

Class 11 · Computer Science (083) · Unit 2 · NCERT Ch 5 · CBSE 2026-27

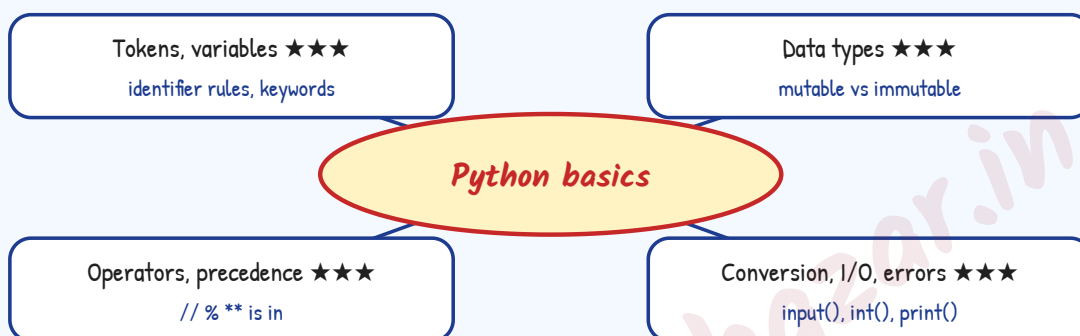
Chapter 5

Getting Started with Python

```
>>> print("Hello")
Hello
>>> 7 // 2
3
```

Features of Python · interactive and script mode · character set · tokens: keywords, identifiers, literals, operators, punctuators · variables, l-value and r-value · comments · data types · mutable and immutable · operators · precedence · expressions and statements · type conversion · input and output · errors: syntax, logical, run-time

Chapter map (★ = most important for exams)



Syllabus and weightage

The base of Unit 2 (45 marks): tokens, data types, operators, expressions, type conversion, input/output and errors. Output-prediction questions on precedence and type conversion are asked every year (also in the Class 12 board, whose questions on these topics are solved here).

★ **Exam tip:** Every output in this chapter was produced by running the code in Python 3.

5.1 Introduction to Python ★★

Python is a high-level, general-purpose language created by **Guido van Rossum** (released 1991). It uses an **interpreter**, which executes the program line by line.

- **Features:** free and open source; simple, English-like syntax; high-level; interpreted; **case-sensitive**; portable (platform independent); rich library; uses **indentation** for blocks; supports procedural and object-oriented programming; used in web, data science, AI.

Execution modes

Interactive mode	Script mode
type a statement at the <code>>>></code> prompt; result shown immediately	write the program in a file (<code>.py</code>), save, run (F5 in IDLE)
quick testing; statements are not saved	programs are saved and can be reused/edited

```
print("Hello World")      # first program
```

Output

```
Hello World
```

5.2 Character set and tokens ★★★

Character set: letters A-Z, a-z; digits 0-9; special symbols (+ - * / () [] { } : ' " , . # etc.); white spaces (space, tab, newline); all Unicode characters.

Tokens are the smallest individual units of a program:

Token	Examples
Keywords: reserved words with special meaning	False, None, True, and, as, break, class, continue, def, del, elif, else, for, from, global, if, import, in, is, lambda, not, or, pass, return, while, with...
Identifiers: names of variables, functions	marks, total_1, _count
Literals: fixed values	25, 3.14, 2+3j, 'hi', True, None
Operators	+ - * / // % ** == and in is
Punctuators (delimiters)	() [] { } , ; . ' " # @

Rules for identifiers ★★★

- Begin with a letter (A-Z, a-z) or underscore (`_`); then letters, digits, underscores.
- Cannot start with a digit; no spaces or special symbols (!, @, #, \$, %, ., -).
- Cannot be a keyword; case-sensitive (Marks $\neq$ marks); any length.

NCERT Q1: invalid: Serial_no. (dot), 1st_Room (starts with digit), Hundred\$ (\$) , Total Marks (space), total-Marks (hyphen), True (keyword). Valid: Total_Marks , _Percentage .

