28-10-2025

Using Python as a Calculator

```
In [1]:  2 + 2

Out[1]:  4
```

```
In [2]:  50 - 5*6

Out[2]:  20
```

```
In [3]:  (50 - 5*6) / 4

Out[3]:  5.0
```

```
In [4]:  8/5

Out[4]:  1.6
```

```
In [5]:  17 / 3

Out[5]:  5.666666666666667
```

```
In [6]:  17 // 3

Out[6]:  5
```

```
In [7]:  17 % 3

Out[7]:  2
```

```
In [8]:  5 * 3 + 2

Out[8]:  17
```

```
In [9]:  5 ** 2

Out[9]:  25
```

```
In [10]:  2 ** 7

Out[10]:  128
```

```
In [11]:  width = 20
          height = 5 * 9
          width * height

Out[11]:  900
```

```
In [12]:  n

          ---------------------------------------------------------------------------
          NameError                                 Traceback (most recent call last)
          Cell In[12], line 1
          ----> 1 n

          NameError: name 'n' is not defined

In [13]:  4 * 3.75 - 1

Out[13]:  14.0

In [14]:  tax = 12.5 / 100
          price = 100.50
          price * tax

Out[14]:  12.5625

In [15]:  price + _

Out[15]:  113.0625

In [16]:  round(_, 2)

Out[16]:  113.06

In [17]:  'spam eggs'

Out[17]:  'spam eggs'

In [18]:  "Paris rabbit got your back :)! Yay!"

Out[18]:  'Paris rabbit got your back :)! Yay!'

In [19]:  '1975'

Out[19]:  '1975'

In [20]:  'Py' 'thon'

Out[20]:  'Python'

In [24]:  word = 'python'

In [25]:  word[0]

Out[25]:  'p'

In [26]:  word[5]

Out[26]:  'n'
```

```
In [27]: word[-1]

Out[27]: 'n'

In [28]: word[-2]

Out[28]: 'o'

In [29]: word[-6]

Out[29]: 'p'

In [30]: word[0:2]

Out[30]: 'py'

In [31]: word[2:5]

Out[31]: 'tho'

In [32]: word[:2]

Out[32]: 'py'

In [33]: word[4:]

Out[33]: 'on'

In [34]: word[-2:]

Out[34]: 'on'

In [35]: word[:2] + word[2:]

Out[35]: 'python'

In [36]: word[:4] + word[4:]

Out[36]: 'python'

In [37]: word[42]

         ---------------------------------------------------------------------------
         IndexError                                Traceback (most recent call last)
         Cell In[37], line 1
         ----> 1 word[42]

         IndexError: string index out of range

In [38]: word[4:42]

Out[38]: 'on'
```

```
In [39]: word[42:]

Out[39]: ''
```

```
In [40]: word[0] = 'J'

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[40], line 1
----> 1 word[0] = 'J'

TypeError: 'str' object does not support item assignment
```

```
In [41]: word[2:] = 'py'

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[41], line 1
----> 1 word[2:] = 'py'

TypeError: 'str' object does not support item assignment
```

```
In [42]: 'J' + word[1:]

Out[42]: 'Jython'
```

```
In [43]: word[:2] + 'py'

Out[43]: 'pypy'
```

```
In [44]: s = 'supercalifragilisticexpialidocious'
         len(s)

Out[44]: 34
```

Lists

```
In [45]: squares = [1, 4, 9, 16, 25]
         squares

Out[45]: [1, 4, 9, 16, 25]
```

```
In [46]: squares[0]

Out[46]: 1
```

```
In [47]: squares[-1]

Out[47]: 25
```

```
In [49]: squares[-3:]

Out[49]: [9, 16, 25]
```

In [50]:

```
squares + [36, 49, 64, 81, 100]
```

Out[50]:

