Topics

- typecasting
- implicit typecasting
- input
- indexing
- slicing

In [1]:

```
# typecasting
# transition from one to data type to another

# why ? it important?
```

In [7]:

```
x='10' #string
y='20' #string

print(int(x)+int(y))
#y+x not working this prefer upper formula to add
```

```
30
```

In [8]:

```
# upon addition , concatenate
fname = 'vipul'
lname = 'pandey'

fname+' '+lname
```

Out[8]:

```
'vipul pandey'
```

In [9]:

```
# typecasting
# builin function

#int()
#float()
#str()
# bool()
```

In [11]:

```
x='10' #string
print('value of x',x)
print(type(x))
print('_'*100)
# print(str(type(x).split()[1].replace('>','')))
int(x) # operation/ temporary basis
print('value of x',x)
print(type(x))
print('_'*100)
x=int(x) # assigning the values
print('value of x',x)
print(type(x))
```

```
value of x 10
<class 'str'>
_______________________________________________________________________________
_____________________
value of x 10
<class 'str'>
_______________________________________________________________________________
_____________________
value of x 10
<class 'int'>
```

In [13]:

```python
# implicit typecasting

x='10'
y=13
#x+y not possible because str + int not possible on this case
```

In [22]:

```python
x='vipul'
y=10
#x+y in this also string is not add to int
```

In [19]:

```python
#numerical data types : int and float
y=50 #int
x=10.0 #float
x+y
```

Out[19]:

```
60.0
```

In [ ]:

```python
# implicit typecasting

# order
# string()
#float()
#int()
#bool()
```

In [17]:

```python
x=10.9 #floot
y=5
x+y
```

Out[17]:

```
15.9
```

In [18]:

```python
a=True
int(a)
float(a)
str(a)
```

Out[18]:

```
'True'
```

In [24]:

```python
# top to bottom
# caution
# loss of data
```

```
a= '10'
int(a)
```

Out[24]:

```
10
```

In [26]:

```
b='vipul'
#int(b) this is also not possible
```

In [16]:

```
# ladder
a='10.9'
print('value of a',a)
print(type(a))
print('_'*100)
a=float(a)
print('value of a',a)
print(type(a))
print('_'*100)
a=int(a)
print('value of a',a)
print(type(a))
print('_'*100)
a=bool(a)
print('value of a',a)
print(type(a))
print('_'*100)
```

```
value of a 10.9
<class 'str'>
_______________________________________________________________________________
_____________________
value of a 10.9
<class 'float'>
_______________________________________________________________________________
_____________________
value of a 10
<class 'int'>
_______________________________________________________________________________
_____________________
value of a True
<class 'bool'>
_______________________________________________________________________________
_____________________
```

In [6]:

```
# input
# built in function

name=input('Please enter your name: \t')
grade=input('Please enter your marks:\t')
```

In [7]:

```
print(name)
print(grade)
```

```
Vipul
A+
```

In [9]:

```python
# sum--> program

x = int(input('enter first number'))
y = int(input('enter second number'))

# x= int(x)
# y = int(y)

#print('sum of x and y is ',int(x)+int(y))

print('sum of x and y is',x+y)
```

```
sum of x and y is 155
```

In [10]:

```python
type(x)
```

Out[10]:

```
int
```

In [7]:

```python
# dynamic binding
# c++

# int a = 10
a=10
a=10.1
a='vipul'
```

In [8]:

```python
# indexing

a='vipul'
# variable_name[index]
a[0]
```

Out[8]:

```
'v'
```

In [9]:

```python
a[2]
```

Out[9]:

```
'p'
```

In [10]:

```python
str2='Prabhakarna Sripalawardhana Attapattu Jayasuriya Laxmansriramkrishna Shi
```

In [11]:

```python
print(str2)
```

```
Prabhakarna Sripalawardhana Attapattu Jayasuriya Laxmansriramkrishna Shivavenka
ta Rajasekara Srinivasana Trichipalli Ekkaparampeer Perambadur Chinnaswami Muth
uswami Venugopal Iyer
```

In [26]:

```python
# len built in function
print(len(a))
print(len(str2))
print(str2[len(str2)-4])
```

```
# counting
print(str2[178])#here we use 178 insted of 179 last value because we take (n-1
#negative indexing
print(a[-4])
print(str2[-4])
```

```
6
179
I
r
u
I
```

In [13]:

```
# slicing
# syntax
# variable_name [ start:end:step_size]

a='gaurav'
a[0:3]
```

Out[13]: 'gau'

In [14]:

```
a[1:5]
```

Out[14]: 'aura'

In [15]:

```
a[-3:]
# right to left --> not possible
# left to right -->possible
```

Out[15]: 'rav'

In [16]:

```
# step_size
n1='123456789'
print(n1[::2])
```

```
13579
```

In [17]:

```
print(n1[1::4])
```

```
26
```

In [18]:

```
print(n1[0:5:2])
```

```
135
```

In [19]:

```
# reverse string(reverse the variable or any text)
print(n1[::-1])
a='gaurav'
print(a[::-1])
```

```
987654321
varuag
```

In [ ]: