Data Science & AI & NIC - Param

Python-For Data Science

Operators

Lecture No.- 01

Recap of Previous Lecture

Topic Language Fundamentals - 03

Topics to be Covered

Topic Operators - 01

Topic : Arithmetic Operators

$$\begin{tikzcd} & \text{operator} \\ 2+3 \arrow[ur, bend left] \arrow[d] \\ \text{operands} \end{tikzcd}$$

Operator ⟸ ◯ 1 ⟹ code

$\underbrace{+,\ -,\ \times,\ /}$, //, %, **

+ ⟹ works on string also

b/w both operands must be of type str (concatenation)

"Pankaj" + 2

+

Ud be laat ⟸

Slide 4

Topic : Relational Operators

<, <=, >, >=

$10 < 20 \rightarrow$ True

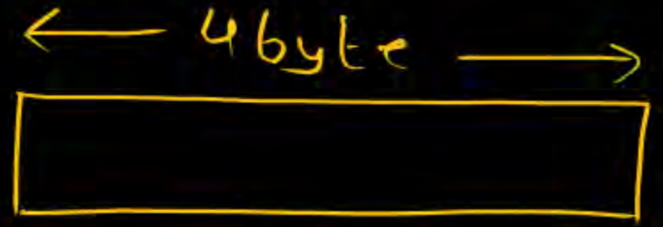
$20 > 10 \rightarrow$ True

$10 < 10 \rightarrow$ False

$10 <= 10 \rightarrow$ True

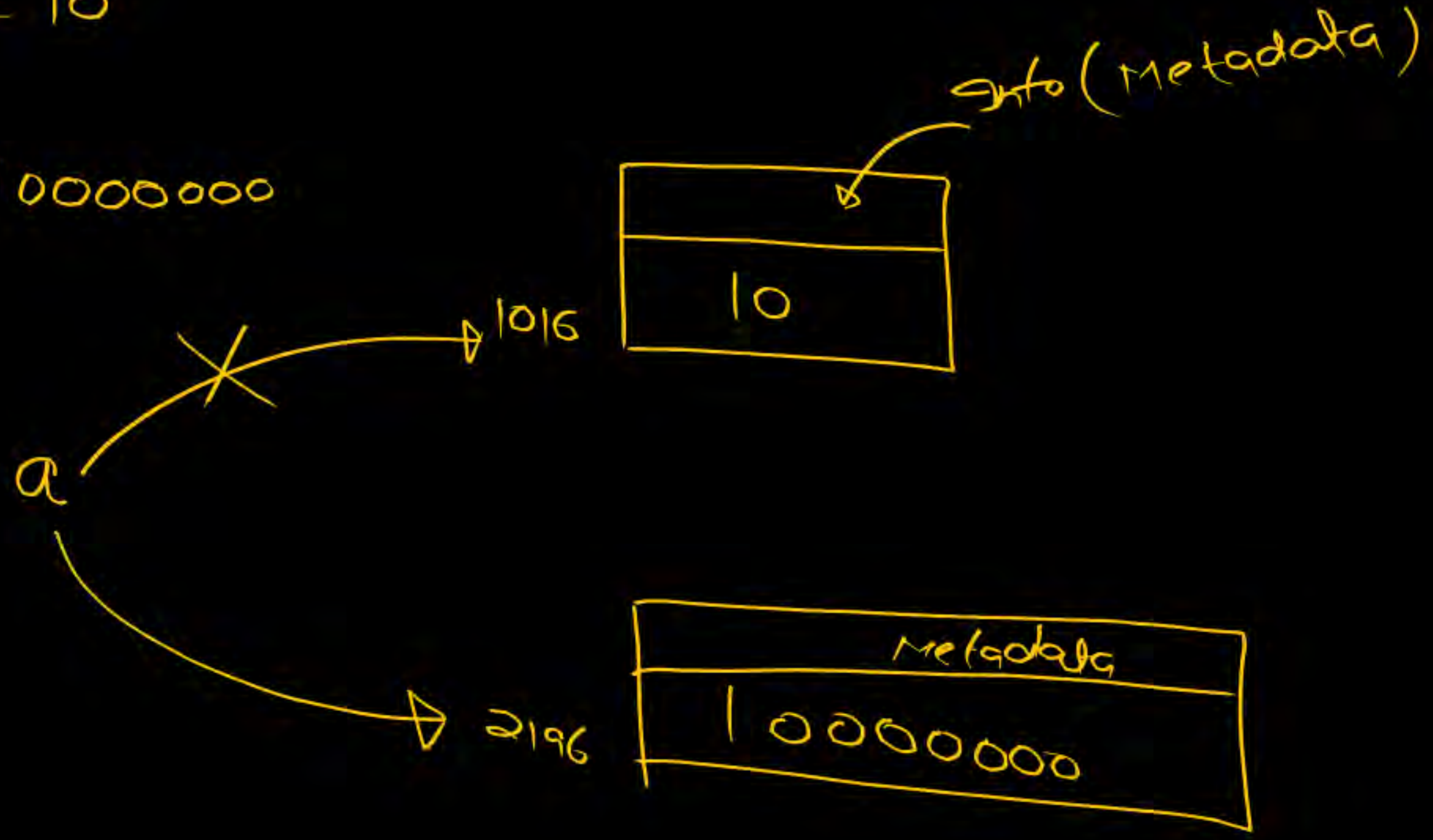
int a = 10

C++/Java/C



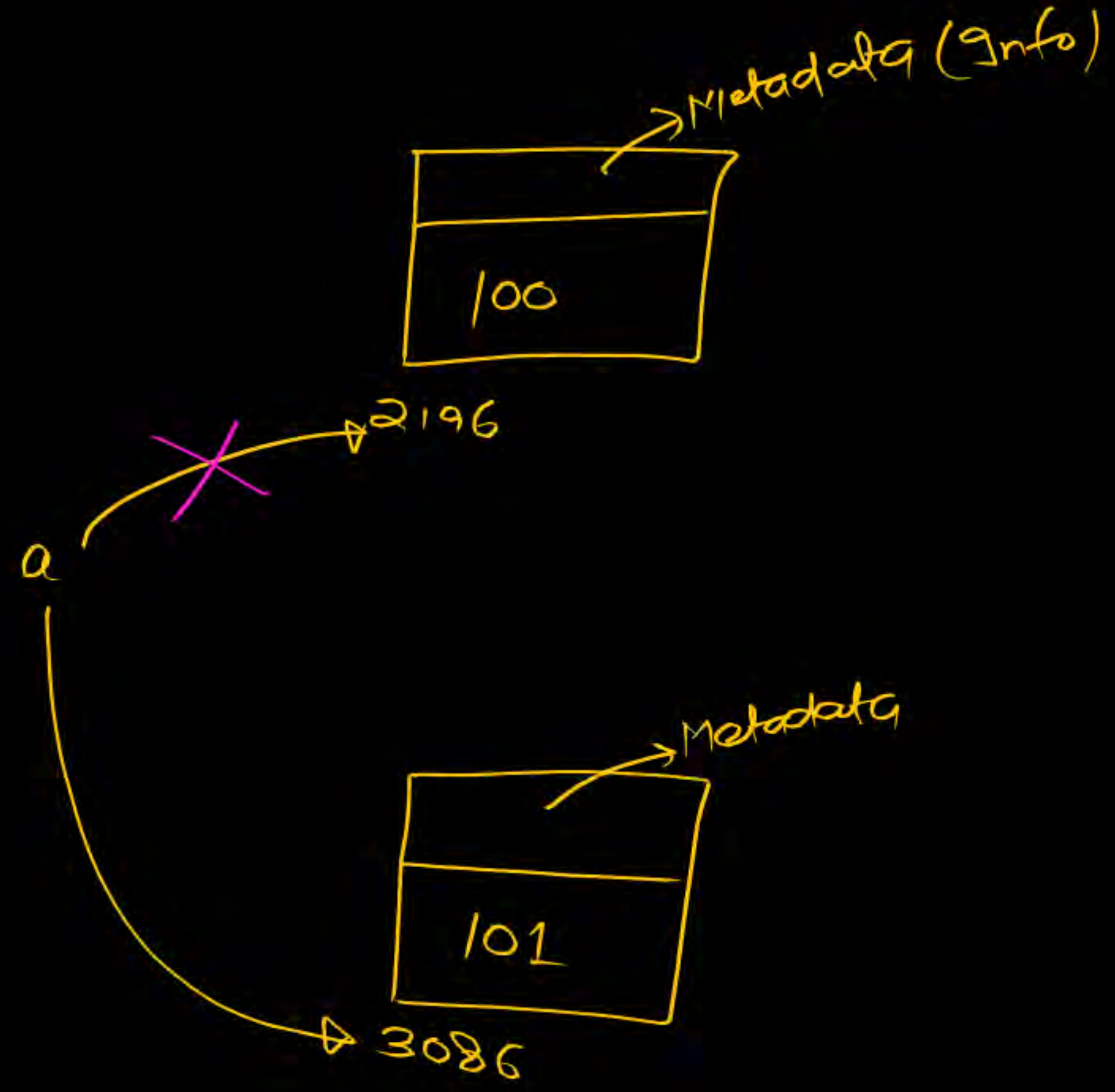
a = 10

a = 10000000



$a = 100$

$a = \boxed{a+1}$



input

To add 2 number

```
a = input()
b = input()
```

Equality operators (works for diff types also)

