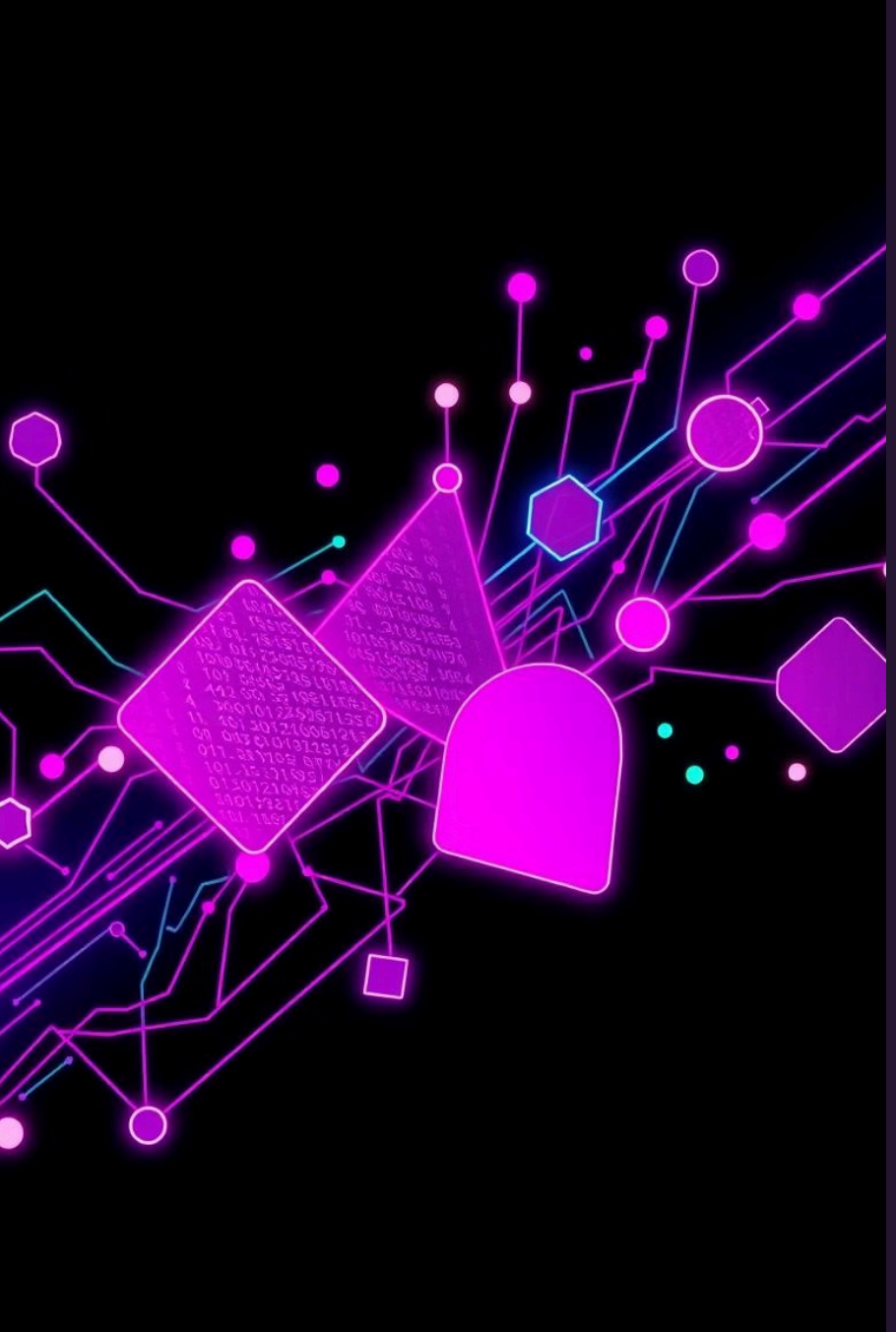


Object-Oriented Programming in PHP

This presentation delves into the world of object-oriented programming (OOP) in PHP, exploring its core concepts, advantages, and real-world applications.

What is OOP?

Code Organization

A programming paradigm that structures code around objects, representing real-world entities. It promotes modularity and reusability, making code easier to maintain and extend.

Data Encapsulation

OOP enforces data security by grouping data and methods together, preventing direct access to data from outside the object.

