



# PYTHON DAY - 3





# DATA TYPES



# • DATA TYPES

1. Numeric
2. Sequence
3. Set
4. Dictionary
5. Boolean



# SET



```
x = {"apple", "orange", "banana"}
```

- Set is a collection of unique items enclosed in curly braces {}
- The items in a set have no defined order, and can be of any type
- sets are mutable
- Duplicate is not allowed

# DUPLICATES WILL BE IGNORED



```
x = {"apple", "orange", "banana", "orange"}
```

```
print(x)
```

# ADD



```
• • •  
  
x = {"apple", "orange", "banana"}  
  
x.add("kiwi")  
  
print(x)
```

- Use add method to add new items in the set

# UPDATE



```
x = {"apple", "orange", "banana"}
```

```
y = {"kiwi", "mango"}
```

```
x.update(y)
```

```
print(x)
```

- Use **update** method to  
combine two set

# REMOVE



```
• • •  
  
x = {"apple", "orange", "banana"}  
  
x.remove("orange")  
  
print(x)
```

- Use remove method to remove specified item from set



# DIFFERENCE



```
x = {1, 2, 3, 4, 5, 6}
```

```
y = {5, 6, 7, 8, 9, 10}
```

```
print(x.difference(y))
```

- To find the values exist in x alone

# INTERSECTION



```
x = {1, 2, 3, 4, 5, 6}
```

```
y = {5, 6, 7, 8, 9, 10}
```

```
print(x.intersection(y))
```

- To find common values exist in both

# SYMMETRIC DIFFERENCE



```
x = {1, 2, 3, 4, 5, 6}
```

```
y = {5, 6, 7, 8, 9, 10}
```

```
print(x.symmetric_difference(y))
```

- All the items which are not common will be in the output set

# DICTIONARY



```
dict = {"name" : "john", "age" : 30, "country" : "India"}
```

- Dictionary is a collection of key-value pairs enclosed in curly braces {}
- Dictionaries are also mutable, key-value pairs can be added, removed, or modified after they are created.
- Duplicates are not allowed

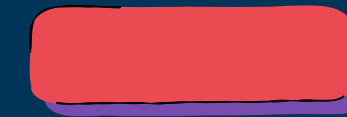
# BOOLEAN



- Boolean is a data type that can have one of two values: True or False
- often used in conditional statements, comparison statements



# OPERATORS



# • OPERATORS



1. Arithmetic operators
2. Relational operators
3. Assignment operators
4. Logical operators
5. Membership operators

# 1. ARITHMETIC OPERATORS



2 + 3 -----> addition

43 - 25 -----> subtraction

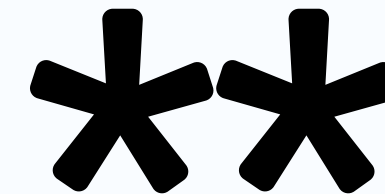
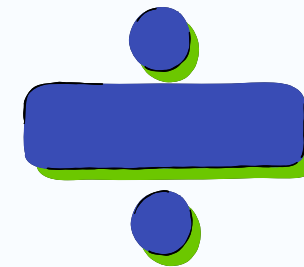
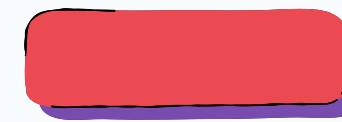
6 \* 5 -----> multiplication

120 / 3 -----> division

10 % 3 -----> modulus

2 \*\* 3 -----> exponentiation

10 // 3 -----> floor division





# 2. RELATIONAL OPERATORS



5 == 5 -----> Equal to

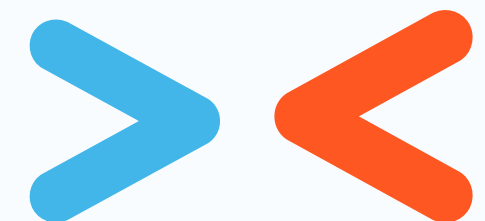
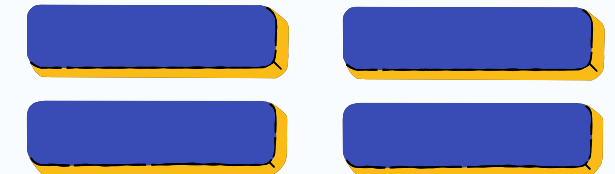
5 != 5 -----> Not equal to

