

EXPRESSIONS, STATEMENT, TYPE CONVERSION, AND
INPUT / OUTPUT IN PYTHON

Date : _____

Page No. : _____

① Expressions

- An expression is a combination of values, variables, operators and function calls that evaluates to a single value.
- Example:** $3 + 5$, $(a * b) / 2$, $x + y * 3$

Types of Expressions

- Arithmetic Expression:** uses arithmetic operators
 $10 + 5 * 2$
- Relational Expression:** uses relational operators
 $a > b$
- Logical Expression:** uses logical operators
 $(a > b) \text{ and } (c < d)$
- Mixed Expression:** combination of above
 $(a + b) * (c - d) / e$

Remember!

Every expression is evaluated to a single value.

② Statement

- A statement is an instruction that Python interpreter can execute.
- Every statement may or may not produce an output.
- Example statements:**

```
x = 10      # Assignment statement
print(x)    # Output statement
y = x + 5    # Assignment statement
print(y)    # Output statement
```

Types of Statements in Python

- Assignment statement → $x = 10$
- Input statement → `input()`, `int()`, `float()`
- Output statement → `print()`
- Conditional statement → `if`, `elif`, `else`
- Looping statement → `for`, `while`
- Jump statement → `break`, `continue`, `pass`

③ Precedence of Operators

- Precedence determines the order in which operators are evaluated in an expression.
- Higher precedence operators are evaluated first.

Precedence	Operator(s)	Description	Example	Evaluates as
High ↓ Low	()	Parentheses	$(2 + 3) * 4$	$5 * 4 = 20$
	**	Exponentiation	$2 ** 3 ** 2$	$2 ** (3 ** 2) = 2 ** 9 = 512$
	+x -x ~x	Unary plus, Unary minus, Bitwise NOT	-5, +10	-5, +10
	* / // %	Multiplication, Division, Floor Division, Modulus	$10 / 3 * 2$	$(10 / 3) * 2 = 3.333 * 2 = 6.666$
	+ -	Addition and Subtraction	$10 - 3 + 2$	$(10 - 3) + 2 = 7 + 2 = 9$
	< <= > >=	Relational operators	$a > b$	Depends on values
	== !=	Equality operators	$a == b$	True or False
	not	Logical NOT	<code>not (a > b)</code>	True or False
	and	Logical AND	$(a > b) \text{ and } (b > c)$	True or False
	or	Logical OR	$(a > b) \text{ or } (b > c)$	True or False
	= += -= *= /= etc.	Assignment operators	$x = 10$	Assigns value to variable

Note:

- Parentheses can change the natural precedence.
- Operators with same precedence are evaluated from Left to Right (except **).

④ Expression and Evaluation of an Expression

- An expression is evaluated (computed) according to the precedence of operators.
- The final result is a single value.

Example 1:

```
expr = 10 + 2 * 3
      ↳ First 2 * 3
      = 10 + 6
      ↳ Add 10 + 6
      = 16
```

Example 2:

```
expr = (10 + 2) * 3 ** 2
      = 12 * 3 ** 2
      ↳ First (10 + 2)
      = 12 * 9
      ↳ Then 3 ** 2
      = 108
```

⑤ Type Conversion

- Converting a value from one data type to another data type.

(A) Implicit Conversion (Automatic)

- Done automatically by Python interpreter.
- No data loss (safe conversion).

Example:

```
a = 10      # int
b = 2.5     # float
c = a + b   # int + float
print(c)    # 12.5 (float)
```

(B) Explicit Conversion (Manual)

- Done manually using type casting functions.
- Possible data loss.
- Common Functions:
`int()`, `float()`, `str()`, `bool()`

Example:

```
a = "25"
b = int(a)    # str → int
print(b + 5)  # 30
```

⑥ Input from Console

- Python provides `input()` function to accept data from the user.
- By default, `input()` takes string as input.
- Use type conversion to convert to required type.

Examples:

```
name = input("Enter your name: ")
print("Hello,", name)

age = int(input("Enter your age: "))
print("Next year you will be", age + 1)
```

⑦ Output to Console

- Python uses `print()` function to display output.
- Multiple values can be printed by separating with commas.

Examples:

```
a = 10
b = 20
print("Sum is:", a + b)          # Output: Sum is: 30
print(a, b, sep=", ")           # Output: 10, 20
print("Value of a =", a)        # Output: Value of a = 10
print("Python", "is", "awesome!", sep=" --- ")
                                # Output: Python --- is --- awesome!
```

Exam Tip

Learn operator precedence well. It helps in solving expressions quickly in exams!

