

```
print("Hello", "World", sep=" ", end="\n")
```

Operators in Python

Types of Operators

1. Arithmetic Operators:

1. + (Addition), - (Subtraction), * (Multiplication), / (Division), % (Modulus), ** (Exponentiation), // (Floor Division).

2. Example:

```
print(10 + 5) # Output: 15  
print(10 ** 2) # Output: 100
```

2. Comparison Operators:

1. == (Equal), != (Not Equal), > (Greater Than), < (Less Than), >= (Greater Than or Equal), <= (Less Than or Equal).

2. Example:

```
print(10 > 5) # Output: True
```

at these one by one what is division what is floor

1. Arithmetic Operators:

1. + (Addition), - (Subtraction), * (Multiplication), / (Division), % (Modulus), ** (Exponentiation), // (Floor Division).

2. Example:

```
print(10 + 5) # Output: 15  
print(10 ** 2) # Output: 100
```

2. Comparison Operators:

1. == (Equal), != (Not Equal), > (Greater Than), < (Less Than), >= (Greater Than or Equal), <= (Less Than or Equal).

2. Example:

```
print(10 > 5) # Output: True  
print(10 == 5) # Output: False
```

3. Logical Operators:

1. and, or, not.

2. Example:

```
print(10 > 5) # Output: True
print(10 == 5) # Output: False
```

3. Logical Operators:

1. and, or, not.

2. Example:

```
print(True and False) # Output: False
print(True or False) # Output: True
print(not True) # Output: False
```

4. Assignment Operators:

1. =, +=, -=, *=, /=, %=, **=, //=.

2. Example:

```
x = 10
x += 5 # Equivalent to x = x + 5
print(x) # Output: 15
```

5. Membership Operators:

1. in, not in.



4. Assignment Operators:

1. `=`, `+=`, `-=`, `*=`, `/=`, `%=`, `**=`, `//=`.

2. Example:

```
x = 10
x += 5 # Equivalent to x = x + 5
print(x) # Output: 15
```

5. Membership Operators:

1. `in`, `not in`.

2. Example:

```
fruits = ["apple", "banana", "cherry"]
print("banana" in fruits) # Output: True
```

6. Identity Operators:

1. `is`, `is not`.

2. Example:

```
x = 10
y = 10
print(x is y) # Output: True
```

Summary

- Variables store data, and Python supports multiple data types.
- Typecasting allows you to convert between data types.
- Use `input()` to take user input and `print()` to display output.



6. Identity Operators:

1. is, is not.

2. Example:

```
x = 10  
y = 10  
print(x is y) # Output: True
```

Summary

- Variables store data, and Python supports multiple data types.
- Typecasting allows you to convert between data types.
- Use `input()` to take user input and `print()` to display output.
- Comments and escape sequences help make your code more readable.
- Python provides a variety of operators for performing operations on data.



Search