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

In [51]:

```
cubes = [1, 8, 27, 65, 125]
```

In [52]:

```
4 ** 3
```

Out[52]:

```
64
```

In [53]:

```
cubes[3] = 64
```

In [54]:

```
cubes
```

Out[54]:

```
[1, 8, 27, 64, 125]
```

In [55]:

```
cubes.append(216)
cubes.append(7 ** 3)
cubes
```

Out[55]:

```
[1, 8, 27, 64, 125, 216, 343]
```

In [56]:

```
rgb = ["Red", "Green", "Blue"]
rgba = rgb
id(rgb) == id(rgba)
```

Out[56]:

```
True
```

In [57]:

```
rgba.append("Alph")
rgb
```

Out[57]:

```
['Red', 'Green', 'Blue', 'Alph']
```

In [58]:

```
correct_rgba = rgba[:]
correct_rgba[-1] = "Alpha"
correct_rgba
```

Out[58]:

```
['Red', 'Green', 'Blue', 'Alpha']
```

In [59]:

```
rgba
```

Out[59]:

```
['Red', 'Green', 'Blue', 'Alph']
```

In [60]:

```
letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
letters
```

Out[60]:

```
['a', 'b', 'c', 'd', 'e', 'f', 'g']
```

In [61]:

```
letters[2:5] = ['C', 'D', 'E']
letters
```

Out[61]:

```
['a', 'b', 'C', 'D', 'E', 'f', 'g']
```

```
In [62]: letters[2:5] = []
         letters
```

```
Out[62]: ['a', 'b', 'f', 'g']
```

```
In [63]: letters[:] = []
         letters
```

```
Out[63]: []
```

```
In [64]: letters = ['a', 'b', 'c', 'd']
         len(letters)
```

```
Out[64]: 4
```

```
In [65]: a = ['a', 'b', 'c']
         n = [1, 2, 3]
         x = [a, n]
         x
```

```
Out[65]: [['a', 'b', 'c'], [1, 2, 3]]
```

```
In [66]: x[0]
```

```
Out[66]: ['a', 'b', 'c']
```

```
In [67]: x[0][1]
```

```
Out[67]: 'b'
```

range() Function

```
In [68]: for i in range(5):
             print(i)
```

```
0
1
2
3
4
```

```
In [69]: list(range(5, 10))
```

```
Out[69]: [5, 6, 7, 8, 9]
```

```
In [70]: list(range(0, 10, 3))
```

```
Out[70]: [0, 3, 6, 9]
```

list(range(-10, -100, -30))

```
In [72]: a = ['Mary', 'had', 'a', 'little', 'lamb']
```

```
for i in range(len(a)):
    print(i, a[i])
```

```
0 Mary
1 had
2 a
3 little
4 lamb
```

In [73]:

```
range(10)
```

Out[73]:

```
range(0, 10)
```

In [74]:

```
sum(range(4))
```

Out[74]:

```
6
```

Data Structures

In [75]:

```
fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']
fruits.count('apple')
```

Out[75]:

```
2
```

In [76]:

```
fruits.count('tangerine')
```

Out[76]:

```
0
```

In [77]:

```
fruits.index('banana')
```

Out[77]:

```
3
```

In [78]:

```
fruits.index('banana', 4)
```

Out[78]:

```
6
```

In [79]:

```
fruits.reverse()
fruits
```

Out[79]:

```
['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange']
```

In [80]:

```
fruits.append('grape')
fruits
```

Out[80]:

```
['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange', 'grape']
```

In [81]:

```
fruits.sort()
fruits
```

Out[81]:

```
['apple', 'apple', 'banana', 'banana', 'grape', 'kiwi', 'orange', 'pear']
```

In [82]:

```
fruits.pop()
```

```
Out[82]: 'pear'
```

The del statement

```
In [83]: a = [-1, 1, 66.25, 333, 333, 1234.5]
         del a[0]
         a

Out[83]: [1, 66.25, 333, 333, 1234.5]

In [84]: del a[2:4]

In [85]: a

Out[85]: [1, 66.25, 1234.5]

In [86]: del a[:]
         a

Out[86]: []

In [87]: del a
```