5.3 Variables ★★

A **variable** is a name that refers to a value (object) in memory. It is created when a value is assigned: `age = 16` . Python is dynamically typed: the same variable can later refer to a value of another type.

l-value and r-value: in $x = y + 5$, the left side (x , where the value is stored) is the **l-value**; the right side ($y + 5$, the value computed) is the **r-value**. $5 = x$ is an error because a literal cannot be an l-value.

```
length, breadth = 10, 20          # NCERT Q2
sum = (length + breadth) / 2
stationery = ['Paper', 'Gel Pen', 'Eraser']
first, middle, last = 'Mohandas', 'Karamchand', 'Gandhi'
fullname = first + ' ' + middle + ' ' + last
print(sum, fullname)
```

Output

15.0 Mohandas Karamchand Gandhi

Comments: start with `#`; ignored by the interpreter; explain the code. Multi-line notes are often written as triple-quoted strings (`""" ... """`).

Every value is an object with an identity (`id()`), type (`type()`) and value.

5.4 Data types ★★★

Category	Type	Example
Number	int	-5, 0, 74
	float	74.0, 3.14, 2.5e3
	complex	3+4j
Boolean	bool	True, False (subtype of int: True = 1)
Sequence	str	'Python', "A"
	list	[1, 'a', 2.5]
	tuple	(1, 2, 3)
None	NoneType	None (absence of value)
Mapping	dict	{'roll': 1, 'name': 'Asha'}

Mutable vs immutable ★★★

Immutable (cannot change in place)	Mutable (can change in place)
int, float, complex, bool, str, tuple	list, dict (also set)

```
s = "cat"; s[0] = "b"          # TypeError: str is immutable
L = [1, 2]; L[0] = 9           # fine: L becomes [9, 2]
x = 5; print(id(x)); x = x + 1; print(id(x)) # new object, new id
```

NCERT Q6 data types: months in a year → int; Delhi resident or not → bool; mobile number → str (no arithmetic, may start with +91/0); pocket money → float; volume of a sphere → float; perimeter of a square → float (int if sides are whole); student's name → str; address → str.

5.5 Operators ★★★

Type	Operators (example → result)
Arithmetic	+ - * ; / true division ($7/2 \rightarrow 3.5$); // floor division ($7//2 \rightarrow 3$; $-7//2 \rightarrow -4$); % remainder ($7\%2 \rightarrow 1$); ** power ($2**3 \rightarrow 8$)
Relational	== != > < >= <= (give True/False); strings compared by Unicode values ('Bye' == 'BYE' → False)

Logical	and, or, not
Assignment	=
Augmented	+= -= *= /= //= %= **=
Identity	is, is not: same object (memory location)?
Membership	in, not in: 'a' in 'cat' → True

Precedence (highest to lowest) ★★★

1. ****** (right to left: $2**3**2 = 2**9 = 512$)
2. unary **+**x **-**x **~**x
3. ***** **/** **//** **%** (left to right)
4. **+** **-**
5. relational **==** **!=** **>** **<** **>=** **<=**, **is**, **in**
6. **not**
7. **and**
8. **or**

Brackets override precedence. **and** / **or** return one of their operands: $3 > 1$ and 2 gives 2; 0 or 5 gives 5.

```
print(14 % 3 ** 2 * 4)          # 14 % 9 * 4
print(2 + 9 * ((3*12) - 8) / 10)
print(24 // 4 // 2, 10 + 6 * 2 ** 2 != 9//4 - 3 and 29 >= 29/9)
```

Output

```
20
27.2
3 True
```

5.6 Expressions, statements, type conversion ★★★

An **expression** is a combination of values, variables and operators that is evaluated to one value ($3 + 4*2$). A **statement** is an instruction the interpreter executes ($x = 5$, `print(x)`, `if ...`).

Implicit conversion (coercion)	Explicit conversion (type casting)
done automatically by Python, to the wider type: $10 + 20.5 \rightarrow 30.5$ (int $\rightarrow$ float)	done by the programmer with <code>int()</code> , <code>float()</code> , <code>str()</code> , <code>bool()</code> ...

```
print(float(10), int(7.9), int('25') + 5, str(25) + '5')
print(int('3.14'))
```

Output

```
10.0 7 30 255
ValueError: invalid literal for int() with base 10: '3.14'
```

`int()` truncates a float towards zero; a string must look like an integer for `int()`. Use `int(float('3.14'))` $\rightarrow 3$.

