3,4,5,1,2,3,4,5]
a.remove(3)
print(a)
#reverse - reverses the order of elements in a list
a= [1,2,3,4,5]
```

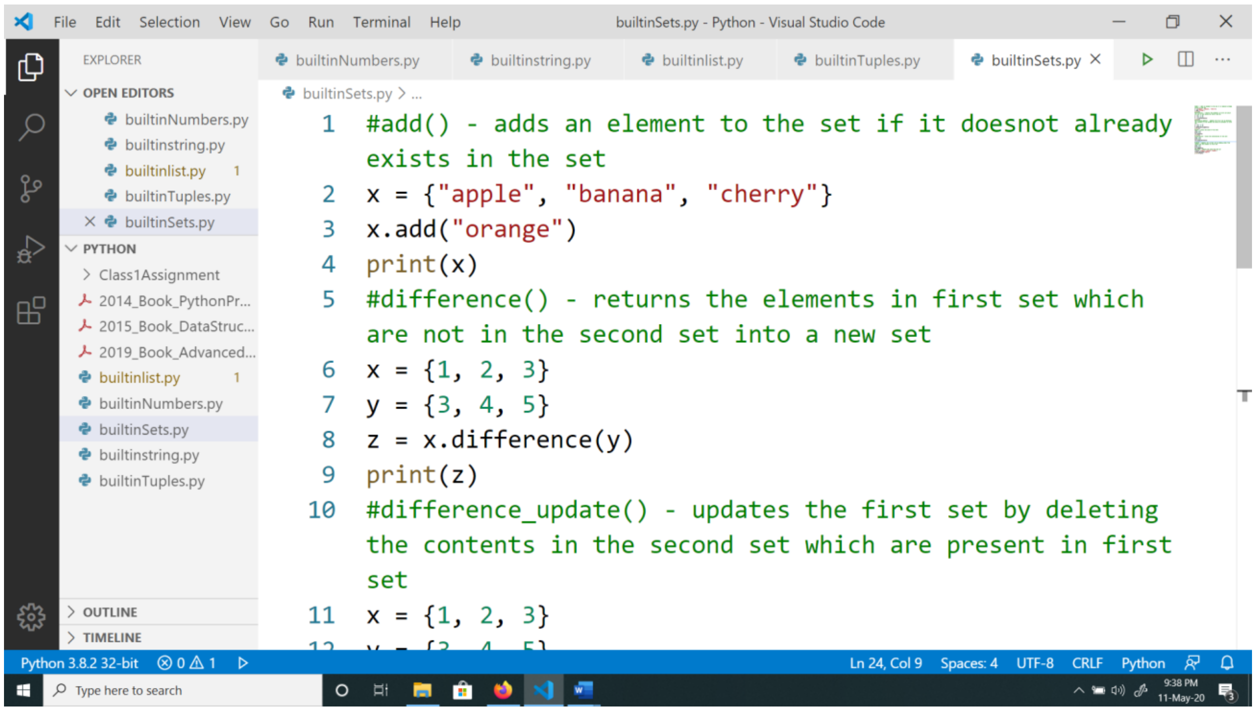


```
#reverse - reverses the order of elements in a list
a= [1,2,3,4,5]
a.reverse();
print(a)
#sort - sorts the array in ascending or descending order\
a=[100,700,300,400]
a.sort()
print("In ascending order: ",a)
a.sort(reverse=True)
print("In descending order: ",a)
```

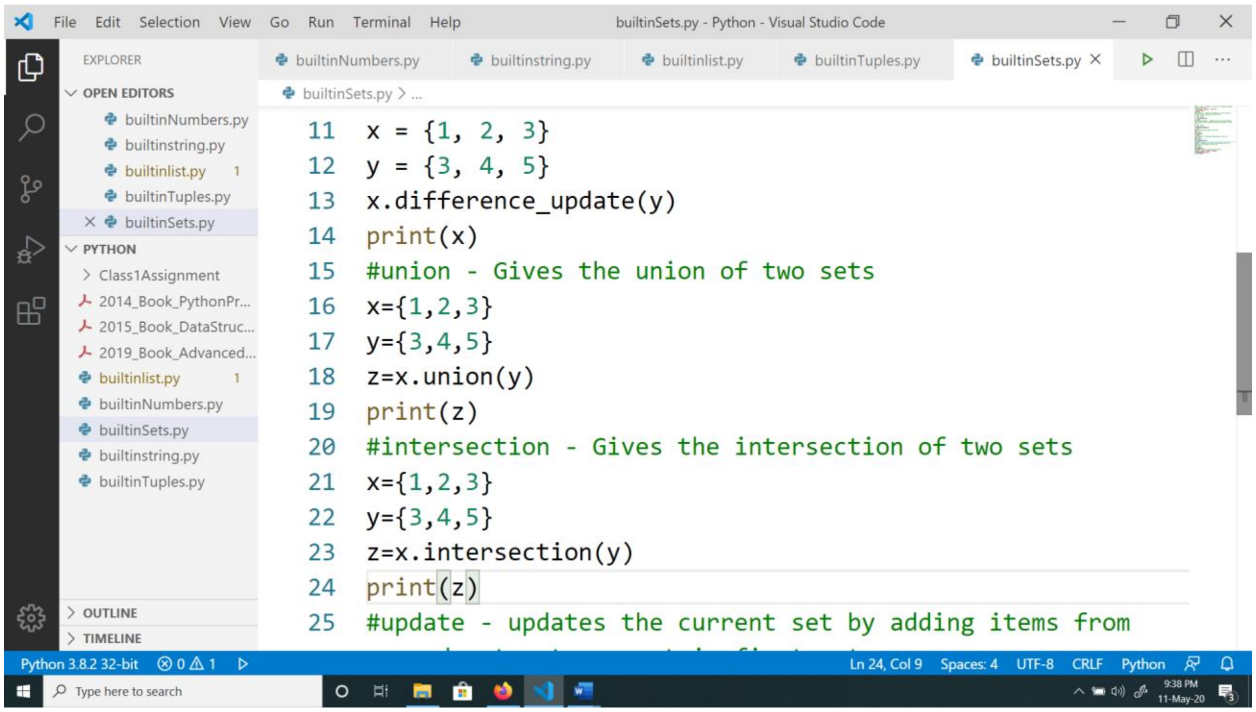
File Edit Selection View Go Run Terminal Help
builtinlist.py - Python - Visual Studio Code
builtinNumbers.py
builtinstring.py
builtinlist.py
Python 3.8.2 32-bit