```
==          10 == 13.2          False
!=          10 == "Pankaj"      False
```

Logical Operators

Q ✓ ______
______

OR

✓ ______
______

(i) and (और) ⇒ No choice → if both arguments (operands) are True, the result is True otherwise False

(ii) or (अथवा) ⇒ choice → if atleast one argument is True, the result is True otherwise False (when both arg. are False)

(iii) not ⇒

not(True)	False
not(False)	True

			result
True	and	True	True
True	and	False	False
False	and	True	False
False	and	False	False

			result
True	or	True	True
True	or	False	True
False	or	True	True
False	or	False	False

$$\begin{tikzcd} a \arrow[d] & \text{and} & b & {} \arrow[l, bend right] \\ \text{False} & & & \end{tikzcd}$$

if a is False then

it return a otherwise b

non-boolean

$$\begin{tikzcd}
0 \arrow[r] & \text{False} \\
\text{Everything else} \arrow[r] & \text{True}
\end{tikzcd}$$

Shayan

$$\begin{tikzcd}
 & 12 \arrow[r] & \text{True} \\
 & 12.3 \arrow[ur] & \\
0.3 \arrow[uurr, bend left] & 0.3 \arrow[uur] & \\
 & \text{"Pankaj"} \arrow[uuur] & \\
 & \text{"ab"} \arrow[uuuur] & \\
 & 0.0 \arrow[r, dash] & \text{False} \\
 & 0+0j \arrow[ur] & \\
 & 0 \arrow[uur] & \\
 & \text{""} \arrow[uuur] &
\end{tikzcd}$$

a and b

if a is False

returns

a

shayan 0x $\boxed{}$

$\Rightarrow$

Shubham

"Pankaj" and $\bigcirc$"sharma"

$\downarrow$

C lang.

$$\begin{tikzcd}
a \arrow[d] & \text{or} & b & \times \arrow[l, bend right] \\
\text{True} & & &
\end{tikzcd}$$

or $\Rightarrow$ if atleast one arg/operand is True

$\Rightarrow$ result is True

$\Rightarrow$ if a is True, it returns a

otherwise it return b

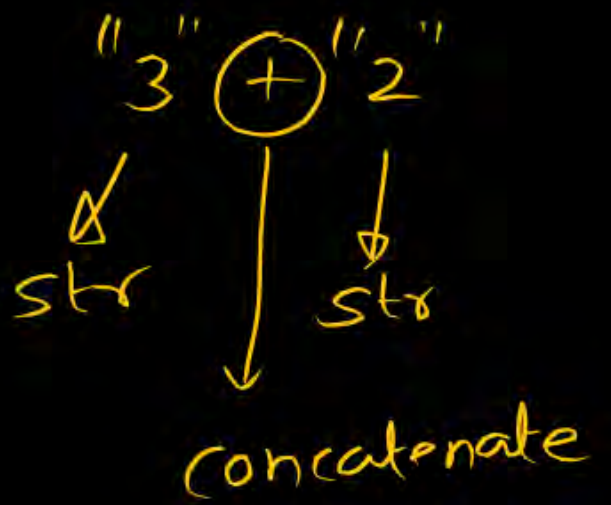
False or False

not (False)   True

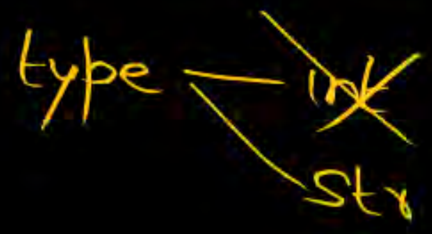
not (True)   False

not (2) $\rightarrow$ True

Pizza $\Longrightarrow$



32



"string" + 10

Topic : Logical Operators

Covered

and

or

not

Topic : Bitwise Operators

Non-CS ⇒ google ⇒

binary number system (0,1)

