

In C++

```
int a = 3;
```

Python

```
a = 3          (backed integer)
```

Actual issue.

Binary type

Hummer

name = "Hummer"

(Backed string)

Mean to binary dena cha rahi thi

① name = b"Hummer"

② name = "Hummer"

bin-name = bin(name)

print(bin-name)

output 'Hummer' (binary)

The correct way is to

use variable-name.encode('utf-8')

for string we do this

~~Ex 4~~

~~set data = { 1, 2, 3 }~~

~~print (type(set-data))~~
~~if output~~
~~<class 'set'>~~

Question - can't we use \times instead * while multiplying?

~~Reason~~

Answer - No, we can't

Reason - Python \times (variable
names, string)

Floor Division - On slide 6 under
Arithmetic Operator

Floor division also counts
Symbol is $\Rightarrow //$

$=$ (Assign right pr job ke
ga wo left walli ko assign
hw ga)

for example

$$a = 5$$

$==$ (for equality, mt lb
left walli cheese equals
right bary equal)

for example,

$$a = 5$$

$$b = 5$$

→ output

$$(a == b)$$

True

$$+=$$

$$a = 10$$

$$a += 5$$

$$a = a + 5$$

$$\textcircled{1} a = 10$$

$$\textcircled{2} a = 10 + 5$$

$$a = 15$$

print(a)
output
15

$$-=$$

$$a = 10$$

$$a -= 5$$

$$\rightarrow a = a - 5$$

$$a \approx 10$$

$$a = 10 - 5$$

$$a = 5 \text{ output}$$

S | O

$\frac{5+10}{10}$
O - Rem

$*$ =

$$a = 10$$

$$a^* = 5$$

$$\text{G } a = a^* \oplus 5$$

D - Rem

$$a = 10$$

$$a^* = 5$$

$$a = 10 * 5$$

$$a = 50 \text{ output}$$

/ =

$$a = 10$$

$$a/ = 5$$

$$\text{G } a = a/ 5$$

$$a = 10$$

$$a = 10 / 5,$$

$$a = 2 \text{ output}$$

% =

$$a = 10$$

$$a \% = 5$$

$$\text{G } a = a \% 5$$

$$a = 10$$

$$a = 10 \% 5$$

$$a = 0 \text{ output}$$

$a + 3$
 $a + b$
 $a + 3$
 $a - 3$
 $a \% 3$

} Expressions

→ Expressions must equal new value
for example, e

Statement → num = arb X