Tuples and Sequences

```
In [88]: t = 12345, 54321, 'hello!'
         t[0]

Out[88]: 12345

In [89]: t

Out[89]: (12345, 54321, 'hello!')

In [90]: u = t, (1, 2, 3, 4, 5)
         u

Out[90]: ((12345, 54321, 'hello!'), (1, 2, 3, 4, 5))

In [91]: t[0] = 88888

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[91], line 1
----> 1 t[0] = 88888

TypeError: 'tuple' object does not support item assignment
```

In [92]:
```
v = ([1, 2, 3], [3, 2, 1])
v
```

Out[92]:
```
([1, 2, 3], [3, 2, 1])
```

In [93]:
```
empty = ()
singleton = 'hello',
len(empty)
```

Out[93]:
```
0
```

In [94]:
```
len(singleton)
```

Out[94]:
```
1
```

In [95]:
```
singleton
```

Out[95]:
```
('hello',)
```

In [96]:
```
x,y,z=t
```

SETS

In [97]:
```
basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
print(basket)
```

```
{'apple', 'pear', 'banana', 'orange'}
```

In [98]:
```
'orange' in basket
```

Out[98]:
```
True
```

In [99]:
```
'crabgrass' in basket
```

Out[99]:
```
False
```

In [100…
```
a = set('abracadabra')
b = set('alacazam')
a
```

Out[100…
```
{'a', 'b', 'c', 'd', 'r'}
```

In [101…
```
a - b
```

Out[101…
```
{'b', 'd', 'r'}
```

In [102…
```
a | b
```

Out[102…
```
{'a', 'b', 'c', 'd', 'l', 'm', 'r', 'z'}
```

In [103…
```
a & b
```

Out[103…

```
{'a', 'c'}
```

In [104…

```
a ^ b
```

Out[104…

```
{'b', 'd', 'l', 'm', 'r', 'z'}
```

In [105…

```
a = {x for x in 'abracadabra' if x not in 'abc'}
a
```

Out[105…

```
{'d', 'r'}
```

Dictionaries

In [106…

```
tel = {'jack': 4098, 'sape': 4139}
tel['guido'] = 4127
tel
```

Out[106…

```
{'jack': 4098, 'sape': 4139, 'guido': 4127}
```

In [107…

```
tel['jack']
```

Out[107…

```
4098
```

In [108…

```
del tel['sape']
tel['irv'] = 4127
tel
```

Out[108…

```
{'jack': 4098, 'guido': 4127, 'irv': 4127}
```

In [109…

```
list(tel)
```

Out[109…

```
['jack', 'guido', 'irv']
```

In [110…

```
sorted(tel)
```

Out[110…

```
['guido', 'irv', 'jack']
```

In [111…

```
'guido' in tel
```

Out[111…

```
True
```

In [112…

```
'jack' not in tel
```

Out[112…

```
False
```

In [113…

```
dict([('sape', 4139), ('guido', 4127), ('jack', 4098)])
```

Out[113…

```
{'sape': 4139, 'guido': 4127, 'jack': 4098}
```

In [114…

```
{x: x**2 for x in (2, 4, 6)}
```

Out[114…

```
{2: 4, 4: 16, 6: 36}
```

In [115…

```
dict(sape=4139, guido=4127, jack=4098)
```

Out[115…

```
{'sape': 4139, 'guido': 4127, 'jack': 4098}
```

In [116…

```
for i in reversed(range(1, 10, 2)):
    print(i)
```

```
9
7
5
3
1
```

In [117…

```
basket = ['apple', 'orange', 'apple', 'pear', 'orange', 'banana']
for i in sorted(basket):
    print(i)
```

```
apple
apple
banana
orange
orange
pear
```

In [118…

```
basket = ['apple', 'orange', 'apple', 'pear', 'orange', 'banana']
for f in sorted(set(basket)):
    print(f)
```

```
apple
banana
orange
pear
```

In [ ]: