23 April 2025 08:46

Day - 1

Python

1) Python 3.10 or above ←

IDE

S/w → 1) Pycharm

Dev

1) Python 3.10 or above
2) Pycharm

Notebook Jupyter ← Data Sc. AIML

Data Sc. AIML

1) Python 3.10 or above
2) Anaconda Navigator (mini Conda)
3) Jupyter Notebook

i

.ipynb

[← List

(← Tuple

{ ← Dictionary

//

#

rateOfInterest

firstName → Camel Case

first_name → Snake_Case

Data Type (in advance) X

int x = 10 (Int)
float x = 10.5 (float)
string x = "Hello" (Str)

x = 100
type(x)
Int

x = 10.25
type(x)
float

x = "Amit"
type(x)
str

Same Type → x = 10
y = "John"
Str (500)

x + y ✗

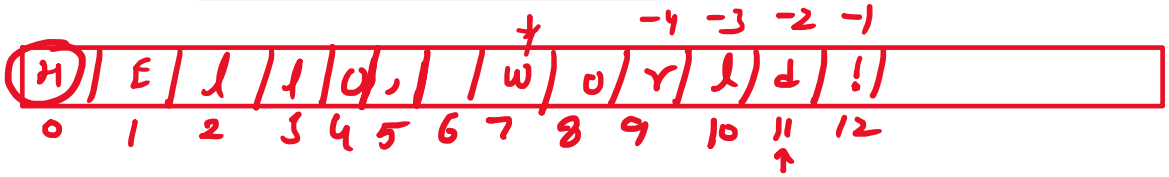
"20060" + "500"
↓
Concatenated
↓
2000050 0
20500

print(f"The number is {loan_amt}")

f-string

```
greeting = "Hello, world!"
```

greeting[13]



greeting[0] greeting[-2]

[7]

[12] !

List [ ] values [2, 4, 6, 8, 15, 20] ← mutable

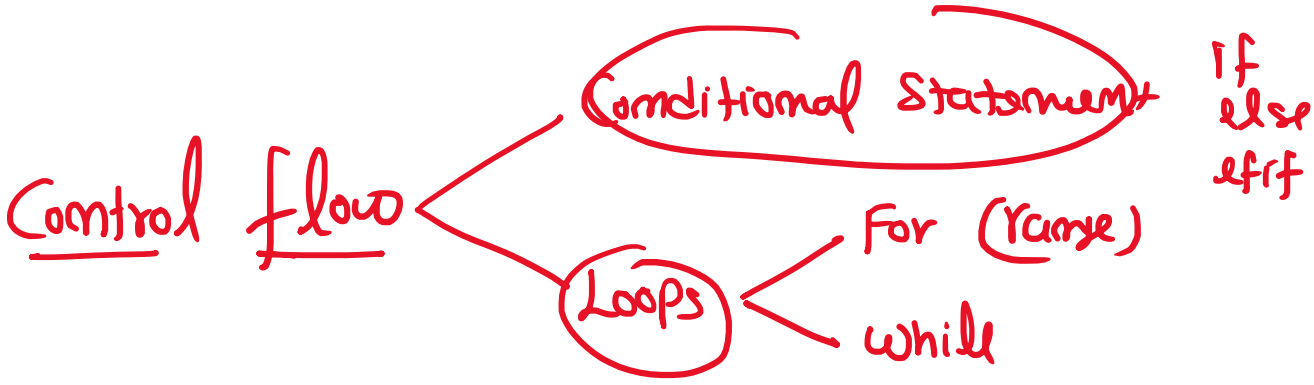
Tuple ( ) values (2, 4, 6, 8, 15, 20) ← Immutable

Dictionary { } K:V, K:V

emp = { "name": "Arun", "age": 23, "sal": 1000 }

K1 V1 K2 V2 K3 V3

emp["sal"]



C, C++, Java

if (x>10){

}

void sum(int a, int b)

{

}

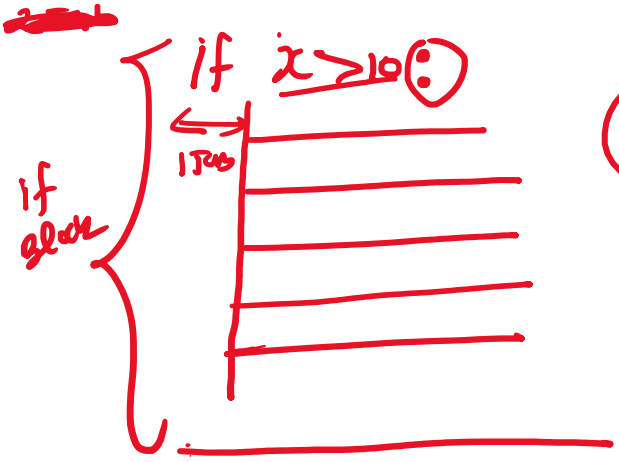
Sample

func^n Block

Samp Code Block

func" Block

1 Tab → 4 spaces



Identation

```
If x>10:
    print("Hello")
print("Demo1")
print("Demo2")
```

x=15
x=8

IF-else

```
If x>10:
    print("Hello")
else:
    print("Not Hello")
print("Demo1")
print("Demo2")
```