5.7 Input and output ★★★

`input(prompt)` always returns a **string**; convert it for arithmetic. `print(*values, sep=' ', end='\n')` prints values separated by `sep` and ends with `end`.

```
name = input("Enter name: ")
age = int(input("Enter age: "))
print("Hello", name, sep=", ", end="! ")
print("You will be 100 in", 2026 + (100 - age))
```

Output

```
Enter name: Riya
Enter age: 16
Hello, Riya! You will be 100 in 2110
```

5.8 Errors ★★★

Error	Meaning	Example
Syntax error	violates the rules of the language; program does not run	<code>print("Hi", if x>5 print(x)</code> (missing colon)
Logical error	runs but gives a wrong result (bug in logic); hardest to find	<code>average = a + b / 2</code> (needs brackets)
Run-time error (exception)	occurs during execution and stops the program	division by zero (<code>ZeroDivisionError</code>), <code>int('abc')</code> (<code>ValueError</code>), undefined name (<code>NameError</code>), <code>"15" + 3</code> (<code>TypeError</code>)

NCERT Q8: (a) `25 / 0` and (b) `num1 = 25; num2 = 0; num1/num2` are both **run-time errors** (`ZeroDivisionError`).

Worked programs (NCERT Q10–Q19)

```
# Q10 Celsius to Fahrenheit
c = float(input("Celsius: "))
print("Fahrenheit:", c * 9/5 + 32)          # 100 → 212.0, 0 → 32.0

# Q11 simple interest
P = float(input("P: ")); R = float(input("R: ")); T = float(input("T: "))
SI = P * R * T / 100
print("Amount payable =", P + SI)

# Q12 days for A, B, C together
x, y, z = 3, 4, 6
print(x*y*z / (x*y + y*z + x*z))          # 72/54 = 1.333...

# Q15 swap without a third variable
a, b = 5, 9
a, b = b, a                                # or: a = a + b; b = a - b; a = a - b
print(a, b)                                # 9 5

# Q16 repeat a string
n = int(input("n: ")); print("GOOD MORNING " * n)

# Q18 volume of spheres
for r in (7, 12, 16):
    print(r, round(4/3 * 3.14159 * r**3, 2))
```

Output

```
7 1436.75
12 7238.22
16 17157.27
```

NCERT exercises, solved ★★★

Q3. Logical expressions.

Ans.

```
(a) 20 + (-10) < 12           # True
(b) not (num3 > 24)           # or num3 <= 24
(c) num1 < 6.75 < num2       # (or num2 < 6.75 < num1)
(d) first < middle < last    # string comparison
(e) len(stationery) == 0     # or stationery == []
```

Q4. Add parentheses so each evaluates to True.

Ans. (a) $0 == (1 == 2) \rightarrow 0 == \text{False} \rightarrow \text{True}$. (b) $2 + (3 == 4) + 5 == 7 \rightarrow 2 + 0 + 5 = 7 \rightarrow \text{True}$. (c) $(1 < -1) == (3 > 4) \rightarrow \text{False} == \text{False} \rightarrow \text{True}$.

Q5. Outputs.

Ans. (a) 25 (num2 was computed when num1 was 4). (b) 64 (right side evaluated first: 6, 2+2). (c) **NameError**: num3 is used on the right before it is defined.

Q9. Dart hits the board of radius 10 if $x**2 + y**2 \leq 100$.

Ans. (0,0) True; (10,10) 200 $\rightarrow$ False; (6,6) 72 $\rightarrow$ True; (7,8) 113 $\rightarrow$ False.

Q7. Outputs with num1 = 4, num2 = 3, num3 = 2.

