TOPIC : Artificial Intelligence with Python - INMOVIDU - Jahnavi N

PYTHON PROGRAMMIG

In [ ]:

FUCNTIONS

A function is a group of statements that together perform a task

we can divide a large program into the basic building blocks known as function, which only runs when it is called

Python Function Types There are two types of functions in Python.

built-in functions: The functions provided by the Python language such as print(), len(), str(), etc.

user-defined functions: The functions defined by us in a Python program.

build in functions

In [14]:

```
abs(-1.2)
```

Out[14]:

```
1.2
```

In [20]:

```
l=[-1,22332,2323,2,32]
max(l)
```

Out[20]:

```
22332
```

In [21]:

```
min(l)
```

Out[21]:

```
-1
```

In [23]:

```
l=[1,2,3,4]
sum(l)
```

Out[23]:

```
10
```

In [24]:

```
pow(2,3)
```

Out[24]:

```
8
```

In [44]:

```
round(18.5)
```

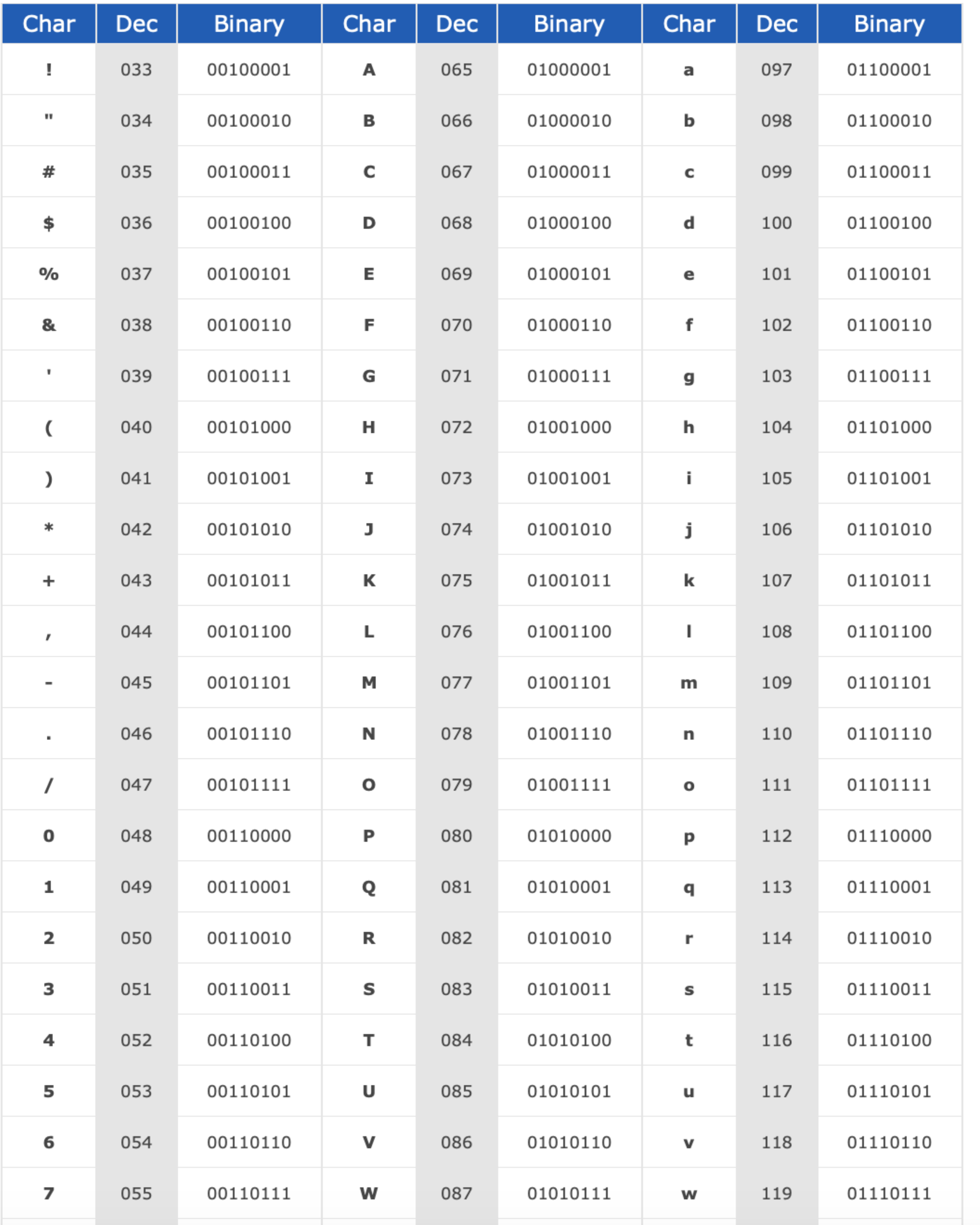
```
Out[44]: 18
```

math module

```
In [38]: import math
         math.ceil(2.9)

Out[38]: 3

In [41]: math.floor(2.1)

Out[41]: 2

In [46]: math.log(2,10)

Out[46]: 0.30102999566398114

In [47]: math.sqrt(25)

Out[47]: 5.0

In [48]: print(25**(1/2))

         5.0
```

ascii

```
In [50]: from IPython import display
         display.Image("11.png")

Out[50]:
```



