# Numbers

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]:
```
5**3 #where** indicates power of number
```

Out[10]:
```
125
```

In [11]:
```
2**7
```

Out[11]:
```
128
```

In [12]:
```
width=20
height=5*9
width*height
```

Out[12]:
```
900
```

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 [24]:
```
price==price+(price*tax)
price
```

Out[24]:
```
127.1953125
```

# Text

In [25]:
```
'spam eggs'
```

Out[25]:
```
'spam eggs'
```

In [26]:
```
"paris rabbit got your back :)! yay!"
```

Out[26]:
```
'paris rabbit got your back :)! yay!'
```

In [27]:
```
'2003'
```

Out[27]:
```
'2003'
```

In [28]:
```
'doesn\'t'
```

Out[28]:
```
"doesn't"
```

In [29]:
```
"doesn't"
```

Out[29]:
```
"doesn't"
```

In [30]:
```
"yes, they said."
```

Out[30]:
```
'yes, they said.'
```

In [31]:
```
"\"yes,\" they said."
```

Out[31]:
```
'"yes," they said.'
```

In [32]:
```
'"isn\'t," they said.'
```

Out[32]:
```
'"isn\'t," they said.'
```

In [33]:
```
s='first line.\nsecond line.'
s
```

Out[33]:
```
'first line.\nsecond line.'
```

In [34]:
```
print(s)
```

```
first line.
second line.
```

In [35]:
```
print('c:\some\name')
```

```
c:\some
ame
```

In [36]:
```
print(r'c:\some\name')
```

```
c:\some\name
```

In [38]:
```
print("""\
Usage: thingy[options]
     -h                    Display this usage message
     -H hostname           Hostname to connect to
""")
```

```
\
Usage: thingy[options]
     -h                    Display this usage message
     -H hostname           Hostname to connect to

```

In [39]:
```
3*'un'+'ium'
```

Out[39]:
'unununium'

In [40]:
```
'py''thon'
```

Out[40]:
'python'

In [47]:
```
text=('put several strings within parentheses''to have them joined together.')
text
```

Out[47]:
'put several strings within parenthesesto have them joined together.'

In [53]:
```
prefix='py'
prefix +'thon'
```

Out[53]:
'python'

In [54]:
```
word='python'
word[0]
```

Out[54]:
'p'

In [55]:
```
word[5]
```

Out[55]:
'n'

In [56]:
```
word[-1]
```

Out[56]:
'n'

In [57]:
```
word[-2]
```

Out[57]:
'o'

In [58]:
```
word[-6]
```

Out[58]:
'p'

In [59]:
```
word[0:2]
```

Out[59]:
'py'

In [60]:
```
word[2:5]
```

Out[60]:

```
'tho'
```

In [61]:

```
word[:2]
```

Out[61]:

```
'py'
```

In [62]:

```
word[4:]
```

Out[62]:

```
'on'
```

In [63]:

```
word[-2:]
```

Out[63]:

```
'on'
```

In [64]:

```
word[:2]+word[2:]
```

Out[64]:

```
'python'
```

In [65]:

```
word[:4]+word[4:]
```

Out[65]:

```
'python'
```

In [66]:

```
word[32]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[66], line 1
----> 1 word[32]

IndexError: string index out of range
```