Ln 36, Col 21 Spaces: 4 UTF-8 CRLF Python
Type here to search
9:19 PM 11-May-20



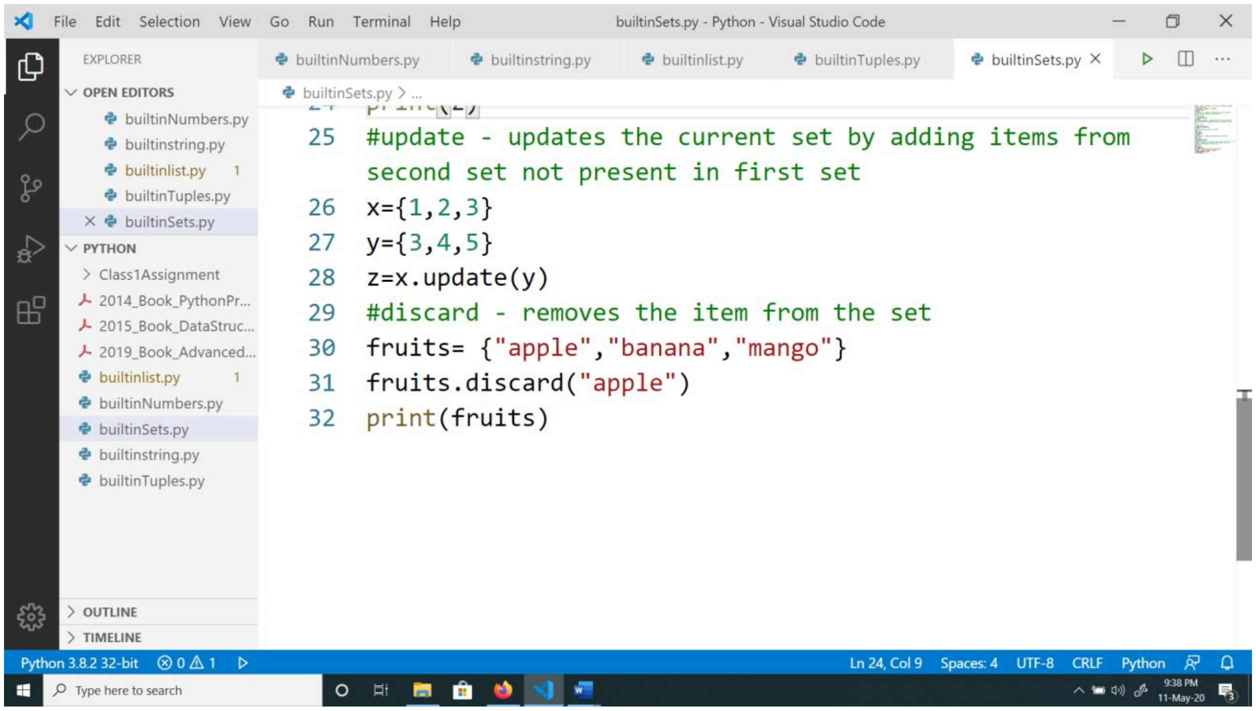
```
#count - counts the number of occurance of an item in a tuple
x=(9,5,6,3,2,1,8,3,2,3)
print(x.count(3))
#index - returns the first index of the occurance of an item in a tuple
x=("banana","mango","apple","grapes","mango")
print(x.index("mango"))
```



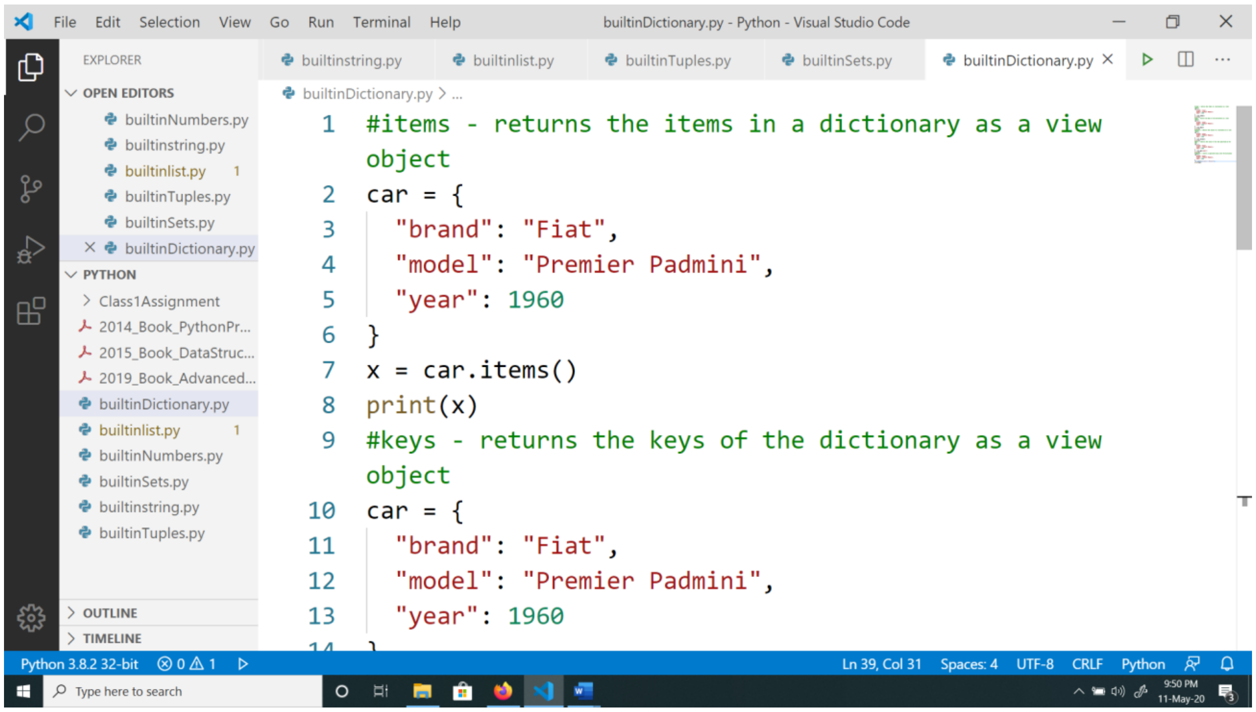
```python
#add() - adds an element to the set if it doesnot already exists in the set
x = {"apple", "banana", "cherry"}
x.add("orange")
print(x)
#difference() - returns the elements in first set which are not in the second set into a new set
