

```
python basics
- Variables

- Data types

- Type casting , how to convert one data type to another data type

- print statements

- basic codes

- input , eval , round

- packages or libraries i.e.very important

- conditional statements

- exceptional handling

- functions

- for and while
python core
- strings

- list

  - list comprehension

- tuple # my students

- dictionary

- lambda functions

- file handling

30days ==== learn

python advanced

- oops EDA ML

python + stats : EDA
python + ML :
python + DL
python+ NLP
python+Genai
```

Strings

- Initializations
- Inbuilt functions
 - print
 - type
 - len
 - min
 - max
 - sum
 - sorted
 - reversed
- Concatenation
- Index operations
- Mutable vs Immutable
- Slice
- Methods

Intializations

```
str1='python'
```

```
str2="python"
```

```
str3="I like 'python'"
str3
```

```
↗ "I like 'python'"
```

```
str4='I like "Python"'
str4
```

```
↗ 'I like "Python"'
```

```
str1='python'
type(str1) # ans
```

```
↗ str
```

```
print(str1)
```

```
↗ python
```

```
str1='python'
len(str1)
```

```
↗ 6
```

```
min(str1)
```

```
↗ 'h'
```

```
max(str1)
```

```
↗ 'y'
```

```
min('123')
ord('1'),ord('p')
```

```
↗ (49, 112)
```

```
min('python123')
```

```
↗ '1'
```

```
sum('python')
```

```
↗ -----
TypeError                                Traceback (most recent call last)
Cell In[19], line 1
----> 1 sum('python')

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
sum('123')
```

```
↗ -----
TypeError                                Traceback (most recent call last)
Cell In[21], line 1
----> 1 sum('123')

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
type
print
len
min
max
sum # Not working
```

```
sorted('python')
```

```
↵ ['h', 'n', 'o', 'p', 't', 'y']
```

iterable

- anything can iterate through loop
- string list tuple dictionary

```
# sorted('python')
sorted('python',reverse=False)
# ascending order
```

```
↵ ['h', 'n', 'o', 'p', 't', 'y']
```

```
sorted('python',reverse=True)
# descending order
```

```
↵ ['y', 't', 'p', 'o', 'n', 'h']
```

```
sorted('python')
sorted('python',reverse=False)
sorted('python',reverse=True)
```

reversed

- sequence means string list tuple dictionary

```
reversed('python')
# It is already reversed
# answer located a memory
# <memory>
```

```
↵ <reversed at 0x1da57a941f0>
```

```
out=reversed('python')
for i in out:
    print(i,end=' ')
```

```
↵ n o h t y p
```

```
list(reversed('python'))
```

```
↵ ['n', 'o', 'h', 't', 'y', 'p']
```

Concatenation

```
str1='hello'
str2='python'
str1+str2
```

```
↵ 'hellopython'
```

```
str1-str2
```

```
↵ -----
TypeError                                 Traceback (most recent call last)
Cell In[43], line 1
----> 1 str1-str2

TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

```
str1/str2
```

```
↵ -----
TypeError                                 Traceback (most recent call last)
Cell In[45], line 1
----> 1 str1/str2

TypeError: unsupported operand type(s) for /: 'str' and 'str'
```

```
str1*str2
```



```
-----
TypeError                                Traceback (most recent call last)
Cell In[47], line 1
----> 1 str1*str2

