Today topics

- Jump Statements
  - Break
  - Continue
  - pass
- while loop
- String concept

In [1]:
```
# write a program to find given number is prime or not
# 23 (1,2,3,.....,22,23)   # total no of factors 2
# 23 (2,3,4,.....21,22)   # total number of factors 0
```

In [4]:
```
n = int(input())  # 7
c = 0  #2
for i in range(1,n+1):#1,2,3,4,5,6,7
    if n%i==0: # 7%1==0, 7%2==0, 7%7==0
        c=c+1    # c=1,2
if c==2:
    print("prime")
else:
    print("not a prime")

```

```
103
prime
```

In [5]:
```
n = int(input())  # 7
c = 0  #2
for i in range(2,n):# 2,3,4,5,6
    if n%i==0: #  7%2==0, 7%3==0
        c=c+1
if c==0:
    print("prime")
else:
    print("not a prime")

```

...

Jump statements

- break
- continue
- pass

In [9]:
```python
# take the inputs from the user and check
# whether 5 is there in the  given range

n1 = int(input())
n2 = int(input())
for i in range(n1,n2+1):  #1,2,3,4,5,
    if i ==5:
        print("5 is there")
        break
    else:
        print("5 is not there")


```

...

In [18]:
```python
s = "Pythonnnnnnnn"
for i in s:# Python nnnnnn
    print(i)
    if i=="n":  #
        print("n is there")
        break

```

...

In [22]:
```python
s = "Python@Programming"
for i in s:
    print(i,end = " ")
```

```
P y t h o n @ P r o g r a m m i n g
```

In [33]:
```python
for i in s: # P y t h o n @ P r o g r a m m i n g
    if i == "@" or i == "g":
        print("continue is there")
        continue
    print(i, end = " ")
    print("yes")
    print("no")
```

```
p yes
no
y yes
no
t yes
no
h yes
no
o yes
no
n yes
no
```

In [37]:
```python
# pass statement
s = "Python@Programming"
for i in s:  # p y t h o n
    if i == "@" and i == "g":
        print("h is there")
        #pass  #continue
    print(i, end = " ")
    print("yes")
    print("no")
```

...

While loop

syntax: while condition:

- statements
incerement/decrement

In [38]:
```python
n = 1
while n<=10:  #11
    print(n)
    n=n+1

```

...

In [42]:
```python
# take the input from user and print all the numbers from input to 50
n = int(input())  #  30
for i in range(n,50):
    print(i,end = " ")
```

```
100
```

In [44]:
```python
n = int(input())
while n<=50:
    print(n)
    n+=1
```

```
100
```

In [1]:
```python
n = int(input()) #5
while n!=0:
    s= int(input()) #50
    t = s/n  #50/0
    print(t)
    break
```

```
5
30
6.0
```

In [1]:
```
n = int(input())
for i in range(1,10):
    s = int(input())
    t = s/n
    print(t)
    break
```

...

In [19]:
```
balance = 10000   # 9000
max_with_draw = balance-1000  # 9000
pin = "8956"
user_pin = input() #8956
while pin==user_pin:
    with_amount = int(input())
    if max_with_draw> with_amount:
        print("take your money")
        break
    else:
        print("limit exceed")
        break
print("enter valid pin")
```

```
1
enter valid pin
```

In [20]:
```
n = int(input())
n1 = int(input())
for i in range(n,n1):#1,2,3,4,5
    t = int(input())
    print(t)
```

```
1
1
```

In [1]:
```
n = int(input()) #3
c=0 #1,2
while n>c:  #3>0 ,3>1,3>2
    name = input()
    age = int(input())
    c=c+1
    print(name)
    print(age)
```

...

String

- immutable ==> we cannot change string once defined

In [21]:
```
s = ""
type(s)
```

Out[21]: str

In [10]:
```
s = "Hello world"
# all the operations on string will based on indexing
```

In [23]:
```
s+"@"
```

Out[23]: 'Hello world@'

In [24]:
```
s
```

Out[24]: 'Hello world'

String elements accessing

- based on indexing only

In [25]:
```
s[5]
```

Out[25]: ' '

In [26]:
```
s[9]
```

Out[26]: 'l'

In [27]:
```
s[-3]
```

Out[27]: 'r'

String slicing

syntax: s[starting index:ending index:step count]

In [28]:
```
s[3:8] # 3,4,5,6,7
```

Out[28]: 'lo wo'

In [29]:
```
s[3:8:2]  #3,5,7
```

Out[29]: 'l o'

In [30]:
```
s[:10:2] #0,2,4,6,8,
```

Out[30]: 'Hlowr'

In [31]:
```
s[0::2]  #0,2,4,6,8,10
```

Out[31]: 'Hlowrd'

In [32]:
```
len(s)
```

Out[32]: 11

In [11]:
```
s[::3]  # 0,3,6,9
```

Out[11]: 'Hlwl'

In [7]:
```
# find the length of the string without using len function
s = "Hello"
c=0
for i in s:
    print(i)
    c=c+1 #1
    print(c)
print(c)

```

...

In [ ]:
```

```