Char	Dec	Binary	Char	Dec	Binary	Char	Dec	Binary
!	033	00100001	A	065	01000001	a	097	01100001
"	034	00100010	B	066	01000010	b	098	01100010
#	035	00100011	C	067	01000011	c	099	01100011
$	036	00100100	D	068	01000100	d	100	01100100
%	037	00100101	E	069	01000101	e	101	01100101
&	038	00100110	F	070	01000110	f	102	01100110
'	039	00100111	G	071	01000111	g	103	01100111
(	040	00101000	H	072	01001000	h	104	01101000
)	041	00101001	I	073	01001001	i	105	01101001
*	042	00101010	J	074	01001010	j	106	01101010
+	043	00101011	K	075	01001011	k	107	01101011
,	044	00101100	L	076	01001100	l	108	01101100
-	045	00101101	M	077	01001101	m	109	01101101
.	046	00101110	N	078	01001110	n	110	01101110
/	047	00101111	O	079	01001111	o	111	01101111
0	048	00110000	P	080	01010000	p	112	01110000
1	049	00110001	Q	081	01010001	q	113	01110001
2	050	00110010	R	082	01010010	r	114	01110010
3	051	00110011	S	083	01010011	s	115	01110011
4	052	00110100	T	084	01010100	t	116	01110100
5	053	00110101	U	085	01010101	u	117	01110101
6	054	00110110	V	086	01010110	v	118	01110110
7	055	00110111	W	087	01010111	w	119	01110111

Stands for "American Standard Code for Information Interchange." ASCII is a character encoding that uses numeric codes to represent characters. These include upper and lowercase English letters, numbers, and punctuation symbols

```
In [51]: ord('a')
```

```
Out[51]: 97
```

```
In [61]: ord('d')
```

```
Out[61]: 100
```

In [53]:
```
chr(65)
```

Out[53]:
```
'A'
```

rotate letters in string

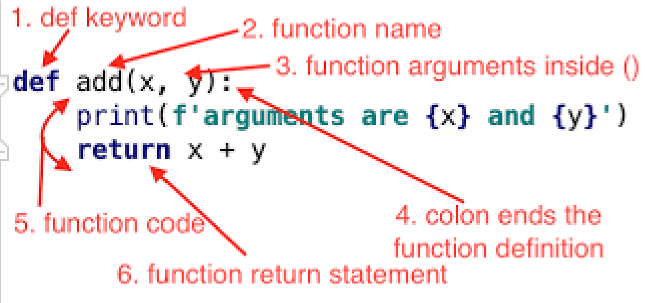
In [63]:
```
s="ABCDEF"
s1=""
for i in s:
    s1=s1+chr(ord(i)+1)

print("so string is ", s1)
```

```
so string is  BCDEFG
```

In [ ]:
```
# above with user defined movement
```

In [62]:
```
from IPython import display
display.Image("download.png")
```

Out[62]:

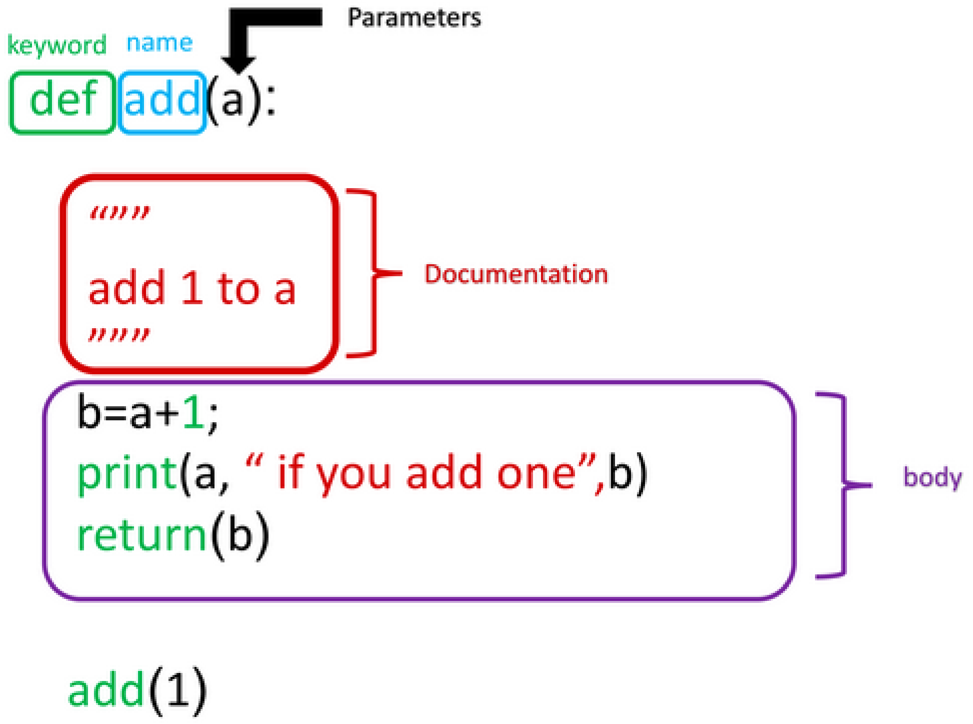


```
def add(x, y):
    print(f'arguments are {x} and {y}')
    return x + y
```

