

Polymorphism

Overview

Polymorphism is a core concept of an object-oriented paradigm that provides a way to perform a single action in different forms. It provides the ability to call the same method on different JavaScript objects. As JavaScript is not a type-safe language, we can pass any type of data member with the methods.

Polymorphism is the OOPs principle that provides the facility to perform one task in many ways.

```
class parent
{
  display () {
    console.log("parent is invoked");
  }
}
```

```
class child extends parent
{
  display() {
    console.log("child is invoked");
  }
}
```

```
var a= new A();
var b = new B();
```

```
a.display();
b.display();
```

```
OUTPUT: parent is invoked
        child is invoked
```

In the above example display() function is used multiple times for two different classes. A is parent class, and B is the child class to A.

No.	Method Overloading	Method Overriding
1)	Method overloading is used to increase the readability of the program.	Method overriding is used to provide the specific implementation of the method already provided by its super class.
2)	Method overloading is performed within the class.	Method overriding occurs in two classes that have IS-A (inheritance) relationship.
3)	In the case of method overloading, the parameters must be different.	In the case of method overriding, the parameter must be the same.
4)	Method overloading is an example of compile-time polymorphism.	Method overriding is an example of run time polymorphism.
5)	Method overloading can't be performed by changing the return type of the method only. The return type can be the same or different in method overloading. But you must have to change the parameter.	The return type must be the same or covariant in method overriding.