

➤ Abstraction:

An abstraction is a way of hiding the implementation details and showing only the functionality to the users. In other words, it ignores the irrelevant details and provides access only for required functions and properties.

Embedding Abstraction in a JavaScript program can enhance the readability of the code and avoid duplication. By providing only important details to the users, it also improves the security of an application.

- Abstract classes in JavaScript

An abstract class is a class that has been declared abstract and has the option to include abstract methods. They cannot be instantiated, but they can be used as subclasses.

In JavaScript, the concept of an abstract class is not natively supported as it is in other languages. It can be written with a custom code to mimic the behavior of an abstract class.

- Advantages of Abstraction

- ✓ **Simplifies code and reduces complexity:** Abstraction presenting a simplified interface and hiding unnecessary details, abstraction reduces the overall complexity of your code. This makes it easier for developers to understand, extend, and maintain the codebase.
- ✓ **Code organization:** Abstraction enables you to organize your code in a modular and hierarchical manner. By abstracting away lower-level details, you can focus on higher-level concepts, making your code more readable and easier to manage.

- ✓ **Flexibility:** provides a level of flexibility by allowing changing the underlying implementation without affecting the code that uses the abstraction. This is particularly valuable when adapting to evolving requirements or integrating with external libraries.

➤ Interface:

An interface is a contract between itself and any class that implements it. Interface can have methods, properties, or events. It contains only declaration of its members and implementation of its members will be given by the class who implements the interface. Interface makes it easy to maintain the program. Following are the specified terms of interface.

- Interface in JavaScript

Interfaces don't exist in JavaScript because it is a dynamic language, where types are changed so often that the developer may not have even realized.

Interfaces are created using prototypes instead of classes. A prototype is basically a constructor function.

JavaScript has the reserved word `interface` in case they ever would like to implement interfaces

- Advantages of Interfaces

- ✓ **Easier Maintenance:** Interfaces facilitate easier maintenance by clearly defining the boundaries between components. When changes are required, developers can focus on updating the specific components affected by the change without worrying about unintended side effects.

- ✓ **Readability and maintainability:** Clearly defined interfaces improve code readability by providing a high-level view of an object's capabilities. This makes it easier for developers to understand the purpose and functionality of various components.
- ✓ **Loose coupling:** When components interact based on interfaces, they become loosely coupled. This means that changes to the internal implementation of one component are less likely to affect others, as long as the interface remains unchanged.

➤ **Difference between Abstraction and interfaces:**

- ✓ An interface can have only abstract methods, an abstract class can have both abstract and non-abstract methods.
- ✓ An interface can only have public members, while an abstract class can have protected and public members.
- ✓ Abstract class can provide the implementation of the interface. Interface can't provide the implementation of an abstract class

-Name: MennaTallah Ashraf Saleh.
-Track: Full stack web dev. Using PHP.
-Eng. Ahmed Almahdy.