Ans. (a) $\text{num1} += 3 + 2 \rightarrow 9$
 (b) $4 ** 5 \rightarrow 1024$
 (c) $\text{num1} ** 5 \rightarrow \text{num1}$ becomes 1024 (nothing printed)
 (d) $'5' + '5' \rightarrow 55$ (string concatenation)
 (e) $4.00/4.0 \rightarrow 1.0$
 (f) $2 + 9*28/10 = 2 + 25.2 \rightarrow 27.2$
 (g) $(24//4)//2 = 6//2 \rightarrow 3$
 (h) 10.0
 (i) **ValueError**: '3.14' is not an integer string
 (j) **False** (case differs)
 (k) True and True $\rightarrow$ True
 (l) $10 + 6*4 = 34$; $9//4 - 3 = -1$; $34 \neq -1$ is True; $29 \geq 3.22$ is True $\rightarrow$ True
 (m) $5 + 10 = 15 < 50$ True and $29 \leq 29$ True $\rightarrow$ True
 (n) $(0 < 6)$ is True, so or short-circuits $\rightarrow$ True

Important exam questions

Differentiate between / and //.

Ans. / gives the true quotient as a float ($7/2 = 3.5$); // gives the floor of the quotient ($7//2 = 3$, $-7//2 = -4$).

What is the difference between `is` and `==`?

Ans. `==` compares values; `is` checks whether both names refer to the same object. `a = [1]; b = [1]: a == b` is True but `a is b` is False.

CBSE board questions on these topics, 2024-2026 (fully solved) ★★★

These Class 12 board questions test Class 11 Python basics.

CBSE 2026 State True or False: In Python, data type of `74` is same as the data type of `74.0`.

Ans. False. `type(74)` is `int`; `type(74.0)` is `float`.

CBSE 2025 What will be the output of `print(14%3**2*4)`? (A) 16 (B) 64 (C) 20 (D) 256

Ans. (C) 20. `**` has the highest precedence: `3**2 = 9`. Then `%` and `*` from left to right: `14 % 9 = 5`; `5 * 4 = 20`.

CBSE 2024 Output of `print(16*5/4*2/5-8)`: (a) -3.33 (b) 6.0 (c) 0.0 (d) -13.33

Ans. (c) 0.0. Left to right: `16*5 = 80`; `80/4 = 20.0`; `20.0*2 = 40.0`; `40.0/5 = 8.0`; `8.0 - 8 = 0.0` (float, because `/` gives a float).

CBSE 2025 Which is the correct identifier? (A) `global` (B) `Break` (C) `def` (D) `with`

Ans. (B) Break. `global`, `def` and `with` are keywords. Python is case sensitive, so `Break` (capital B) is not the keyword `break`.

CBSE 2025 Identify the invalid statement: (A) `print("A",10,end="*")` (B) `print("A",sep="*",10)` (C) `print("A",10,sep="*")` (D) `print("A"*10)`

Ans. (B). A positional argument (10) cannot follow a keyword argument (`sep="*"`); this is a `SyntaxError`.

CBSE 2025 Which operator evaluates to True if the variables on either side point to the same memory location? (A) `is` (B) `is not` (C) `and` (D) `or`

Ans. (A) is, the identity operator (compares object identity, i.e. `id()`).

CBSE 2024 Which statement will raise an error? (a) `print("A"*3)` (b) `print(5*3)` (c) `print("15" + 3)` (d) `print("15" + "13")`

Ans. (c). A str cannot be added to an int: `TypeError`. (a) gives AAA, (b) 15, (d) 1513.

CBSE 2026 Which expression evaluates to True? (A) `2>3 and 2<3` (B) `3>1 and 2` (C) `3>1 and 3>2` (D) `3>1 and 3<2`

Ans. (C). True and True $\rightarrow$ True. (A) and (D) give False; (B) gives 2 (and returns its last operand), which is truthy but not the value True, so the scheme also accepted (B).

CBSE 2025 Write one example each of (i) Syntax Error (ii) Implicit type conversion.

Ans. (i) `print(2 ; 5)` or `if x>5 print(x)` (missing colon): the program does not run.

(ii) `x = 10; y = 20.5; print(x + y)` $\rightarrow$ 30.5: the int 10 is converted to float automatically.

Practice set

- Output: `print(2**3**2, -7//2, -7%3)`
- Output: `print(type(10/2), 10//3.0)`
- Output: `print('5'*3, 5*3)`
- Output: `print(0 or 'x', 5 and 0)`
- Which are mutable: str, list, tuple, dict?
- Write a program to find the average of three numbers.

