Ch-1 Python Revision Tour-I

Variables and Data Types

What is a Variable?

A **variable** is a name that refers to a value stored in memory. In Python, variables are created when you assign a value to them.

```
x = 10
name = "Sita"
pi = 3.14
```

Rules for Naming Variables:

- Must begin with a letter or underscore (_)
- Can contain letters, digits, and underscores
- Case-sensitive (Age and age are different)
- Cannot be a Python keyword (e.g., if, for)

Data Types in Python:

Data Type	Example	Description
int	x = 5	Integer numbers
float	pi = 3.14	Decimal numbers
str	name = "Sita"	Sequence of characters
bool	is_valid = True	Boolean values (True/False)
list	[1, 2, 3]	Mutable ordered sequence
tuple	(1, 2, 3)	Immutable ordered sequence
dict	{"a": 1}	Key-value pairs
None	x = None	Null type or empty value

Operators in Python

Arithmetic Operators

Operator	Description	Example
+	Addition	5 + 2 = 7
-	Subtraction	5 - 2 = 3
*	Multiplication	5 * 2 = 10
/	Division	5 / 2 = 2.5
//	Floor Division	5 // 2 = 2
%	Modulus (Remainder)	5 % 2 = 1
**	Exponentiation	2 ** 3 = 8

Relational (Comparison) Operators

Operator	Description	Example
==	Equal to	5 == 5 → True
!=	Not equal to	5 != 3 → True
>	Greater than	5 > 3 → True
<	Less than	5 < 3 → False
>=	Greater or equal	5 >= 5 → True
<=	Less or equal	5 <= 3 → False

Logical Operators

Operator	Description	Example
and	True if both true	True and False → False
or	True if any true	True or False → True
not	Reverses the result	not True → False

Assignment Operators

Operator	Example	Meaning
=	x = 5	Assign value
+=	x += 2	x = x + 2
-=	x -= 1	x = x - 1
*=	x *= 3	x = x * 3
/=	x /= 2	x = x / 2

Membership Operators

```
'a' in 'apple'      # True
'z' not in 'zebra'  # False
```

Identity Operators

```
x = [1, 2]
y = x
print(x is y)       # True (same object)
print(x is not y)   # False
```

Conditional Statements

if Statement: Executes a block only if the condition is true.

```
age = 18
if age >= 18:
    print("Eligible to vote")
```

if-else Statement: Two paths: one if the condition is true, another if false.

```
num = 5
if num % 2 == 0:
    print("Even")
else:
    print("Odd")
```

if-elif-else Statement

Multiple conditions can be checked in order.

```
marks = 85
if marks >= 90:
    print("Grade A")
elif marks >= 75:
    print("Grade B")
elif marks >= 60:
    print("Grade C")
else:
    print("Grade D")
```

Loops in Python

for Loop

Used when the number of iterations is known.

```
for i in range(1, 6):
    print(i)
```

while Loop

Used when the number of iterations is not known in advance.

```
count = 1
while count <= 5:
    print(count)
    count += 1
```

Nested Loops

One loop inside another.

```
for i in range(1, 4):
    for j in range(1, 4):
        print(i, j)
```