Statements

If

```
If w == alice :
        Print("cricket")
```

If -else

```
If w==alice:
    Print ("cricket")
else:
      Print("football")
```

elif

```
If -else                                (we can give many statements
If w== alice:                                  elif 1,2 etc… )
    Print ("cricket")
Elif w== rainy:
    Print(umbrella)
else:
      Print("football")
```

Nested if

```
If w==" sunny":
    if the time_is = "day":
          Print("play out side ")
    Else:
```

```
        Print("be in the house")
Elif w==night
      Print("sleep")
```

Code for a calculator

```
Num1=int(input ("no"))
Num2=int(input("no"))
Operator   #(logic) = input ("give the operator")
If operator== ("+"):
    print ("the addition is : ",num1+num2)
Elif operator == ("-"):
  Print (f"the sub is {num1-num2}")
Elif operator == ("*"):
     Print (f"the mul is {num1*num2}")
Elif operator == ("/"):
     Print (f"the divison is {num1/num2}")
Else :
    Print ("the input is in valid ")
```

Code for calculator :

```
N=int(input("give the number: "))
i=1
while i<=10:
print(f"{n}*{i}={n*i}")
i+=1
```

#for i in range (1,11)

Print(f"{n}*{i}={n*i})

Code for factorial :

N=int(input("enter a number: "))

Factorial = factorial*n

n-=1 #n=n-1

print(factorial)

Strings

Data type used to represent textual data " ", ' ', "" '"

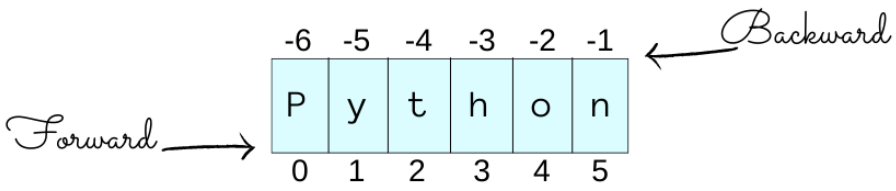
Ex : "hello"

Indexing

It can assin the numbes for string

There will be positive and negative indexing

(accessing of characters)



String slicing

0	1	2	3	4	5	6	7	8	9	10
H	e	l	l	o		W	o	r	l	d
-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Print(s[1])= e

Print(s[-2])=l

Print(s[1:3])=el

Print(s[1:-1])=ello_worl

Print(s[:4])=hell

Print(s[6:])=world

print (S[::2])=hlowrd

Print(s[1::2])=el_ol

Print(s[::-1])=dlrow_olleh

String concatentation

Basically adding of to strings

Name=”john”

Lastname = “son”

Full_name =name+” ”+lastname

Print(“Full_name :”)

Out put :

John son

Stirng length

Length of string

(“Hello_world”)=11

String methods #still need to refer f

Methods use to covert in uppercase,removing spaces,interchanging,splitting etc..

S=”Hello world”

Print(s.upper()) output: HELLO WORLD

Print(s.lower()) output: hello world

Print(s.replace(‘0’,’a’)) output:hallo warld

Print(s.count(l)) output: 3

#counts the letter how many times it repeted

Print(s.strip()) #output: ‘Hello, world’

String formatting

Name=”Alice”

Age = 30

Print(f”my name is {name} and I am {age} years old”)

#print(“my name is () and I am () years old”, name,age) without formatting

operators

logical operators

-=

+=

>=

<=

statements

while loops

```
i=1
while i<=5
i=i+1
print ("i")
```

for loop)

```
numbers=[ 1,2,3,4,5]
for item in numbers :          #output 12345
 print(item)
```

- Continue: the certain item will be skiped
- Break : just it will break the code

list []

pandas we use like pd #uses in data frames

#uses new variable df

```
#Import the Pandas framework, defined as pd
import pandas as pd

#Define our color variable as a list
color = ['blue','green','red','yellow']
#Define our fruit variable as a list
fruit = ['blueberry','apple','cherry','banana']

#Define the df variable as formatted columns for color and fruit
df=pd.DataFrame(columns=['color', 'fruit'])
#Label our columns as color and fruit
df['color'],df['fruit']=color,fruit

#Print everything
print(df)
```

output

```
0   color    fruit
1   blue      blueberry
2   green     apple
```

```
3   red      cherry
3   yellow   banana
```

tuples()

tuples are like list but we cant change or add elements

empty_tuple () #a tuple which is empty

integer_tuple() #integer_tuple=(1,2,3)

output: (1,2,3)

mixed_tuple(0,Hello,1.2,”world”)

n_tuple = ("kubernetes", "cloud native", [8, 6, 7, 5, 3, 0, 9])
print(n_tuple[0])
output: Kubernetes
print(n_tuple[0][2])
output: b

Print(n_tuple[0:2]

Output : (Kubernetes, cloud-native)

We can add tuples concatenation

tuple('the_number')

print(('after ')+('the_number'))

#output: after the_number

Sets

As like tuples ans lists sets will not have duplicate elements #same elements twice like {1,2,2,3} ={1,2,3}

Set can also be used to remove duplicates from a list!

We cant change data in sets we can add elements

insert,remove

Range

num = range (5)

print("number")

- Grade calculator

- Palindrome

Radar is a palindrom same even if you reverse

```
S=input("give a string ")
Reverse=s[::-1]
If reverse==s
    Print("the given input is a palindrome")
Else:
    print("it is not a palindrome ")
```

Sort

Sort is arranging the data in a particular oreder or formate

Bubble sort:

Arrage data by swapping like comparing elements

Function

Declaration /defining

```
Function_sum = sum(a,b)
                    #parameters
        Print(5+2)
                #arguments
```

Postions arguments (a,b)(5,2) a=5 b=2

Keyword arguments(a,b)add(b=5,a=2) #still need reference

Recursion :

The function calling itself

```
Def demo()
    Print("hello")
Demo()
#output: hello
```

Now we see how recursion works

```
Def demo()
   Print(hello)
    Demo ()
Demo ()   #output = hello (print in loop)
```

Indirect recurring

```
def num(n)
    if n<=0:
        return 0
    print(n,end= " ")
    num1(n-1)
def num1(n):
    print(n,end=" ")
    num(n-1)
num1(10)
#output 10,9,8,7,6,5,4,3,2,1
```

Factorial through recursion

```
n=int(input("Enter number :"))

def fact(n):
    if n==0:
       return 1
    else:
       return n*fact(n-1)
result=fact(n)
print(f"factorial of {n}is {result}")
   #output : Enter number :3
factorial of 3is 6
```