1 digit ⇒ bit

decimal (0-9)

binary (0,1)

$$23 \longrightarrow \overset{\text{binary}}{?}$$

$$23 \longrightarrow 10111$$

2) 23 (11
22
1 ← Rem

2	23	Rem
2	11	1
2	5	1
2	2	1
2	1	0
Stop	0	1

2) 2 (1
2
0 ← Rem

2) 1 (0
0
1 ← Rem

binary $\longrightarrow$ decimal

$$\begin{matrix} 1 & 0 & 1 & 1 & 1 \\ 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{matrix}$$

$$\Rightarrow 1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 1\times 2^1 + 1\times 2^0$$

$$= 1\times 16 + 0 + 1\times 4 + 1\times 2 + 1\times 1$$

$$= 16+0+4+2+1$$

$$= \boxed{23}$$

$$\begin{matrix} 9 & 3 & 4 \\ 10^2 & 10^1 & 10^0 \end{matrix}$$

$$9\times 10^2 + 3\times 10^1 + 4\times 10^0$$

bitwise operator

bits

binary value

$1 \& 0 \Rightarrow 0$
$0 \& 1 \Rightarrow 0$
$0 \& 0 \Rightarrow 0$
$1 \& 1 \Rightarrow 1$

(i) & : bitwise and operator

$a = 5 \& 6$

$5 \Rightarrow$ binary $\Rightarrow$ 0101 ✓
&
$6 \Rightarrow$ binary $\Rightarrow$ 0110

0100

$2^3\ 2^2\ 2^1\ 2^0$

$\Rightarrow \boxed{4}$

(ii) | : bitwise OR

$a = 5 \mid 6$

$$\left.\begin{array}{l} 0 \mid 1 = 1 \\ 1 \mid 0 = 1 \\ 1 \mid 1 = 1 \end{array}\right\}$$ if atleast one bit is 1, result is 1

$0 \mid 0 = 0$

$$\begin{array}{lll} 5 & \Rightarrow & 0101 \\ 6 & \Rightarrow & 0110 \\ \hline & & 0111 \\ \hline & & 2^2\,2^1\,2^0 \end{array}$$

$\Rightarrow 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

$= 4 + 2 + 1$

$= \boxed{7}$

(iii) XOR (^)

1 ^ 1 = 0

0 ^ 0 = 0

1 ^ 0 = 1

0 ^ 1 = 1

} if bits are diff ⇒ 1

iv) ~

v) <<

vi) >>

a = 5 ^ 6

5 ⇒ 0101

6 ⇒ 0110

0011

$2^3\ 2^2\ 2^1\ 2^0$

$\Rightarrow 2^1 + 2^0$

$= 2 + 1$

$= \textcircled{3}$

```
In [1]: a=10
        b=20

In [2]: a+b

Out[2]: 30

In [3]: a-b

Out[3]: -10

In [4]: a*b

Out[4]: 200

In [5]: b/a

Out[5]: 2.0

In [6]: a=10
        b=4

In [8]: a/b#always floating point arithmetic , retrun float value

Out[8]: 2.5

In [9]: a//b #floor division

Out[9]: 2

In [10]: #// it can perform both floor division as well as floating
         20.0//6

Out[10]: 3.0

In [11]: 20.0//6.0

Out[11]: 3.0

In [12]: 20/6

Out[12]: 3.3333333333333335

In [13]: #modulas operator a%b ===>remainder when a is divided by b
         20%3

Out[13]: 2

In [14]: #power operator **
         #2**10  1024
         2**10
```

```
Out[14]: 1024
```

```
In [15]: a="pankaj"
         b="sharma"
         a+b
```

```
Out[15]: 'pankajsharma'
```

```
In [16]: "pankaj" + 10
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[16], line 1
----> 1 "pankaj" + 10

TypeError: can only concatenate str (not "int") to str
```

```
In [17]: #we can also use string type with * operator
         #but in that case one argument must be int type and other must be str type
         "pankaj" * 2
```

```
Out[17]: 'pankajpankaj'
```

```
In [18]: 2*"pankaj"
```

```
Out[18]: 'pankajpankaj'
```

```
In [19]: "pankaj"*"sharma"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[19], line 1
----> 1 "pankaj"*"sharma"

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

```
In [20]: 2.5*"pankaj"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[20], line 1
----> 1 2.5*"pankaj"

