Shortcut	Explanation
print("Hello")	Prints a message to the console.
x = input("Enter: ")	Takes user input as a string.
x = int(input("Enter: "))	Takes user input as an integer.
len(lst)	Returns the length of a list or string.
range(5)	Generates numbers from 0 to 4.
x, y = y, x	Swaps values of x and y.
import math	Imports the math module.
math.sqrt(16)	Returns square root of 16.
math.pow(2, 3)	Raises 2 to the power of 3.
sum([1, 2, 3])	Returns sum of list elements.
max(1, 5, 3)	Returns the largest number.
min(1, 5, 3)	Returns the smallest number.
round(3.14159, 2)	Rounds a number to 2 decimal places.
random.randint(1, 10)	Generates a random integer between 1 and 10.
random.choice(['a', 'b', 'c'])	Returns a random item from a list.
Hello'.lower()	Converts string to lowercase.
Hello'.upper()	Converts string to uppercase.
hello'.capitalize()	Capitalizes the first letter.
hello'.startswith('h')	Checks if string starts with 'h'.
hello'.endswith('o')	Checks if string ends with 'o'.
hello'.replace('h', 'y')	Replaces 'h' with 'y'.
hello world'.split()	Splits string into a list by spaces.
,'.join(['a', 'b', 'c'])	Joins list elements with commas.
hello'.count('l')	Counts occurrences of 'l'.
hello'.find('e')	Returns index of 'e'.
lst = [1, 2, 3]	Creates a list.
lst.append(4)	Adds 4 to the end of the list.
lst.insert(1, 99)	Inserts 99 at index 1.
lst.pop()	Removes and returns the last item.
lst.remove(2)	Removes first occurrence of 2.
lst.reverse()	Reverses the list.
lst.sort()	Sorts the list in ascending order.
lst.sort(reverse=True)	Sorts the list in descending order.
list(set(lst))	Removes duplicates from a list.
[i**2 for i in range(5)]	List comprehension to create squares.
sum(lst) / len(lst)	Computes the average of a list.
x in lst	Checks if x is in list.
{1, 2, 3} & {2, 3, 4}	Finds the intersection of two sets.
`{1, 2, 3}	{2, 3, 4}`
{1, 2, 3} - {2, 3}	Finds difference of two sets.
{x: x**2 for x in range(3)}	Dictionary comprehension.
dict.keys()	Returns all keys in a dictionary.
dict.values()	Returns all values in a dictionary.
dict.items()	Returns key-value pairs.
dict.get('key', 'default')	Returns value or default if key not found.

dict.update({'a': 1})	Updates dictionary with new key-value pair.
del dict['key']	Deletes a key from dictionary.
try-except	Handles exceptions.
try: x=1/0 except: print("Error")	Catches division by zero error.
with open('file.txt', 'r') as f	Opens a file safely.
f.read()	Reads entire file contents.
f.readline()	Reads one line from a file.
f.readlines()	Reads all lines into a list.
f.write("Hello")	Writes text to a file.
os.getcwd()	Gets current working directory.
os.listdir()	Lists all files in a directory.
os.rename('old.txt', 'new.txt')	Renames a file.
os.remove('file.txt')	Deletes a file.
time.sleep(2)	Pauses execution for 2 seconds.
datetime.datetime.now()	Gets current date and time.
datetime.date.today()	Gets today's date.
lambda x: x*2	Creates an anonymous function.
map(lambda x: x*2, [1, 2, 3])	Applies function to each list item.
filter(lambda x: x>2, [1, 2, 3])	Filters list based on condition.
from collections import Counter	Imports Counter for counting elements.
Counter([1, 1, 2, 2, 2])	Counts occurrences of each element.
all([True, True, False])	Returns False if any element is False.
any([True, False, False])	Returns True if any element is True.
zip([1,2], [3,4])	Combines two lists into pairs.
enumerate(['a', 'b'])	Returns index-value pairs.
isinstance(5, int)	Checks if 5 is an integer.
issubclass(bool, int)	Checks if bool is a subclass of int.
globals()	Returns all global variables.
locals()	Returns all local variables.
type(5)	Returns the type of a variable.
id(5)	Returns memory address of an object.
callable(len)	Checks if len is callable.
hash("hello")	Returns hash value of an object.
sorted([3,1,2])	Returns a sorted list.
re.match(r'\d+', '123abc')	Matches regex pattern.
re.findall(r'\d+', 'abc123xyz')	Finds all matches of regex pattern.
sys.exit()	Exits the program.
sys.argv	Gets command-line arguments.
itertools.permutations([1,2,3])	Generates all permutations.
itertools.combinations([1,2,3], 2)	Generates all combinations of 2 elements.
Shortcut	**Explanation**
itertools.cycle([1,2,3])	Cycles through an iterable indefinitely.
itertools.accumulate([1,2,3])	Computes running totals of a list.
itertools.chain([1,2], [3,4])	Combines multiple iterables into one.
itertools.combinations_with_replacement([1,2	Generates all combinations allowing repetition.
from functools import reduce	Imports reduce() function.

reduce(lambda x, y: x + y, [1, 2, 3])	Reduces a list to a single value using a function.
set([1, 2, 2, 3])	Creates a set and removes duplicates.
frozenset([1, 2, 3])	Creates an immutable set.
dict.fromkeys(['a', 'b', 'c'], 0)	Creates a dictionary with default values.
{k: v for k, v in zip(keys, values)}	Creates a dictionary from two lists.
{k: k**2 for k in range(5)}	Dictionary comprehension.
del my_dict['key']	Deletes a key-value pair from a dictionary.
copy.deepcopy(obj)	Creates a deep copy of an object.
hex(255)	Converts number to hexadecimal.
bin(255)	Converts number to binary.
oct(255)	Converts number to octal.
abs(-10)	Returns the absolute value.
divmod(10, 3)	Returns quotient and remainder (3,1).
complex(1, 2)	Creates a complex number.
str(123).zfill(5)	Pads string with zeros ("00123").
str.lstrip(), str.rstrip(), str.strip()	Removes spaces from left, right, or both.
bytes("hello", "utf-8")	Converts a string to bytes.
bytearray(5)	Creates a mutable byte array.
memoryview(b"hello")	Creates a memoryview object of bytes.
chr(65)	Converts ASCII code to character ('A').
ord('A')	Converts character to ASCII code (65).
isinstance(10, (int, float))	Checks if variable is an int or float.
callable(print)	Checks if an object is callable.
help(str)	Displays documentation for the str type.
dir(str)	Lists all attributes and methods of str.
eval('2 + 3')	Evaluates a string expression (5).
exec('print(2 + 3)')	Executes a string of Python code.
repr(123.456)	Returns a string representation ('123.456').
format(3.14159, ".2f")	Formats a number to 2 decimal places.
"hello".encode("utf-8")	Encodes a string into bytes.
b'hello'.decode("utf-8")	Decodes bytes into a string.
next(iter([1, 2, 3]))	Gets the next item from an iterator.
all([True, False, True])	Returns False if any element is False.
any([False, False, True])	Returns True if any element is True.