If

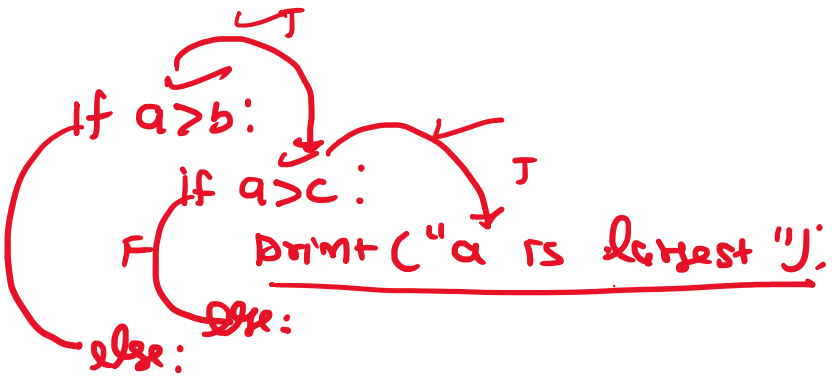
elif

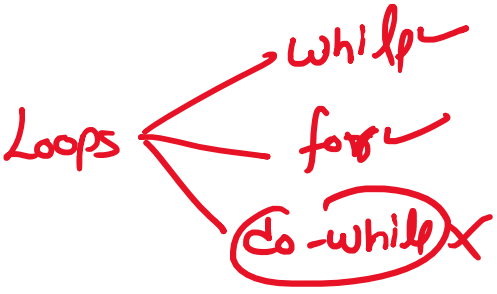
```
If x>10:
    print("x is greater than 10")
elif x>5:
    print("x is greater than5")
else
    print("x is neither gt 10 Nor 5")
```

x=15
x=8
x=4

Nested if

a
b
c






```
for (i=0; i<10; i++)
{
    print(i);
}
```

```
for i in range(0, 10):
    print(i)
```

range(start, stop, step)
start = 0, stop = n, step = 1

range(10)

range(3, 10)

range(3, 10, 2)

0 1 2 3 ... 9

4 5 6 ... 9

range(3, 10, 2)

start = 3, stop = 10, step = 2

3
5
7
9

List, Tuples, Dict, Sets

→ l1 = [2, 4, 6, 8, 15, 20, 25] ← iterables

```
for i in l1:
    print(i)
```

2
4
6
8
.
.
25

Continue
&
Break

```
for i in range(0,10):
    if i==5:
        continue
    print(i)
```

0
1
2
3
4
6
7
8
9

```
for i in range(0,10):
    if i==5:
        break;
    print(i)
```

0
1
2
3
4

while

Initialization: i=0 | Cond: i<10 | Code block: print(i) | Inc/Decr: i++ i--

0

```
x=0
while x<10:
    print(x)
    x=x+1
```

0
1
2
.
.
9

What is the significance of func^n

- Reusability
- Predefined

```
def f1():
    {
    =
    =
    =
    }
```

3 [= f1()
= f1()
= f1()
]

```
def mysum(a, b):
    s = a+b
    return s
```

mysum(2,4)

→ Print(mysum(10,20))

mysum()
mysum(10)
mysum(10,20)
mysum(10,20,30)

```
def mysum(a, b=0, c=30):
    s = a+b+c
    print(f"sum is {s}")
```

~~mysum()~~ # 600 10+200+300
mysum(10) # 510
mysum(10,25) #335 10+25+300
mysum(10,25,35) #70 335
10+25+35
=70

*args Variable Arguments

```
def mysum(*args):
    tot=0
    for val in args:
        tot = tot + val
    print(f"Sum is {tot}")
```

mysum(2, 4, 6, 8, 10, 50)

mysum(10)

(10, 20)

**kwargs

```
def empinfo(id, name):
    print(f"id is {id} name is {name}")
    id is Hitesh, name is 101

empinfo(101, "Hitesh")
=> empinfo("Hitesh", 101)        Positional
empinfo(name="Hitesh", id=101)
```

```
def empinfo(id, name):
    print(  )
```

**kwargs

```
def empinfo(**kwargs)
```

```
=> empinfo(101, "John")
=> empinfo(102, "Alice", 34)
=> empinfo(103, "Tom", 35, 7000)
```