TypeError: can't multiply sequence by non-int of type 'str'
```

```
str1-str2
```

```
TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

```
str1/str2
```

```
TypeError: unsupported operand type(s) for /: 'str' and 'str'
```

```
str1*str2
```

```
TypeError: can't multiply sequence by non-int of type 'str'
```

```
'python'*2
```



```
'pythonpython'
```

```
'python'+'python'
```



```
'pythonpython'
```

index

```
# () indicates function call
```

```
# [] access the elements or values
```

```
-6  -5  -4  -3  -2  -1
p   y   t   h   o   n
0   1   2   3   4   5
```

```
str1='python'
```

```
str1[0],str1[-6]
```



```
('p', 'p')
```

```
str1[6]
```



```
-----
IndexError                                Traceback (most recent call last)
Cell In[59], line 1
----> 1 str1[6]

IndexError: string index out of range
```

```
str1='python'
```

```
print(str1[0])
```

```
print(str1[1])
```

```
print(str1[2])
```

```
print(str1[3])
```

```
print(str1[4])
```

```
print(str1[5])
```

```
#print(str1[i]) i== 0 to 6
```



```
p
y
t
h
o
n
```

```
str1='python hello'
```

```
n=len(str1) # human count only
```

```
for i in range(n):
```

```
    print(str1[i])
```



```
p
y
t
h
o
n
```

```
h
e
l
l
o
```

```
str1='python'
n=len(str1) # human count only
for i in range(n):
    print(str1[i],end=' ')
```

```
→ p y t h o n
```

```
for i in str1:
    print(i,end=' ')
```

```
→ p y t h o n
```

```
i in range() ==== i behave as number
str1[i]
i in str1      ==== i behave as element
i
```

Note

- when you have a use case with index then use range method access
- when you have a use case with out index then use in method access

```
# Q1) WAP ask the user get a indexes of 'a' from a given string
#     str1='hai hai hai'
#     ans: 1,5,9
```

```
# idea:
# step-1: iterate each letter using range
# step-2: if str1[i]=='a'
# step-3:     i
str1='hai hai hai'
n=len(str1)
for i in range(n):
    #print(n-i,str1[i],i)
    if str1[i]=='a':
        print(i)
# step-1: i=0  if str1[0]=='a'  'h'=='a'  F
# step-2: i=1  if str1[1]=='a'  'a'=='a'  T  print(1)
```

```
→ 1
   5
   9
```

```
# Q2) Wap ask the user how many 'a' are present
#     str1='hai hai hai'  ans=3
```

```
str1='hai hai hai'
n=len(str1)
count=0
for i in range(n):
    if str1[i]=='a':
        count=count+1
print(count)
```

```
→ 3
```

```
str1='hai hai hai'
count=0
for i in str1:
    if i=='a':
        count=count+1
print(count)
```

```
→ 3
```

```
# Q3) Wap ask the user find the all vowels ('a e i o u')
# str1='hello hai how are you'
# all vowels and count also

str5='hello hai how are you'
```

```

str6='aeiou'
count=0
for i in str5: 'h'
    for j in str6:  #'a' e i o u
        if i==j:
            print(i)
            count+=1
print(count)

praneeth kankanti 20:26
str1 = 'hello hai how are you'
count = 0
for i in range(0,len(str1)):
    if str1[i] == 'a' or str1[i] == 'e' or str1[i] == 'i' or str1[i] == 'o' or str1[i] == 'u':
        count +=1
count

```

```

# 1: No need of range here
str1 = 'hello hai how are you'
for i in str1:
    if i in 'aeiou':
        print(i)

# step-1: i='h'  if 'h' in 'aeiou'  F
# step-2: i='e'  if 'e' in 'aeiou'  T

```

```

↔ e
o
a
i
o
a
e
o
u

```

```

# Q4) Update above code with
# only one vowel

```

Empty string

```

s=''
s=s+'p'
s=s+'y'

```

```

↔ 'py'

```

```

str1='python'
s=''
for i in str1:
    s=s+i
print(s)

```

```

↔ python

```

```

# Q5) reverse a string
# str1='python'
# s='nohtyp'

```

```

# Q6) remove the spaces in a string
# str1='hai how are you'
# s='haihowareyou'

```

```

# Q7) remove digits from string
# str1='hai123hello435'
# s='haihello'

```

```

#Q8) str1='banana'
#      s='b@n@n@'

```

```

↔ 'b@n@n@'

```

Start coding or [generate](#) with AI.

