BROADWAY INFOSYS
Since 2008

CERTIFIED ISO 9001:2015 COMPANY

Building Global IT Professionals since 2008

15K+
Certified Students

152+
Courses

70+
Qualified Teachers

www.broadwayinfosys.com

Contents

1.2 Python Basics

- Algorithms
- Data Types & Variables (String, Integer, Float, Complex, Boolean, None)
- Input and Output Functions
- Working with the format() method, f-strings, & escape sequences
- Basic Arithmetic & Operators
- Type casting, type checking, & validation

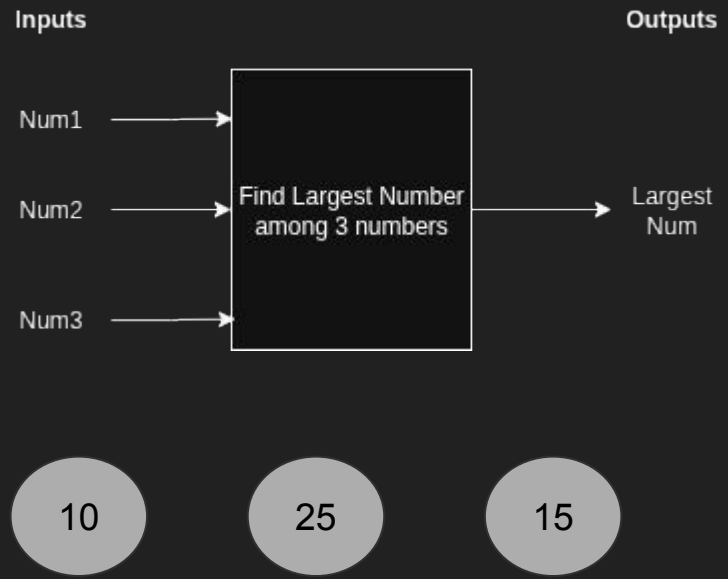
https://broadwayinfosys.com/python/python-programming

Algorithms: The Heart of Programming

- An algorithm is a step-by-step solution to a problem.
- An algorithm is like a recipe: a set of steps to solve a problem.
- Every algorithm has:
  a. Input – What you start with
  b. Steps – What you do
  c. Output – The result
- Example: Algorithm for Making Milk Tea
  a. Take a teapot, put it on the stove
  b. Light the stove
  c. Put milk in the teapot
  d. Add tea leaves to a teapot
  e. Stir for 3–5 minutes
  f. Add sugar to the teapot
  g. Stir for 1-2 minutes
  h. Turn off the stove
  i. Serve tea it on a tea cup
- Key Concepts:
  a. Input: Milk, tea leaves, sugar, etc.
  b. Process: Boiling, stirring, mixing
  c. Output: A cup of tea
- **Takeaway: Algorithms are not just code—they're structured problem-solving steps.**

Algorithms: The Heart of Programming

- Problem: Find the largest of three numbers
- **Black box view:**



- Think about how you will solve the problem.
- **Algorithm, surface:**
  a. Compare Num1 and Num2
  b. Keep the largest
  c. Compare largest to Num3
  d. Print the largest
- **Algorithm, in detail:**
  a. Compare Num1 and Num2
  b. If Num1 > Num2, then maxNum = Num1
     - Otherwise, maxNum = Num2
  c. Compare maxNum and Num3
  d. If Num3 > maxNum, then maxNum = Num3
     - Otherwise, do nothing
  e. Print maxNum