x = {1, 2, 3}
y = {3, 4, 5}
z = x.difference(y)
print(z)
#difference_update() - updates the first set by deleting the contents in the second set which are present in first set
x = {1, 2, 3}
```


File Edit Selection View Go Run Terminal Help
builtinSets.py - Python - Visual Studio Code
EXPLORER
OPEN EDITORS
PYTHON
OUTLINE
TIMELINE
Python 3.8.2 32-bit
Ln 24, Col 9 Spaces: 4 UTF-8 CRLF Python
Type here to search

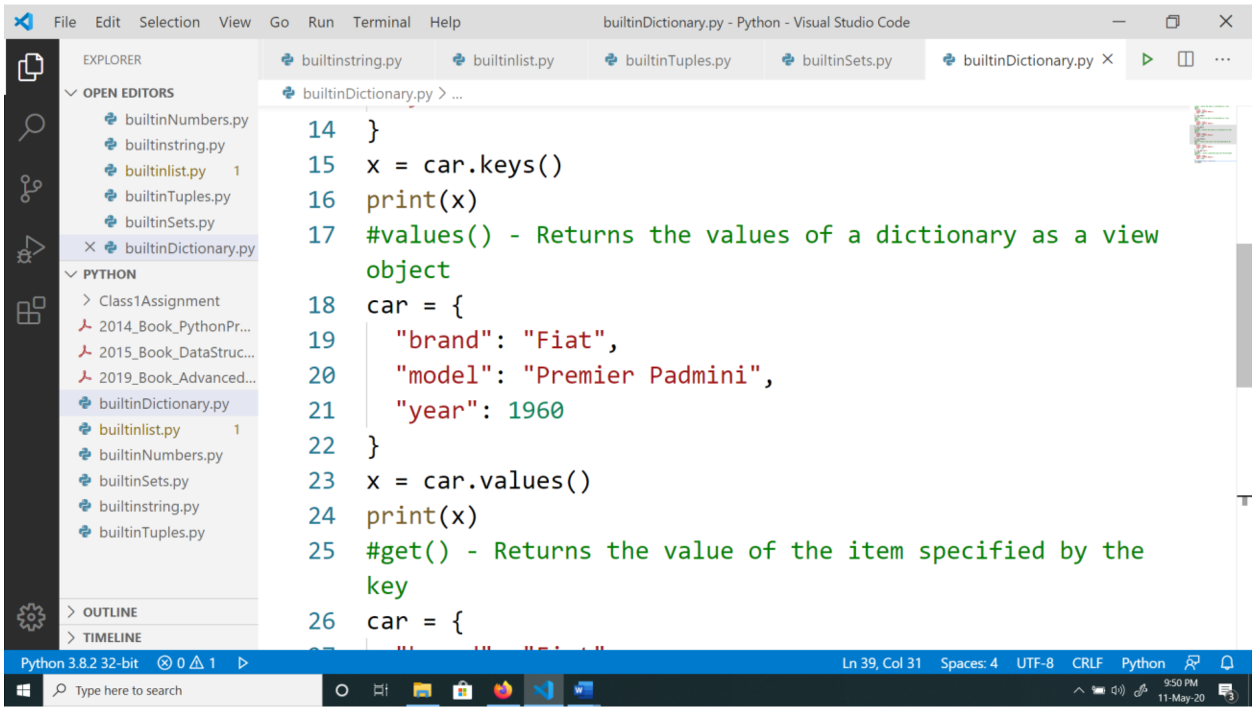
```python
x = {1, 2, 3}
y = {3, 4, 5}
x.difference_update(y)
print(x)
#union - Gives the union of two sets
x={1,2,3}
y={3,4,5}
z=x.union(y)
print(z)
#intersection - Gives the intersection of two sets
