Answer 1

A constructor is a special function that creates and initializes an object instance of a class. In JavaScript, a constructor gets called when an object is created using the new keyword.

The purpose of a constructor is to create a new object and set values for any existing object properties.

When a constructor gets invoked in JavaScript, the following sequence of operations take place:

- A new empty object gets created.
- The this keyword begins to refer to the new object and it becomes the current instance object.
- The new object is then returned as the return value of the constructor.

Answer 2

“This” keyword refers to an object that is executing the current piece of code. It references the object that is executing the current function. If the function being referenced is a regular function, “this” references the global object. If the function that is being referenced is a method in an object, “this” references the object itself.

The JavaScript “this” keyword is one of the most widely used keywords. It can seem complex at first, but once you start using “this” keyword, everything will become clear.

Answer 3

- **bind**: It binds the function with provided value and context but it does not executes the function. To execute function you need to call the function.
- **call**: It executes the function with provided context and parameter.
- **apply**: It executes the function with provided context and parameter as array.

Call and apply are pretty interchangeable. Just decide whether it’s easier to send in an array or a comma separated list of arguments.

Call is for comma (separated list) and Apply is for Array.

Bind is a bit different. It returns a new function. Call and Apply execute the current function immediately.

Bind is great for a lot of things. We can use it to curry functions like in the above example. We can take a simple hello function and turn it into a helloJon or helloKelly. We can also use it for events like onClick where we don’t know when they’ll be fired but we know what context we want them to have.

Answer 4

JavaScript is not a class-based object-oriented language. But it still has ways of using object oriented programming

OOP is normally explained with 4 key principles that dictate how OOP programs work. These are inheritance, encapsulation, abstraction and polymorphism. Let's review each of them.

Inheritance is the ability to create classes based on other classes. With inheritance, we can define a parent class (with certain properties and methods), and then children classes that will inherit from the parent class all the properties and methods that it has

Encapsulation is another key concept in OOP, and it stands for an object's capacity to "decide" which information it exposes to "the outside" and which it doesn't. Encapsulation is implemented through public and private properties and methods.

Abstraction is a principle that says that a class should only represent information that is relevant to the problem's context. In plain English, only expose to the outside the properties and methods that you're going to use. If it's not needed, don't expose it.

Then there is polymorphism (sounds really sophisticated, doesn't it? OOP names are the coolest... ☺). Polymorphism means "many forms" and is actually a simple concept. It's the ability of one method to return different values according to certain conditions.

Answer 5

Abstraction

Abstraction is a principle that says that a class should only represent information that is relevant to the problem's context. In plain English, only expose to the outside the properties and methods that you're going to use. If it's not needed, don't expose it.

This principle is closely related to encapsulation, as we can use public and private properties/methods to decide what gets exposed and what doesn't.

Abstraction hides certain details and only show the essential features of the object. It tries to reduce and factor out details so that the developer can focus on a few concepts at a time. This approach improves understandability as well as maintainability of the code. Abstraction helps us to reduce code duplication.

Answer 6

Polymorphism is one of the core concepts of object-oriented programming languages where poly means many and morphism means transforming one form into another. Polymorphism means the same function with different signatures is called many times. In real life, for example, a boy at the same time may be a student, a class monitor, etc. So a boy can perform different operations at the same time. This is called polymorphism.

Features of Polymorphism:

- Programmers can use the same method name repeatedly.
- Polymorphism has the effect of reducing the number of functionalities that can be paired together.

Answer 7

The JavaScript inheritance is a mechanism that allows us to create new classes on the basis of already existing classes. It provides flexibility to the child class to reuse the methods and variables of a parent class.

The JavaScript extends keyword is used to create a child class on the basis of a parent class. It facilitates child class to acquire all the properties and behavior of its parent class.

Points to remember

It maintains an IS-A relationship.

The extends keyword is used in class expressions or class declarations.

Using extends keyword, we can acquire all the properties and behavior of the inbuilt object as well as custom classes.

We can also use a prototype-based approach to achieve inheritance.

Answer 8

You might already be familiar with systems like a vending machine, a bank locker, or even a purse. All of these contain some items in them. Items that are contained within a system are made available to use through the system.

For example, you can not directly access a snack from a vending machine without having to use the machine. The vending machine has a few defined procedures which allow you to access the items in it. This is exactly what is meant by Encapsulation.

In the world of programming, encapsulation is the bundling of data with the methods that operate on that data.

Advantages:

Protection of Object against Illegal Access:

Decoupling Implementation Details

Shared Mutable State

Answer 9

JavaScript classes have introduced as an easy way and a syntactical sugar to write constructor functions. They are mainly used to create new objects.

To define a class in JavaScript, we use the keyword class and we give it a name with the first letter capitalized. Then we will need to define a constructor method inside of the class.

```
class User{

  constructor(firstName, lastName){

    this.firstName = firstName;

    this.lastName = lastName;

  }



}
```

Answer 10

The super keyword is used to call the constructor of its parent class to access the parent's properties and methods.

The super.prop and super[expr] expressions are valid in any method definition in both classes and object literals. The super(...args) expression is valid in class constructors.

Syntax

```
super([arguments]) // calls the parent constructor.

super.propertyOnParent

super[expression]
```