1. def keyword
2. function name
3. function arguments inside ()
5. function code
4. colon ends the function definition
6. function return statement

In [68]:
```
display.Image("https://i.imgur.com/9z8uo2n.png")
```

Out[68]:



```
def add(a):
    """
    add 1 to a
    """
    b=a+1;
    print(a, " if you add one",b)
    return(b)

add(1)
```

In [66]:

```
def helo(x):
    return x+10

y=helo(10)
print(y)
```

```
20
```

In [67]:

```
def fun():
    pass
print(type(fun))
```

```
<class 'function'>
```

In [69]:

```
def hello():
    print("Hello World!")

if __name__ == "__main__":
    hello()
```

```
Hello World!
```

In [72]:

```
def hello():
    print("Hello World!")
hello()
```

```
Hello World!
```

In [73]:

```
def add(a,b):
    c=a+b
    return c
x=2
y=5
z=add(x,y)
print(z)
```

```
7
```

In [75]:

```
def add(a,b):
    c=a+b
    print(c)
x=int(input("enter no1" ))
y=int(input("enter no2 "))
add(x,y)
```

```
enter no112
enter no2 13
25
```

add number in the list

In [77]:

```
l=[1,2,1,1]
print(add_list(l))

def add_list(l):
    s=0
    for i in l:
        s=s+i
    return s
```

```
5
```

In [78]:

```
def add_list(l):
    s=0
    for i in l:
        s=s+i
    return s

l=list(map(int,input("enter list numbers ").split(",")))
print(add_list(l))
```

```
enter list numbers 12,122,32, 323,23, 2
514
```

odd number

In [80]:

```
x=int(input(" enter "))
if (x%2 ==0):
    print('its  even')
else:
    print('its odd')
```

```
 enter 5
its odd
```

In [82]:

```
def check_odd(x):
    if(x%2!=0):
        return True
    else:
        return False


l=[1,5,2,24,9]
for i in l:
    if(check_odd(i)):
        print("{}  is odd".format(i) )
```

```
1  is odd
5  is odd
9  is odd
```

default parameter

In [84]:

```
def hai(x="everone"):
    return "welcome "+x
name="jahnavi"
print(hai(name))
```

```
welcome jahnavi
```

recursion - function call by function it self

In [85]:

```
def fun(x):
    print(x)
    if(x<=0):
        return 0
    fun(x-1)
```

```
# main
fun(5)
```

```
5
4
3
2
1
0
```

In [89]:

```
x=5
y=1
for i in range(1,6):
    y=y*i
print(y)
```

```
120
```

In [ ]:

```
def fac(x):
    if(x==0 or x==1):
        return 1
    return x*fac(x-1)
print(fac(5))
```

scope

In [90]:

```
#global
def fun():
    print(s)

s = "hai"
fun()
```

```
hai
```

In [92]:

```
#global
def fun():
    s="yo"
    print(s)
s= "hai"
print("here",s)
fun()
```

```
here hai
yo
```

In [99]:

```
#befor assign
def fun():
    print(s)
    #s = "hello"
    print(s)

s = "hai"
fun()
print(s)
```

```
hai
hai
hai
```

```
In [94]: x=10
         def fun():
             return x
         fun()

Out[94]: 10
```

```
In [95]: #error
         def fun():
             x1=10
             return x1
         fun()
         print(x1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-95-73e50271c7ad> in <module>
      3     return x1
      4 fun()
----> 5 print(x1)

NameError: name 'x1' is not defined
```

List comprehension

List comprehension offers a shorter syntax when you want to create a new list based on the values of an existing list.

```
In [ ]: l=[2,3,4,5,6,7,87,53,2,22]
        ll=[]
        for i in l:
            if(i>3):
                ll.append(i)
        print(ll)
```

```
In [100…: ll=[x**2 for x in l if x>3]
```

```
In [101…: print(ll)

[25, 576, 81]
```

new = [expression for item in iterable if condition]

```
In [ ]: l=[1,2,3]
        ll=[(x**2) for x in l]
```

```
In [ ]: print(ll)
```

```
In [ ]:
```