Algorithms: Converting to Code

- Problem: Find the largest of three numbers

python Copy Edit

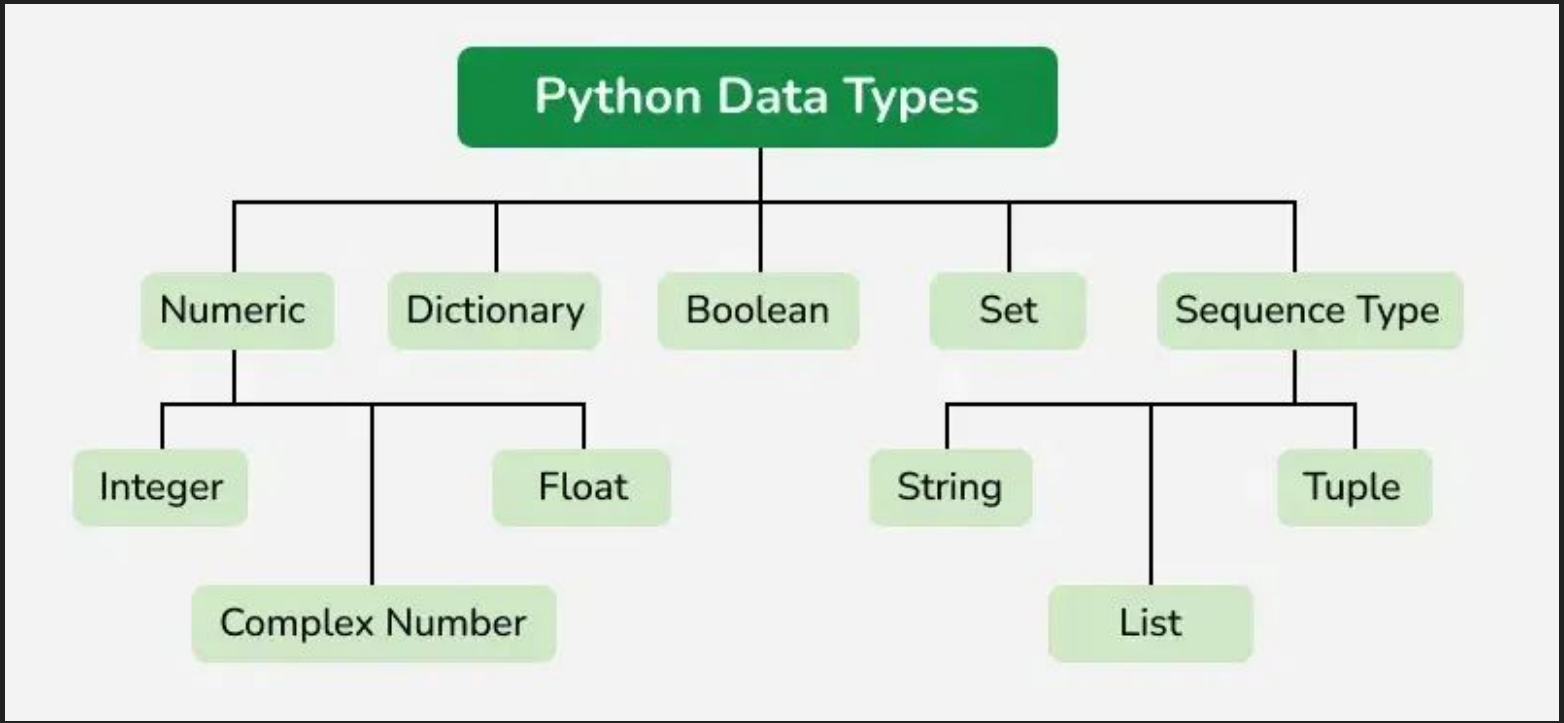
```python
a = 10
b = 25
c = 15

if a > b:
    max_num = a
else:
    max_num = b

if c > max_num:
    max_num = c

print("The largest number is:", max_num)
```

Data Types


Python Data Types
Numeric
Dictionary
Boolean
Set
Sequence Type
Integer
Float
Complex Number
String
Tuple
List

Data Types

Type	Example	Description
int	`10, -5`	Whole numbers
float	`3.14, -0.5`	Decimal numbers
str	`"hello", '5alphabet'`	Text (always in quotes)
bool	`True, False`	Logical values (Yes/No)
list	`[1, 2, 3, “apple”]`	Ordered collection (can mix types)
dict	`{"name": "Ali", “age”: 25}`	Key-value pairs

Variables

- Variables are used to store data in a program.
- Think of them as containers that hold information you want to use later.
- Their values can change during the program.
- Variables hold different types of data in them.

python Copy Edit

```python
name = "Alice"      # str
age = 25            # int
pi = 3.14159        # float
z = 2 + 3j          # complex
is_valid = True     # bool
nothing = None      # NoneType
```

Input and Output Functions

- Functions to read data from user or file. Example: input()
- Functions to write data to console or file. Example: print()

python Copy Edit

```python
name = input("Enter your name: ")
```

python Copy Edit

```python
print("Hello,", name)
```

String Formatting Methods

- format() method:

```python
print("Hello, {}. You are {} years old.".format("Alice", 25))
```

- f-strings (Python 3.6+):

```python
name = "Bob"
age = 30
print(f"Hello, {name}. You are {age} years old.")
```

String Formatting Methods

- Escape Sequences:
  - \n – Newline
  - \t – Tab
  - \\ – Backslash
  - \" or \' – Quote

python Copy Edit

```python
print("Name:\tJohn\nAge:\t25\nLocation:\tKathmandu")
```

```
Name:	John
Age:	25
Location:	Kathmandu
```

Arithmetic Operators

- Operators are symbols that perform operations on variables and values.

