

Assignment Operators:

Assignment operators assign values to variables.

Operator	Example	Description
=	x = 7	Assigns value 7 to x
+=	x += 3	x = x + 3
-=	x -= 2	x = x - 2
*=	x *= 4	x = x * 4
/=	x /= 5	x = x / 5
//=	x //= 2	x = x // 2 (integer division)
%=	x %= 3	x = x % 3 (modulus)
**=	x **= 2	x = x ** 2 (exponentiation)

Identity Operators:

Identity operators compare memory locations:

Operator	Description	Example (assume a = b)
is	True if same object	a is b
is not	True if different object	a is not b

Membership Operators

Membership operators test membership in a sequence (like lists, strings):

Operator	Description	Example
in	True if found in sequence	'a' in 'apple'
not in	True if not found	2 not in

Conditional Statements

if Statement

Syntax:if condition:

body of if statement

Example:

```
number = 5
```

```
if number > 0:
```

```
    print("Positive number")
```

if.else Statement

Syntax:

```
if condition:
```

```
    # body of if statement
```

```
else:
```

```
    # body of else statement
```

Example:

```
python
```

PYHTON

```
number = 0
```

```
if number > 0:
```

```
    print("Positive number")
```

else:

```
    print("Not a positive number")
```

if...elif...else Ladder

Syntax:

```
if condition1:
```

```
    # body of if
```

```
elif condition2:
```

```
    # body of elif
```

```
else:
```

```
    # body of else
```

Example:

```
python
```

```
number = 0
```

```
if number > 0:
```

```
    print("Positive")
```

```
elif number == 0:
```

```
    print("Zero")
```

```
else:
```

```
    print("Negative")
```

Simplified Syntax:

Python allows single-line conditional expressions:

```
print("Positive") if number > 0 else print("Not positive")
```