5 > 3 -----> Greater than

5 < 3 -----> Less than

5 >= 5 -----> Greater than or equal to

5 <= 3 -----> Less than or equal to



# 3. ASSIGNMENT OPERATORS



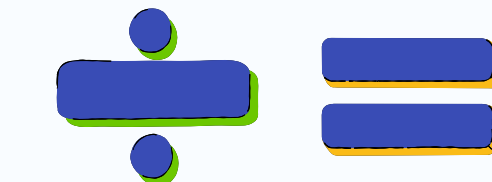
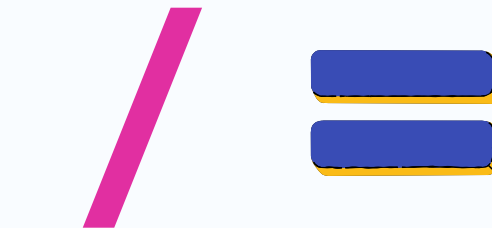
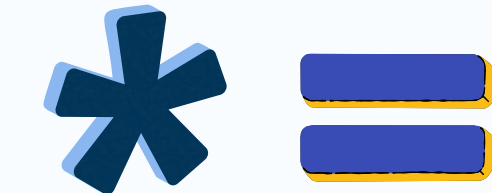
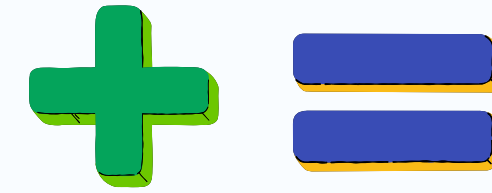
$x = x + 1$   $\text{-----}>$   $x += 1$

$x = x - 1$   $\text{-----}>$   $x -= 1$

$x = x * 2$   $\text{-----}>$   $x *= 2$

$x = x / 3$   $\text{-----}>$   $x /= 3$

$x = x \% 2$   $\text{-----}>$   $x \% = 2$



# 4. LOGICAL OPERATORS



**or**

**and**

**not**

X	Y	X OR Y	X AND Y	NOT X
true	true	true	true	false
true	false	true	false	false
false	true	true	false	true
false	false	false	false	true

# 5. MEMBERSHIP OPERATORS

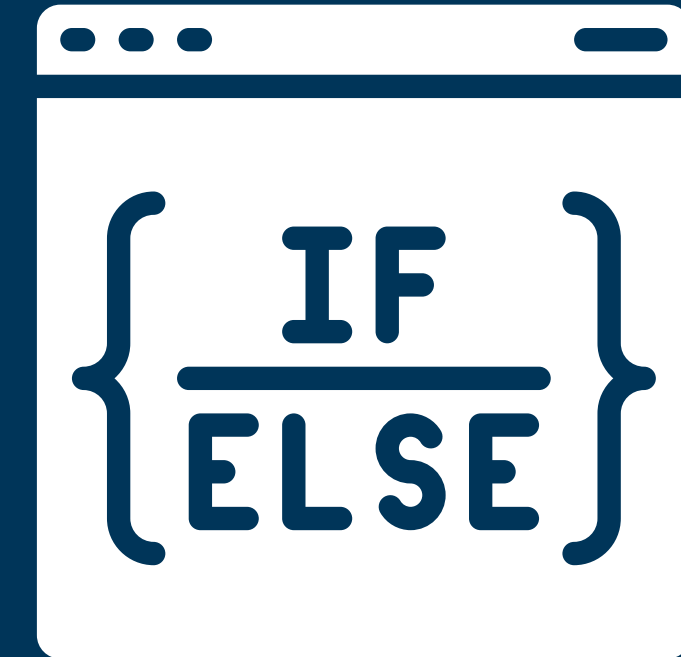


`in`

`not in`



# CONDITIONAL STATEMENTS



# • **CONDITIONAL STATEMENTS**



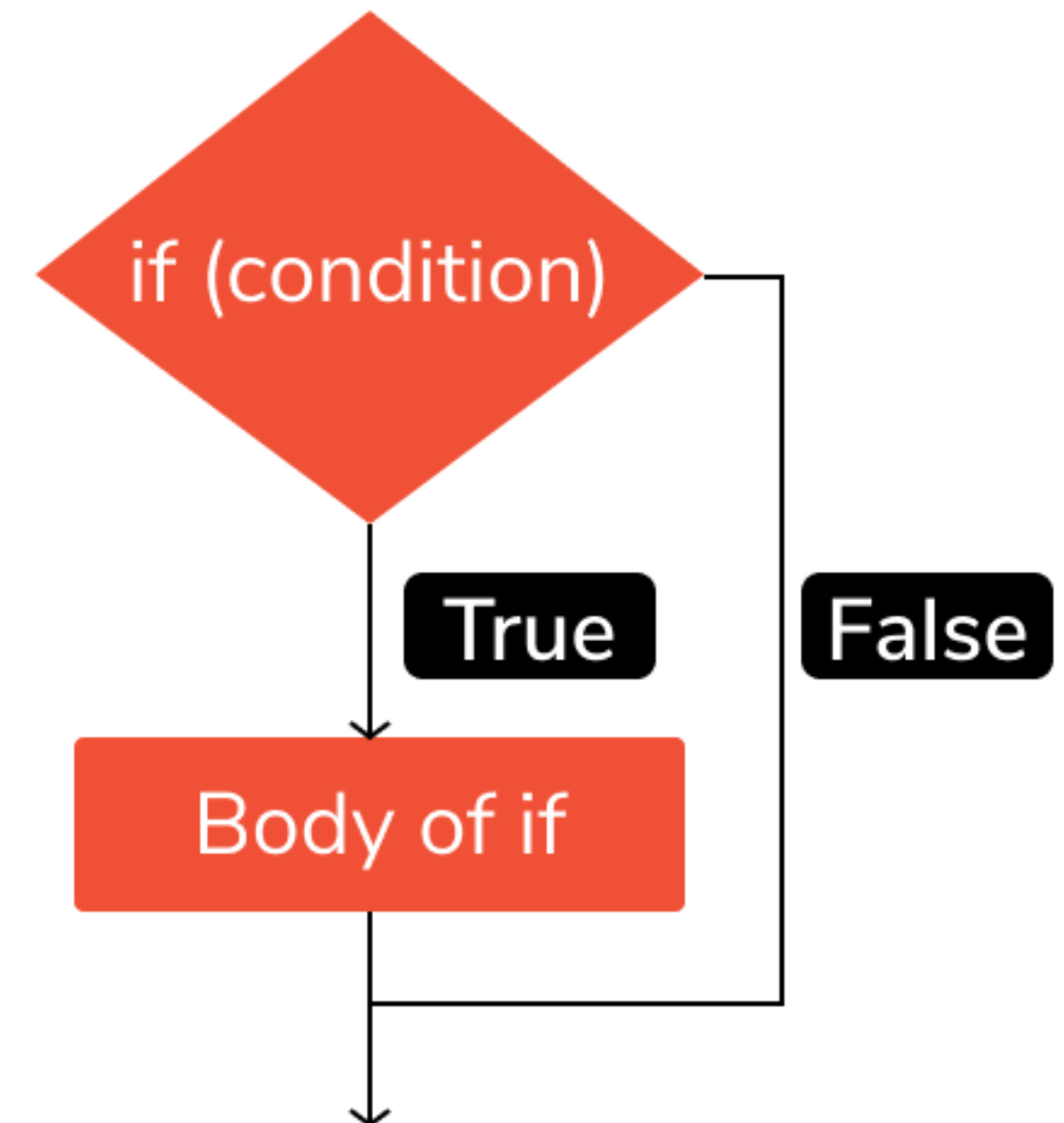
1. if
2. elif
3. if elif else
4. nested if



# CONDITIONAL STATEMENTS



- If statements in Python are used to control the flow of a program
- They allow the program to make decisions based on certain conditions



# • if



```
a = 33
```

```
b = 200
```

```
if b > a:
```

```
    print("b is greater than a")
```



# intendation



```
● ● ●  
  
a = 33  
b = 200  
if b > a:  
    print("b is greater than a")
```



```
● ● ●  
  
a = 20  
b = 30  
if b > a:  
print("b is greater than a")
```



# • elif



```
● ● ●  
  
a = 50  
b = 50  
  
if b > a:  
    print("b is greater than a")  
elif a == b:  
    print("a and b are equal")
```

