Polymorphism in python lest you use the same function name for different types of objects.

* This means one function or method can act in many forms.
* It makes your code flexible and easy to manage.
* It helps you write code that can work with various objects.
* You do not need to know their exact type.
* This saves time.
* It also makes your programs more adaptable.

3Ways to use polymorphism in python

* You can use polymorphism in python In a few ways:
* With Bulit-in function’s own function often use polymorphism.
* With inheritance (Method Overriding): child classes can change methods they get from their parent classes.

Polymorphism with Bulit-in Functions

Many built-in python functions work in different ways.Their action change based on the type of data you give them. The len() function is a good example.

* When you use len() on a string,it counts the characters.
* When you use len() on a list, it counts the item.
* Whwn you use len() on a dictionary,it counts the keys.

# Use len() with a string

my string = ”hello”

Print (len(my \_string))

# Use len() with a list

my list =[1,2,3,4]

Print(len(my\_list))

# Use len() with a dictionary

my dict = {“a”:1, “b”:2}

Print(len (my\_dict))