Answers

- Ans.**
1. 512 -4 2
 2. `<class 'float'> 3.0`
 3. 555 15
 4. x 0
 5. list, dict.
 6. `a, b, c = map(float, input().split()); print((a + b + c) / 3)`

Multiple choice: 20 MCQs

- Python was created by:
 - (a) James Gosling (b) Dennis Ritchie (c) Guido van Rossum (d) Bjarne Stroustrup
- Python program files have extension:
 - (a) .py (b) .pt (c) .p (d) .pyt
- Prompt of interactive mode:
 - (a) \$ (b) # (c) :: (d) >>>
- Valid identifier:
 - (a) 2nd (b) _total (c) for (d) my-var
- A keyword:
 - (a) none (b) None (c) Main (d) value
- `7 // 2` gives:
 - (a) 3.5 (b) 1 (c) 4 (d) 3
- `7 % 3` gives:
 - (a) 1 (b) 2 (c) 2.33 (d) 0
- `2 ** 3 ** 2` gives:
 - (a) 64 (b) 36 (c) 512 (d) 18
- Immutable type:
 - (a) set (b) list (c) dict (d) tuple
- Mutable type:
 - (a) str (b) list (c) int (d) tuple
- `input()` returns a:
 - (a) int (b) float (c) str (d) bool
- Explicit conversion example:
 - (a) `int('25')` (b) `True + 1` (c) `'a' * 2` (d) `3 + 4.5`
- Identity operator:
 - (a) in (b) is (c) == (d) and
- Membership operator:
 - (a) is (b) = (c) in (d) not
- Highest precedence:
 - (a) * (b) + (c) and (d) **
- Error when dividing by zero at run time:
 - (a) ZeroDivisionError (b) NameError (c) logical (d) SyntaxError
- Program runs but gives a wrong answer:
 - (a) logical error (b) syntax error (c) run-time error (d) no error
- Default value of print's end parameter:
 - (a) space (b) tab (c) comma (d) '\n' newline
- `type(3+4j)`:
 - (a) float (b) complex (c) int (d) str

20. In $x = y + 1$, x is the:

(a) r-value (b) literal (c) l-value (d) operator

1 (c) · 2 (a) · 3 (d) · 4 (b) · 5 (b) · 6 (d) · 7 (a) · 8 (c) · 9 (d) · 10 (b) · 11 (c) · 12 (a) · 13 (b) · 14 (c) · 15 (d) · 16 (a) · 17 (a) · 18 (d) · 19 (b) · 20 (c)

Assertion-Reason (6)

(a) Both A and R true, R explains A. (b) Both true, R does not explain A. (c) A true, R false. (d) A false, R true.

1. A: `int(input())` is needed for adding two typed numbers. R: `input()` returns a string.

2. A: `'Bye' == 'BYE'` is True. R: Python is case-sensitive.

3. A: `s[0] = 'b'` gives an error for a string `s`. R: Strings are immutable.

4. A: `10 + 20.5` gives 30.5. R: Python converts int to float implicitly.

5. A: `1st_name` is a valid identifier. R: Identifiers can contain digits.

6. A: `**` is evaluated right to left. R: `2**3**2 = 64`.

Key: 1 (a) · 2 (d) · 3 (a) · 4 (a) · 5 (d) · 6 (c)

- **Last-night summary:** Python: interpreted, case-sensitive, indentation; interactive (`>>>`) vs script (.py). Tokens: keywords, identifiers (letter/_ first, no symbols/spaces, not keyword), literals, operators, punctuators. l-value = left (storage), r-value = right (value).
- Types: int, float, complex, bool, str, list, tuple, None, dict; immutable: numbers, bool, str, tuple; mutable: list, dict. Operators: / vs //, %, **, relational, and/or/not, augmented, is/is not, in/not in; precedence `**` > unary > `* / // %` > `+ -` > relational > not > and > or.
- Implicit vs explicit conversion (`int()`, `float()`, `str()`); `input()` returns str; `print(sep, end)`. Errors: syntax (rules), logical (wrong output), run-time (`ZeroDivisionError`, `ValueError`, `TypeError`, `NameError`).