What is Polymorphism?

In this lesson, the concept of Polymorphism will be explained which is an important concept related to OOP.

Definition

> The word **Polymorphism** is a combination of two Greek words, **Poly** means *many* and **Morph** means *forms*.

In programming, *polymorphism* refers to the same object exhibiting different forms and behaviors.

For example, take the Shape Class. The exact shape you choose can be anything. It can be a rectangle, a circle, a polygon or a diamond. So, these are all shapes but their properties are different. This is called **Polymorphism**.

Shape

Rectangle   Circle   Polygon   Diamond

Many Shapes
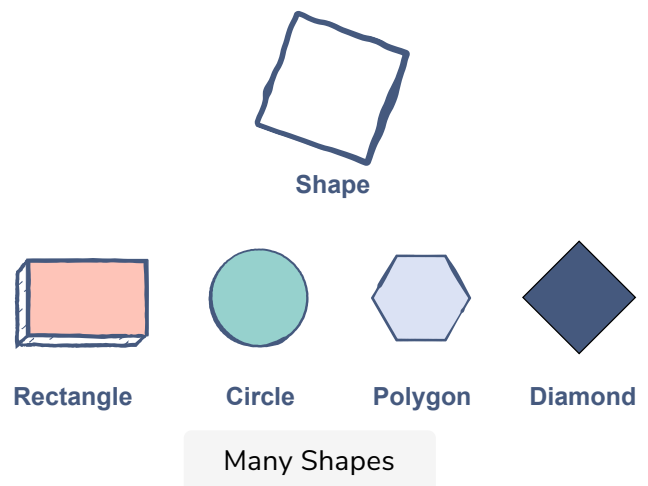
A Brief Introduction

Assume there is a base class named **Shape** from which the subclasses **Rectangle, Circle, Polygon, and Diamond** are derived. Also consider that the Shape class has a method called `calculateArea()`, which is inherited by all

subclasses mentioned. With polymorphism, each subclass may have its way of implementing the method. So, for example, when the `calculateArea()` method is called in an object of the **Rectangle class**, the method might respond by displaying the area of the rectangle. On the other hand, when the same method is called in an object of the **Circle class**, the circle's area might be calculated and displayed on the screen.

In effect, polymorphism cuts down the work of the developer because when the time comes to create more specific subclasses with certain unique attributes and behaviors, the developer can alter the code in the particular portions where the responses differ. All other pieces of the code can be left untouched.

So far, we've learnt what the polymorphism is. In the next lesson, we will learn how to achieve polymorphism in OOP.