Operator	Description	Example	Result
+	Addition	5 + 3	8
-	Subtraction	5 - 2	3
*	Multiplication	4 * 2	8
/	Division	10 / 2	5.0
//	Floor Division	7 // 2	3
%	Modulus (remainder)	7 % 2	1
**	Exponentiation	2 ** 3	8

Arithmetic Operators

python

Copy Edit

```python
a = 10
b = 3

print("a + b =", a + b)     # 13
print("a - b =", a - b)     # 7
print("a * b =", a * b)     # 30
print("a / b =", a / b)     # 3.333...
print("a // b =", a // b)   # 3
print("a % b =", a % b)     # 1
print("a ** b =", a ** b)   # 1000
```

Comparison (Relational) Operators

Operator	Description	Example	Result
==	Equal to	5 == 5	True
!=	Not equal to	5 != 3	True
>	Greater than	5 > 3	True
<	Less than	5 < 3	False
>=	Greater than or equal	5 >= 5	True
<=	Less than or equal	5 <= 6	True

Comparison (Relational) Operators

python

Copy Edit

```python
x = 5
y = 10

print("Equal:", x == y)         # False
print("Not Equal:", x != y)     # True
print("Greater Than:", x > y)   # False
print("Less Than:", x < y)      # True
print("Greater or Equal:", x >= y) # False
print("Less or Equal:", x <= y)    # True
```

Logical Operators

Operator	Description	Example	Result
`and`	Both conditions	`True and False`	`False`
`or`	Either condition	`True or False`	`True`
`not`	Inverts the result	`not True`	`False`

Logical Operators

- Truth table:

X	Y	X and Y	X or Y	not(X)	not(Y)
T	T	T	T	F	F
T	F	F	T	F	T
F	T	F	T	T	F
F	F	F	F	T	T

Logical Operators

python Copy Edit

```python
a = True
b = False

print("a and b:", a and b)  # False
print("a or b:", a or b)    # True
print("not a:", not a)      # False
```

Assignment Operators

Operator	Description	Example	Equivalent
=	Assign value	x = 5	
+=	Add and assign	x += 2	x = x + 2
-=	Subtract and assign	x -= 1	x = x - 1
*=	Multiply and assign	x *= 3	x = x * 3
/=	Divide and assign	x /= 2	x = x / 2
%=	Modulo and assign	x %= 3	x = x % 3
**=	Exponent and assign	x **= 2	x = x ** 2

Assignment Operators

python

Copy Edit

```python
x = 5
x += 2   # x = x + 2
print("x after += 2:", x)   # 7

x -= 1   # x = x - 1
print("x after -= 1:", x)   # 6

x *= 3   # x = x * 3
print("x after *= 3:", x)   # 18

x /= 2   # x = x / 2
print("x after /= 2:", x)   # 9.0

x %= 4   # x = x % 4
print("x after %= 4:", x)   # 1.0

x **= 3  # x = x ** 3
print("x after **= 3:", x)  # 1.0
```

Membership Operators

Operator	Description	Example	Result
`in`	Value exists in group	`'a' in 'apple'`	`True`
`not in`	Value doesn't exist	`'z' not in 'apple'`	`True`

Membership Operators

python Copy Edit

```python
name = "Pujan"

print("'j' in name:", 'j' in name)         # True
print("'z' not in name:", 'z' not in name) # True
```

Identity Operators

Operator	Description	Example	Result
`is`	Same object	`x is y`	`True/False`
`is not`	Not same object	`x is not y`	`True/False`

Identity Operators

python Copy Edit

```python
a = [1, 2, 3]
b = a
c = [1, 2, 3]

print("a is b:", a is b)       # True (same object)
print("a is c:", a is c)       # False (different objects with same value)
print("a is not c:", a is not c) # True
```

Type Casting & Checking

- One type can be cast to another using <type>():

```python
x = int("10")      # from str to int
y = float(5)       # from int to float
z = str(25.5)      # from float to str
```

- To check type of object, use type(object) or isinstance(object):

```python
type(42)           # <class 'int'>
isinstance(42, int)   # True
```

Type Validation

- We will learn if..else statement in next chapter

```python
user_input = input("Enter your age: ")

if user_input.isdigit():
    age = int(user_input)
    print(f"You are {age} years old.")
else:
    print("Invalid input. Please enter a number.")
```

Hands-on

Q/A

Assignments

1. Write an algorithm to make an omelette.
2. Ask the user for their name and age, then print a greeting using an f-string.
3. Take two numbers as input, cast them, and print their sum, product, and quotient.
4. Try formatting output with escape sequences like \n and \t.
5. Write a Python program to check if a person is eligible to vote. For example: If a person is at or over 18 years of age, he/she is eligible to vote.
6. Search whether there are other ways of formatting a string.

THANK YOU!

BROADWAY INFOSYS
Since 2008

Shree Ganesh Marg, Subidhanagar, Tinkune, Kathmandu
01-4117578 / 4111849 / 984100200
www.broadwayinfosys.com