x={1,2,3}
y={3,4,5}
z=x.intersection(y)
print(z)
#update - updates the current set by adding items from
```



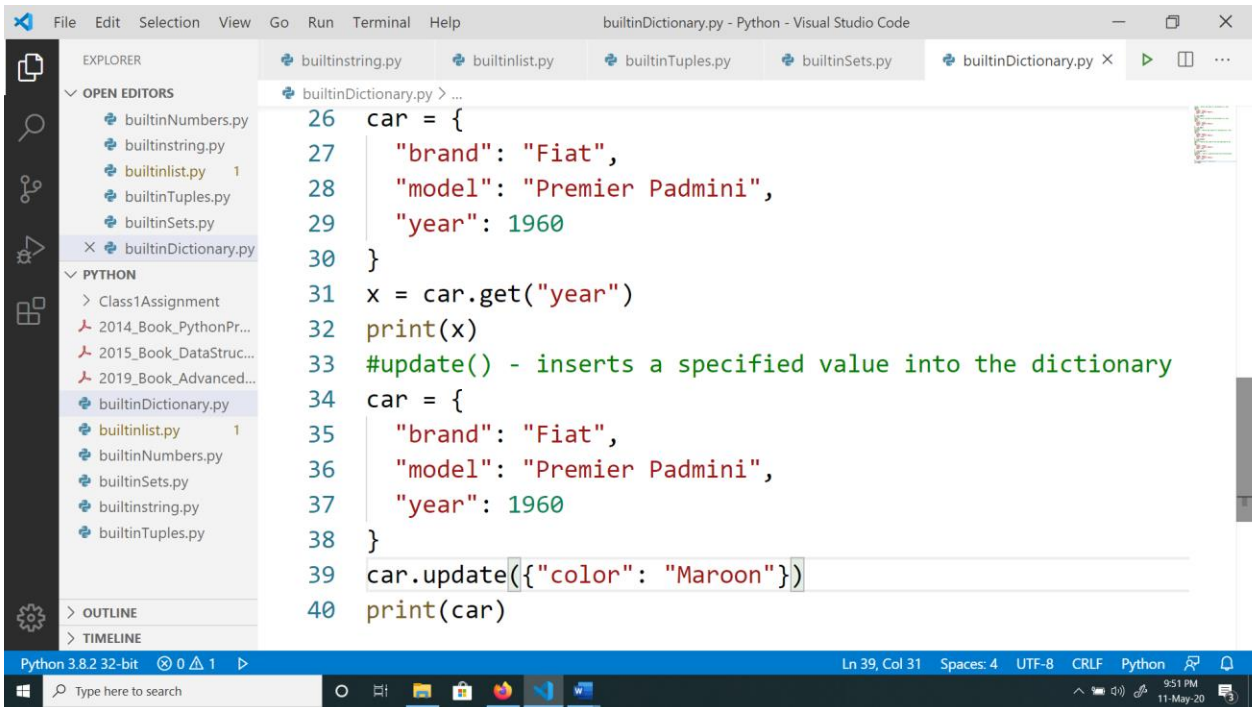
```
#update - updates the current set by adding items from second set not present in first set
x={1,2,3}
y={3,4,5}
z=x.update(y)
#discard - removes the item from the set
fruits= {"apple","banana","mango"}
fruits.discard("apple")
print(fruits)
```



```
#items - returns the items in a dictionary as a view object
car = {
  "brand": "Fiat",
  "model": "Premier Padmini",
  "year": 1960
}
x = car.items()
print(x)
#keys - returns the keys of the dictionary as a view object
car = {
  "brand": "Fiat",
  "model": "Premier Padmini",
  "year": 1960
```



```python
}
x = car.keys()
print(x)
#values() - Returns the values of a dictionary as a view object
car = {
  "brand": "Fiat",
  "model": "Premier Padmini",
  "year": 1960
}
x = car.values()
print(x)
#get() - Returns the value of the item specified by the key
car = {
```



```python
car = {
  "brand": "Fiat",
  "model": "Premier Padmini",
  "year": 1960
}
x = car.get("year")
print(x)
#update() - inserts a specified value into the dictionary
car = {
  "brand": "Fiat",
  "model": "Premier Padmini",
  "year": 1960
}
car.update({"color": "Maroon"})
print(car)
```