Boolean Values and Expressions

Python has a special type, bool, with two values: True and False (case-sensitive).

They are not strings — True ≠ "True".

Comparison operators:

== (equal), != (not equal), >, <, >=, <=.

Logical Operators

Combine boolean expressions with:

and → true only if both are true.

or → true if at least one is true.

not → reverses truth value.

Logical opposites (e.g., < is the opposite of >=).

Operator Precedence

Order of evaluation (highest → lowest):

1.** (exponent)

2.*, /, //, % (multiplication/division)

3.+, - (addition/subtraction)

4.Relational (==, !=, <, >, <=, >=)

5.not

6.and

7. or

Parentheses can be used for clarity, but Python respects these rules automatically.

Conditional Execution: Binary Selection

If/else lets the program choose between two paths.

```
if x % 2 == 0:
    print("even")
else:
    print("odd")
```

Unary Selection (if without else)

If the condition is True, the block runs.

```
if x < 0:
    print("Negative number not allowed")
print("This always runs")
```

Nested Conditionals

You can put an if inside another.

Example:

```
if x < y:
    print("x < y")
else:
```

```
    if x > y:
        print("x > y")
    else:
        print("x == y")
```

Chained Conditionals (elif)

A cleaner alternative to nested if:

```
if x < y:
    print("x < y")
elif x > y:
    print("x > y")
else:
    print("x == y")
```