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

```
In [21]: a=2+3j
         b=3+5j
```

```
In [22]: a+b
```

```
Out[22]: (5+8j)
```

```
In [23]: a=10
         b=20
```

```
In [24]: a<b
```

```
Out[24]: True

In [25]: a>b

Out[25]: False

In [26]: a<=b

Out[26]: True

In [27]: a>=b

Out[27]: False

In [28]: # relational operator is applicable to string also

In [29]: a="pankaj"
         b="shamra"

In [30]: a<b  #p comes before s in dictionary

Out[30]: True

In [31]: a>b

Out[31]: False

In [32]: a="neeraj"
         b="neetu"
         a<b #r comes before t in dictionary order

Out[32]: True

In [33]: a=100
         id(a)

Out[33]: 140732873744264

In [34]: a=a+1
         id(a)

Out[34]: 140732873744296

In [35]: 2 + 3j <4+5j

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[35], line 1
----> 1 2 + 3j <4+5j

TypeError: '<' not supported between instances of 'complex' and 'complex'

In [36]: True<True
```

```
Out[36]: False
```

```
In [37]: #chaining is possible
         10<20<30<40  #result kya ayega 10<20==>true , 20<30==>true , 30<40==>true
```

```
Out[37]: True
```

```
In [38]: 10<20>30<40
```

```
Out[38]: False
```

```
In [39]: a=input()
         b=input()
```

```
20
30
```

```
In [40]: a+b
```

```
Out[40]: '2030'
```

```
In [41]: a=input()
         print(type(a))
```

```
20
<class 'str'>
```

```
In [42]: #input()==>takes input in the form of string
         a=input()#20 ==>'20'
         b=input()#30 ==>'30'
         a+b # '20' + '30' ==>'2030'
```

```
20
30
Out[42]: '2030'
```

```
In [43]: #string ===>integer conversion     int('20') ===>20   int('30')==>30
```

```
In [44]: a=input()
         n1=int(a)
         b=input()
         n2=int(b)
```

```
10
20
```

```
In [45]: n1
```

```
Out[45]: 10
```

```
In [46]: n2
```

```
Out[46]: 20
```

```
In [47]: type(n1)
```

```
Out[47]: int
```

```
In [48]: type(n2)
```

```
Out[48]: int
```

```
In [49]: a= int(input())
         b=int(input())
         c=a+b
         print(c)
```

```
10
20
30
```

```
In [50]: #logical operators ===> works on all type of data
         #and
         10 and 20 #10==>true return 20
```

```
Out[50]: 20
```

```
In [51]: 0 and 20 #1st operamnd is false ==>it return first operand as answer
```

```
Out[51]: 0
```

```
In [52]: 10 and "pankaj" #1st operand is true ==> it returns 2nd operand as answer
```

```
Out[52]: 'pankaj'
```

```
In [53]: 10.0 and 2+3j
```

```
Out[53]: (2+3j)
```

```
In [54]: 2+3j  and "pankaj"
```

```
Out[54]: 'pankaj'
```

```
In [55]: "pankaj" and "shamra"
```

```
Out[55]: 'shamra'
```

```
In [56]: #for string only empty string represent False
         "" and "pankaj"
```

```
Out[56]: ''
```

```
In [57]: #a or b
         10 or 20 # 1st boperand is true ==> the answer is 1st operand
```

```
Out[57]: 10
```

```
In [59]: 0 or 20.3#1st operand is flase ===>the answer is 2nd operand
```

```
Out[59]: 20.3
```

```
In [60]: "pankaj" or "sharma"

Out[60]: 'pankaj'

In [61]: "" or "pankaj"

Out[61]: 'pankaj'

In [62]: not 10

Out[62]: False

In [63]: not 10.0

Out[63]: False

In [64]: not "pankaj"

Out[64]: False

In [65]: not 0

Out[65]: True

In [66]: not 0.0

Out[66]: True

In [67]: not ""

Out[67]: True

In [68]: not 0+0j

Out[68]: True

In [69]: #bitwise operators ===> int and bool pe work karte hain only
         a=5 & 6

In [70]: a

Out[70]: 4

In [71]: a= 10.3 & 4 #ud ke laat

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[71], line 1
----> 1 a= 10.3 & 4

TypeError: unsupported operand type(s) for &: 'float' and 'int'

In [72]: True & True
```

```
Out[72]: True

In [73]: a=5|6
         a

Out[73]: 7

In [74]: a=5^6

In [75]: a

Out[75]: 3

In [ ]:
```

THANK - YOU