In [1]:

```
name = "python"
print(type(name))
```

```
<class 'str'>
```

In [2]:

```
s1='''Hi this is nsrit college
      welcome to second day of workshop
      lets learn something new'''
print(5)
print(6)
print(6)
```

```
5
6
6
```

#Python operators:

1. Arithmetic operators
2. Relational operators
3. Logical operators
4. Bitwise operators
5. Assignment operators
6. Membership operators
7. Identity operators

In [ ]:

```
help()
```

```

Welcome to Python 3.7's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at https://docs.python.org/3.7/tutorial/. (ht
tps://docs.python.org/3.7/tutorial/.)

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics".  Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "spam", type "modules spam".

help> keywords

Here is a list of the Python keywords.  Enter any keyword to get more hel
```

In [1]:

```
import keyword
print(keyword.kwlist,)
print(dir(keyword))
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'fo
r', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'no
t', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
['__all__', '__builtins__', '__cached__', '__doc__', '__file__', '__loader_
_', '__name__', '__package__', '__spec__', 'iskeyword', 'kwlist', 'main']
```

In [9]:

```
#Strings in python:
s = "Hello NSRIT Students"
print(s)
print(type(s))
print(len(s))
s1 = "CSE 2nd Yr Students"
print(s+s1)        #String concatenation
print(s*2)         #String multiplication
print(s+str(500))  #String addition by type casting
```

```
Hello NSRIT Students
<class 'str'>
20
Hello NSRIT StudentsCSE 2nd Yr Students
Hello NSRIT StudentsHello NSRIT Students
Hello NSRIT Students500
```

In [10]:

```
#String methods:
print(dir(str),end=" ")
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc_
_', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__
getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__ite
r__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new_
_', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__se
tattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize', 'casefo
ld', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'forma
t', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'is
digit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'i
stitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partit
ion', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstri
p', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'tran
slate', 'upper', 'zfill']
```

In [14]:

```
s = "python programming"
print(s.upper())        #Changing to uppercase.
print(s.isupper())      #Checking the condition for uppercase or not.
s1 = "APSSDC"
print(s1.lower())       #Changing to lowercase.
print(s.capitalize())   #Capitalizing the string 's'.
print(s.title())        #Changing the string 's' as title.
print(s1.title())       #Changing the string 's1' as title.
```

```
PYTHON PROGRAMMING
False
apssdc
Python programming
Python Programming
Apssdc
```

In [18]:

```
s = " Knowledge is WEALTH "
print(s.split())     #Splitting the string
#print(s.strip())     #Stripping the string
print(s.lstrip())
print(s.rstrip())
```

```
['Knowledge', 'is', 'WEALTH']
Knowledge is WEALTH 
 Knowledge is WEALTH
```

In [19]:

```
s = "Engineering students of vizag"
print(s.count("E"))
print(s.count("e"))
print(len(s))
```

```
1
3
29
```

In [24]:

```
#String boolean functions
a = "123"
print(a.isdigit())
s = "abc"
print(s.isalpha())
var = "abc123"
print(var.isalnum())
s1 = "S"
print(s1.islower())
print(s.isalnum())
print(s1.isdigit())
print(var.isalpha())
s2 = "Python Workshop"
print(s2.istitle())
s3 = "missing those days"
s4 = s3.swapcase()
print(s4)
print(s4.swapcase())
```

```
True
True
True
False
True
False
False
True
MISSING THOSE DAYS
missing those days
```

In [33]:

```
#String slicing:
s = "NSRIT Engineering College"
print(s[3])
print(s[7])
print(s[-7])
print(s[3:17])
print(s[-4:-1])
```

```
I
n
C
IT Engineering
leg
```

In [41]:

```
#Task-1:
s = input("Enter a palindrome:")
if(s == s[::-1]):
    print(s," is palindrome")
else:
    print(s," is not a palindrome")
```

```
Enter a palindrome:oyo
oyo  is palindrome
```

In [58]:

```
#Task-2:
s = "A P S S D C P Y T H O N"
s=s[::2]
s=s.capitalize()
s[::-1]
```

Out[58]:

```
'nohtypcdsspA'
```

In [60]:

```
s = "A P S S D C P Y T H O N"
s.replace("A P S S D C P Y T H O N","nohtypcdsspA")
```

Out[60]:

```
'nohtypcdsspA'
```

In [64]:

```
s = "A P S S D C P Y T H O N"
s=s[::2]
s=s[::-1]
s=s.lower()
s.replace("a","A")
```

Out[64]:

```
'nohtypcdsspA'
```

In [1]:

```
s = "A P S S D C P Y T H O N"
s=s.lower()
s=s.capitalize()
s=s[::-1]
s.replace(" ","")
```

Out[1]:

```
'nohtypcdsspA'
```

In [5]:

```
#Task-3
s =input("Enter a string:")
lower=0
upper=0
digit=0
alnum=0
for i in s:
    if i.isdigit():
        digit==1
    elif i.isupper():
        upper+=1
    elif i.islower():
        lower+=1
    else:
        alnum+=1
print(len(s))
print(lower)
print(upper)
print(digit)
print(alnum)
```

```
Enter a string:jsdhgf^%$#7654HFJYF
19
6
5
0
4
```

In [6]:

```
s = input("Enter a string:")
upper,lower,digit,alnum=0,0,0,0
for i in range(len(s)):
    if(s[i]>='A' and s[i]<='Z'):
        upper+=1
    elif(s[i]>='a' and s[i]<='z'):
        lower+=1
    elif(s[i]>='0' and s[i]<='9'):
        digit+=1
    else:
        alnum+=1
print(len(s))
print(upper)
print(lower)
print(digit)
print(alnum)
```

```
Enter a string:jwqhgf%$#643KHVF7654
20
4
6
7
3
```

In [ ]:

```
#Functions:
syntax:
    def functionname(arguments):
        statements......
    functionname(arguments)
```

In [ ]:

```
# Type of functions:
1. In built functions
2. User defined functions
```

In [ ]:

```
print()
input()
type()
range()
len()
def()
```

In [25]:

```
#Task-1:
#Print all even and odd numbers in range b/w 1 and 300 and also print their sum,sub,mul
def even():
    for i in range(1,300):
        if(i%2==0):
            print(i,end=" ")
def odd():
    for i in range(1,300):
        if(i%2!=0):
            print(i,end=" ")
even()
odd()
```

```
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54
56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100 102 10
4 106 108 110 112 114 116 118 120 122 124 126 128 130 132 134 136 138 140 14
2 144 146 148 150 152 154 156 158 160 162 164 166 168 170 172 174 176 178 18
0 182 184 186 188 190 192 194 196 198 200 202 204 206 208 210 212 214 216 21
8 220 222 224 226 228 230 232 234 236 238 240 242 244 246 248 250 252 254 25
6 258 260 262 264 266 268 270 272 274 276 278 280 282 284 286 288 290 292 29
4 296 298 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47
49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 9
9 101 103 105 107 109 111 113 115 117 119 121 123 125 127 129 131 133 135 13
7 139 141 143 145 147 149 151 153 155 157 159 161 163 165 167 169 171 173 17
5 177 179 181 183 185 187 189 191 193 195 197 199 201 203 205 207 209 211 21
3 215 217 219 221 223 225 227 229 231 233 235 237 239 241 243 245 247 249 25
1 253 255 257 259 261 263 265 267 269 271 273 275 277 279 281 283 285 287 28
9 291 293 295 297 299
```

In [28]:

```
#def evenoddcount():
even=0
odd=0
for i in range(1,300):
    if(i%2==0):
        even+=i
    else:
        odd+=i
#evenoddcount()
print(even)
print(odd)
print(even+odd)
print(even-odd)
```

```
22350
22500
44850
-150
```

In [35]:

```
#Task-2:
#Print table of user input number
print("1st table")
n=int(input("Enter a number:"))
for i in range(1,13):
    print(n,"*",i,"=",n*i)
print("----------")
print("2nd table")
m=int(input("Enter a number:"))
for i in range(1,13):
    print(m,"*",i,"=",m*i)
print("----------")
```

```
1st table
Enter a number:3
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
3 * 10 = 30
3 * 11 = 33
3 * 12 = 36
----------
2nd table
Enter a number:5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
5 * 11 = 55
5 * 12 = 60
----------
```

In [36]:

```
n=int(input("Enter a number:"))
for n in range(1,n+1):
    print("   ")
    print(n,"st table")
    print("----------")
    for i in range(1,13):
        print(n,"*",i,"=",n*i)
```

```
Enter a number:3

1 st table
----------
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
1 * 10 = 10
1 * 11 = 11
1 * 12 = 12

2 st table
----------
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
2 * 11 = 22
2 * 12 = 24

3 st table
----------
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
3 * 10 = 30
3 * 11 = 33
3 * 12 = 36
```

In [41]:

```
#Task-3:
#Print all primes in given range
n=int(input("Enter the range:"))
c=0
for i in range(1,n+1):
    if(n%i==0):
        c+=1
if(c==2):
    print("Prime number")
else:
    print("Not a prime number")
```

```
Enter the range:3
Prime number
```

In [42]:

```
#Task-4:
#Print the factorial of given number
n=int(input("Enter a number:"))
fact=1
for i in range(1,n+1):
    fact=fact*i
print("The factorial of ",n,"is",fact)
```

```
Enter a number:4
The factorial of  4 is 24
```

In [46]:

```
#Task-5:
#Check the number is perfect or not
n=int(input("Enter a number:"))
sum=0
for i in range(1,n):
    if(n%i==0):
        sum=sum+i
if(sum==n):
    print("Perfect number")
else:
    print("Not a perfect number")
```

```
Enter a number:28
Perfect number
```

In [ ]:

```
#Types of arguments
1. default arguments
2. keyword arguments
3. arbitrary arguments
```

In [47]:

```
#Default arguments
def workshop(college="NSRIT",place="VIZAG"):
    return "Python workshop at "+college+" "+place
workshop()
```

Out[47]:

```
'Python workshop at NSRIT VIZAG'
```

In [51]:

```
#Keyword arguments
def cock(action,colour):
    return "It is in "+colour+" colour and it can "+action
cock(colour="red",action="fly")
```

Out[51]:

```
'It is in red colour and it can fly'
```

In [50]:

```
#Arbitrary arguments
def players(*names):
    return names
players("Kohli","Rohit","Bumrah")
```

Out[50]:

```
('Kohli', 'Rohit', 'Bumrah')
```

In [ ]:

```

```