# • else



```
a = 300
```

```
b = 20
```

```
if b > a:
```

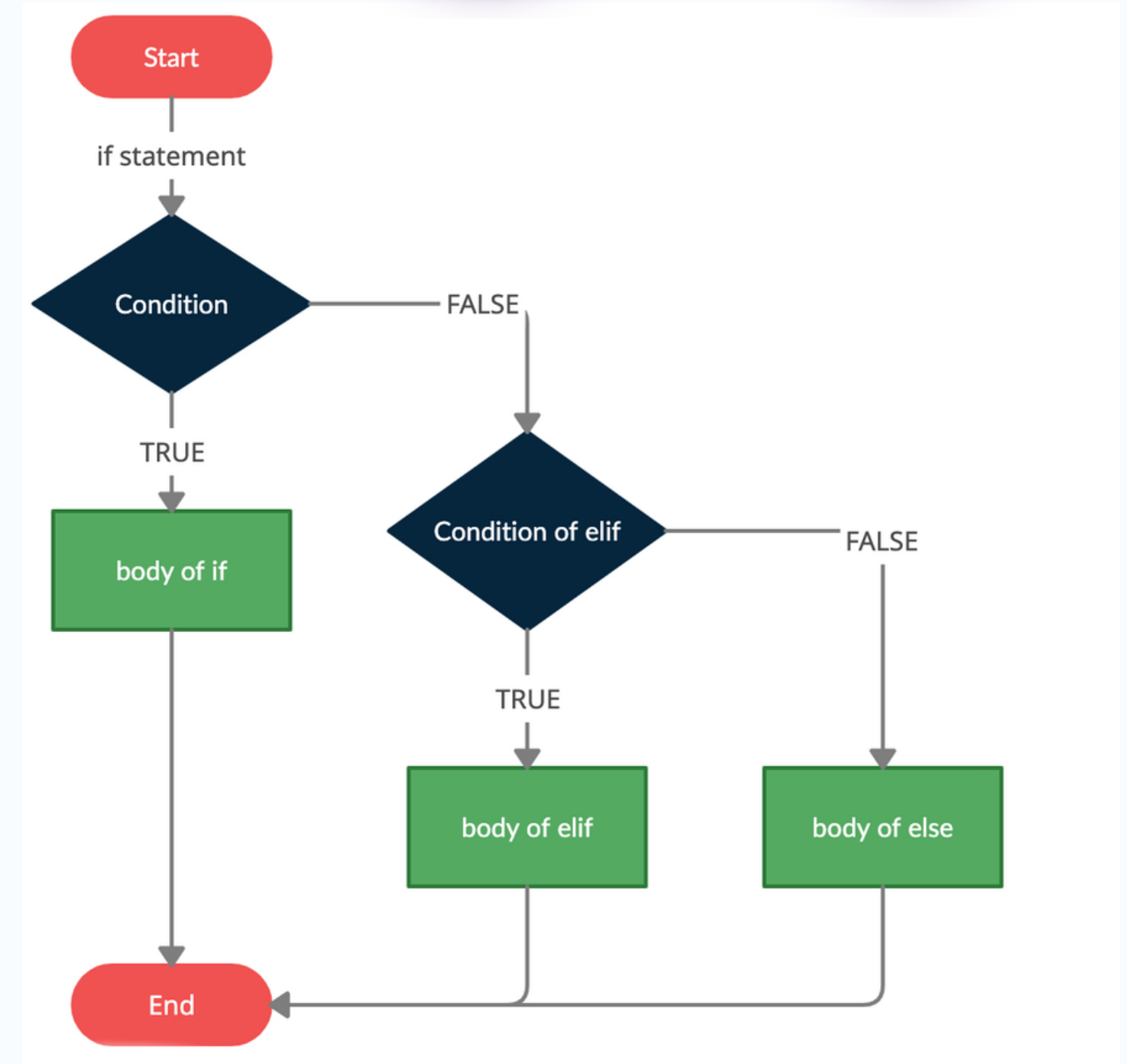
```
    print("b is greater than a")
```

```
elif a == b:
```

```
    print("a and b are equal")
```

```
else:
```

```
    print("a is greater than b")
```



# • Nested if



```
x = 25
```

```
if x > 10:
```

```
    print("Above ten,")
```

```
    if x > 20:
```

```
        print("and also above 20!")
```

```
else:
```

```
    print("but not above 20.")
```

- pass statement

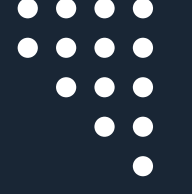


```
a = 30
```

```
b = 100
```

```
if b > a:
```

```
    pass
```



# THANKS FOR WATCHING

