**Explain the difference between an abstract class and an interface in PHP.**

In PHP, both abstract classes and interfaces are used to define a contract that classes must adhere to. They both facilitate the implementation of polymorphism and help define a blueprint for classes. However, there are key differences between abstract classes and interfaces:

***Abstract Class:***

An abstract class is a class that cannot be instantiated on its own. It is meant to be subclassed by other classes.

Abstract classes can contain a mix of abstract (unimplemented) and concrete (implemented) methods.

Abstract classes can also have properties with or without values.

**Inheritance:**

Subclasses (concrete classes) that extend an abstract class must provide concrete implementations for all abstract methods defined in the abstract class.

Subclasses inherit both the structure (methods and properties) and behavior (method implementations) of the abstract class.

A class can extend only one abstract class in PHP due to single inheritance limitations.

**Use Cases:**

Abstract classes are useful when you want to provide a common base class with some shared implementation that should be inherited by its subclasses.

They are suitable for situations where you have a "is-a" relationship between the base class and its subclasses (e.g., a "Vehicle" abstract class with "Car" and "Motorcycle" subclasses).

***Interface:***

An interface is a contract that defines a set of method signatures but does not provide any method implementations.

Interfaces cannot have properties, and all declared methods are implicitly abstract and public.

Classes that implement an interface must provide concrete implementations for all methods defined in the interface.

**Inheritance:**

A class can implement multiple interfaces, allowing for multiple inheritance of method contracts.

Unlike abstract classes, interfaces do not provide any method implementations; they only define method signatures.

**Use Cases:**

Interfaces are useful when you want to define a common set of methods that unrelated classes should implement. It enforces a contract that specifies what methods must be available in implementing classes.

They are suitable for situations where you have a "can-do" relationship between classes (e.g., classes that can be serialized should implement the "Serializable" interface).

In summary, the main difference between abstract classes and interfaces in PHP is that abstract classes can contain both method declarations and implementations and support single inheritance, while interfaces can only contain method declarations and support multiple inheritance. Your choice between them depends on the specific design requirements of your application. You may use abstract classes when you want to provide some common implementation, and you may use interfaces when you want to define a contract that multiple classes should adhere to.