In [67]:

```
word[2:32]
```

Out[67]:

```
'thon'
```

# Lists

In [68]:

```
squares=[1,4,9,16,25]
squares
```

Out[68]:

```
[1, 4, 9, 16, 25]
```

In [70]:

```
squares[0]
```

Out[70]:
```
1
```

In [71]:
```
squares[-1]
```

Out[71]:
```
25
```

In [72]:
```
squares[-3:]
```

Out[72]:
```
[9, 16, 25]
```

In [73]:
```
squares+[36,49,64,81,100]
```

Out[73]:
```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

In [74]:
```
cubes=[1,8,27,65,125]
4**3
```

Out[74]:
```
64
```

In [76]:
```
cubes[3]=64
cubes
```

Out[76]:
```
[1, 8, 27, 64, 125]
```

In [77]:
```
rgb=["red","greeen","blue"]
rgba=rgb
id(rgb)==id(rgba)
```

Out[77]:
```
True
```

In [78]:
```
rgba.append("alpha")
rgb
```

Out[78]:
```
['red', 'greeen', 'blue', 'alpha']
```

In [79]:
```
correct_rgba=rgba[:]
correct_rgba[-1]="alpha"
correct_rgba
```

Out[79]:
```
['red', 'greeen', 'blue', 'alpha']
```

In [80]:
```
rgba
```

Out[80]:
```
['red', 'greeen', 'blue', 'alpha']
```

In [81]:
```
letters=['a','b','c','d','e','f','g']
letters
```

Out[81]:
```
['a', 'b', 'c', 'd', 'e', 'f', 'g']
```

In [82]:
```
letters[2:5]=['c','d','e']
letters
```

Out[82]:
```
['a', 'b', 'c', 'd', 'e', 'f', 'g']
```

In [83]:
```
letters[2:5]=[]
letters
```

Out[83]:
```
['a', 'b', 'f', 'g']
```

In [84]:
```
letters[:]
```

Out[84]:
```
['a', 'b', 'f', 'g']
```

In [85]:
```
letters[:]=[]
letters
```

Out[85]:
```
[]
```

In [86]:
```
letters=['a','b','c','d']
len(letters)
```

Out[86]:
```
4
```

In [87]:
```
a=['a','b','c']
n=[1,2,3]
x=[a,n]
x
```

Out[87]:
```
[['a', 'b', 'c'], [1, 2, 3]]
```

In [88]:
```
x[0]
```

Out[88]:
```
['a', 'b', 'c']
```

In [89]:
```
x[0][1]
```

Out[89]:
```
'b'
```

# First step towards programming

In [90]:

```
a,b=0,1
while a<10:
    print(a)
    a,b=b,a+b
```

```
0
1
1
2
3
5
8
```

In [91]:

```
i=256*256
print('the value of i is',i)
```

```
the value of i is 65536
```

In [92]:

```
a,b=0,1
while a<1000:
    print(a,end=',')
    a,b=b,a+b
```

```
0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,
```

# IF STATEMENTS

In [93]:

```
x=int(input("please enter an integer:"))
```

In [94]:

```
if x<0:
    x=0
    print('negative changed to zero')
elif x==0:
    print('zero')
elif x==1:
    print('singe')
else:
    print('more')
```

```
more
```

# For statements

In [95]:

```
words=['cat','window','defenstrate']
for w in words:
```

```
    print(w,len(w))
```

```
cat 3
window 6
defenstrate 11
```

In [109]:

```
users={'hans':'active','elenore':'inactive','dog':'inactive'}
#remove all active users
for user, status in users.copy().items():
    if status== 'active':
        del users[user]

#create dictionary of inactive users
inactive_users={}
for user, status in users.items():
    if status=='inactive':
        inactive_users[user]=status

print(inactive_users)
```

```
{'elenore': 'inactive', 'dog': 'inactive'}
```

# The Range() Function

In [110]:

```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

In [111]:

```
list(range(5,10))
```

Out[111]:

```
[5, 6, 7, 8, 9]
```

In [112]:

```
list(range(0,10,3))
```

Out[112]:

```
[0, 3, 6, 9]
```

In [113]:

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

Out[113]:

```
[-10, -40, -70]
```

In [114]:

```
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 [115]:

```
range(10)
```

Out[115]:

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

In [116]:

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

Out[116]:

```
6
```

## break and continue statements

In [117]:

```
for n in range(2,10):
    for x in range(2,n):
        if n%x==0:
            print(f"{n} equals {x}*{n//x}")
            break
```

```
4 equals 2*2
6 equals 2*3
8 equals 2*4
9 equals 3*3
```

In [118]:

```
for num in range(2,10):
    if num%2==0:
        print(f"found an even number {num}")
        continue
        print(f"found an odd numbe {num}")
```

```
found an even number 2
found an even number 4
found an even number 6
found an even number 8
```

## else clauses on Loops

In [119]:

```
for n  in range(2,10):
    for x in range(2,n):
        if n%x==0:
            print(n,'equals',x,'*',n//x)
            break
        else:
            print(n,'is a prime number')
```

```
3 is a prime number
4 equals 2 * 2
5 is a prime number
5 is a prime number
5 is a prime number
6 equals 2 * 3
```

```
7 is a prime number
7 is a prime number
7 is a prime number
7 is a prime number
7 is a prime number
8 equals 2 * 4
9 is a prime number
9 equals 3 * 3
```

In [ ]:

```

```