TYPES OF OPERATORS

DATA TYPES

In Python, data types define the kind of values that can be stored and manipulated. Here are the main data types in Python with examples:

. Numeric Types

- **int**: Represents integers.

```
Python
x = 5
y = -10
```

- ***float: Represents floating-point numbers (decimals).***

python

Copy code

a = 3.14

b = -0.001

- ***complex: Represents complex numbers.***

python

Copy code

c = 2 + 3j

2. Sequence Types

- ***str: Represents strings (text).***

python

Copy code

name = "Alice"

greeting = 'Hello, World!'

- ***list: An ordered, mutable collection of items.***

python

Copy code

fruits = ["apple", "banana", "cherry"]

- ***tuple: An ordered, immutable collection of items.***

python

TYPES OF OPERATORS

Copy code

coordinates = (10, 20)

3. Mapping Type

- dict: A collection of key-value pairs.

python

Copy code

person = {"name": "Alice", "age": 30}

4. Set Types

- set: An unordered collection of unique items.

python

Copy code

unique_numbers = {1, 2, 3, 3} # {1, 2, 3}

- frozenset: An immutable version of a set.

python

Copy code

immutable_set = frozenset([1, 2, 3])

5. Boolean Type

- bool: Represents True or False.

python

Copy code

is_active = True

is_admin = False

6. None Type

- NoneType: Represents a null value.

python

Copy code

result = None

Example Code

Here’s a small example demonstrating some of these data types:

TYPES OF OPERATORS

```
python

Copy code

# Numeric types

age = 25

height = 5.9

complex_number = 1 + 2j


# String

name = "Bob"


# List

hobbies = ["reading", "cycling", "hiking"]


# Tuple

coordinates = (10.0, 20.0)


# Dictionary

user = {"username": "bob123", "logged_in": True}


# Set

unique_items = {1, 2, 3, 4, 4}


# Boolean

is_logged_in = False


# None

data = None


print(name, age, height, hobbies, user)
```

TYPES OF OPERATORS

Operators in Python are special symbols used to perform operations on variables and values. Here are the main types of operators in Python, along with examples:

1. Arithmetic Operators

These are used to perform mathematical operations.

- **Addition (+)**

```
python
Copy code
result = 5 + 3  # 8
```

- **Subtraction (-)**

```
python
Copy code
result = 5 - 3  # 2
```

- **Multiplication (*)**

```
python
Copy code
result = 5 * 3  # 15
```

- **Division (/)**

```
python
Copy code
result = 5 / 2  # 2.5
```

- **Floor Division (//)**

```
python
Copy code
result = 5 // 2  # 2
```

- **Modulus (%)**

```
python
Copy code
result = 5 % 2  # 1 (remainder)
```

- **Exponentiation (**)**

```
python
Copy code
result = 2 ** 3  # 8 (2 raised to the power of 3)
```

2. Comparison Operators

These are used to compare values.

TYPES OF OPERATORS

- **Equal to (==)**

```
python
Copy code
result = (5 == 5)  # True
```

- **Not equal to (!=)**

```
python
Copy code
result = (5 != 3)  # True
```

- **Greater than (>)**

```
python
Copy code
result = (5 > 3)  # True
```

- **Less than (<)**

```
python
Copy code
result = (5 < 3)  # False
```

- **Greater than or equal to (>=)**

```
python
Copy code
result = (5 >= 5)  # True
```

- **Less than or equal to (<=)**

```
python
Copy code
result = (5 <= 3)  # False
```

3. Logical Operators

These are used to combine conditional statements.

- **AND (`and`)**

```
python
Copy code
result = (5 > 3 and 8 > 5)  # True
```

- **OR (`or`)**

```
python
Copy code
result = (5 > 3 or 8 < 5)  # True
```

- **NOT (`not`)**

TYPES OF OPERATORS

```
python
Copy code
result = not(5 > 3)  # False
```

4. Assignment Operators

These are used to assign values to variables.

- **Assignment (=)**

```
python
Copy code
x = 5
```

- **Add and assign (+=)**

```
python
Copy code
x += 3  # x = x + 3
```

- **Subtract and assign (-=)**

```
python
Copy code
x -= 2  # x = x - 2
```

- **Multiply and assign (*=)**

```
python
Copy code
x *= 2  # x = x * 2
```

- **Divide and assign (/=)**

```
python
Copy code
x /= 2  # x = x / 2
```

5. Bitwise Operators

These are used to perform bit-level operations.

- **AND (&)**

```
python
Copy code
result = 5 & 3  # 1 (binary: 0101 & 0011 = 0001)
```

- **OR (|)**

```
python
Copy code
result = 5 | 3  # 7 (binary: 0101 | 0011 = 0111)
```

TYPES OF OPERATORS

- **XOR (^)**

```
python
Copy code
result = 5 ^ 3  # 6 (binary: 0101 ^ 0011 = 0110)
```

- **NOT (~)**

```
python
Copy code
result = ~5  # -6 (inverts the bits)
```

- **Left Shift (<<)**

```
python
Copy code
result = 5 << 1  # 10 (binary: 0101 becomes 1010)
```

- **Right Shift (>>)**

```
python
Copy code
result = 5 >> 1  # 2 (binary: 0101 becomes 0010)
```

6. Identity Operators

These are used to compare the memory locations of two objects.

- **is**

```
python
Copy code
a = [1, 2, 3]
b = a
result = (a is b)  # True
```

- **is not**

```
python
Copy code
c = [1, 2, 3]
result = (a is not c)  # True
```

7. Membership Operators

These are used to test for membership in a sequence.

- **in**

```
python
Copy code
fruits = ["apple", "banana", "cherry"]
result = ("banana" in fruits)  # True
```

TYPES OF OPERATORS

- **not in**

```
python
Copy code
result = ("orange" not in fruits)  # True
```

Example Code

Here's a small code snippet that demonstrates some of these operators:

```
python
Copy code
# Arithmetic Operators
a = 10
b = 3
print(a + b)  # Addition
print(a - b)  # Subtraction
print(a * b)  # Multiplication
print(a / b)  # Division

# Comparison Operators
print(a == b)  # Equal to
print(a > b)   # Greater than

# Logical Operators
print(a > b and b > 0)  # AND
print(a > b or b < 0)   # OR

# Assignment Operators
x = 5
x += 2  # x becomes 7
print(x)

# Membership Operators
my_list = [1, 2, 3]
print(2 in my_list)
```