Core OOP Concepts

1 **Classes**

Blueprints for creating objects. They define properties (data) and methods (actions) that objects inherit.

2 **Objects**

Instances of classes, representing real-world entities with specific data values and methods.

3 **Inheritance**

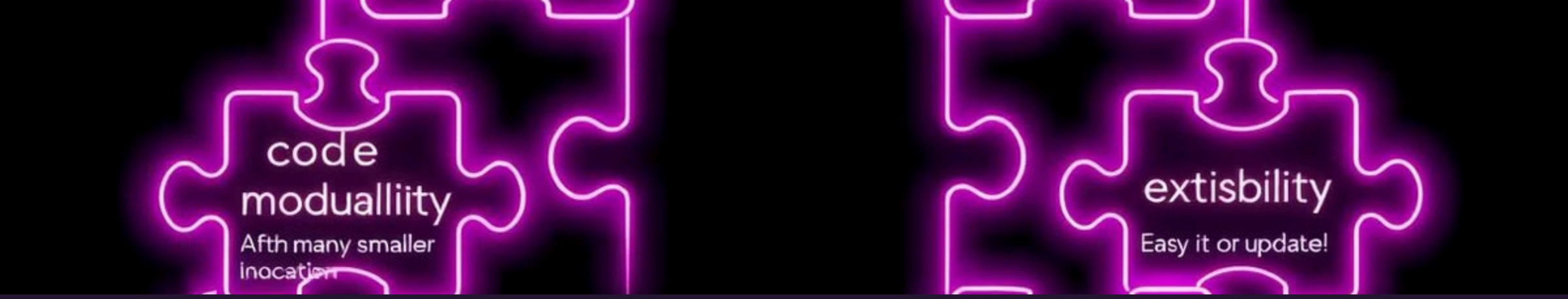
A mechanism for creating new classes (child classes) based on existing classes (parent classes), inheriting properties and methods.

4 **Encapsulation**

The practice of hiding data within an object, protecting it from external access and modifying it only through defined methods.

5 **Polymorphism**

The ability of objects of different classes to respond differently to the same method call.



Advantages of OOP

Modularity

OOP breaks down complex problems into smaller, self-contained units (objects), making code easier to understand and maintain.

Reusability

OOP promotes code reuse by inheriting properties and methods from existing classes, reducing development time and effort.

Extensibility

New features can be easily added without altering existing code, by extending existing classes or creating new ones.

Maintainability

Changes in one part of the code are less likely to affect other parts, simplifying debugging and maintenance.

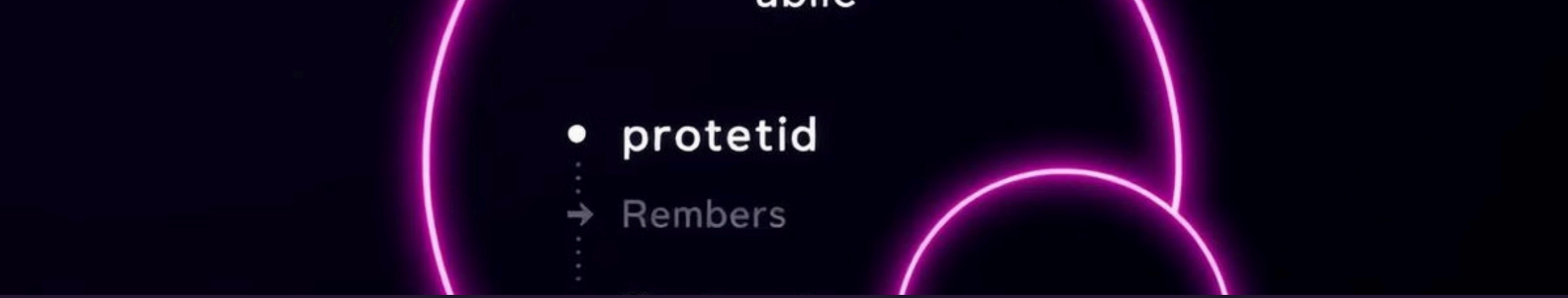
Creating Classes and Objects in PHP

Class Definition

Use the "class" keyword to define a new class with properties and methods.

Object Instantiation

Create an instance of a class using the "new" keyword, creating an object with its own unique properties and methods.



Access Modifiers

1

Public

Accessible from anywhere, both inside and outside the class.

2

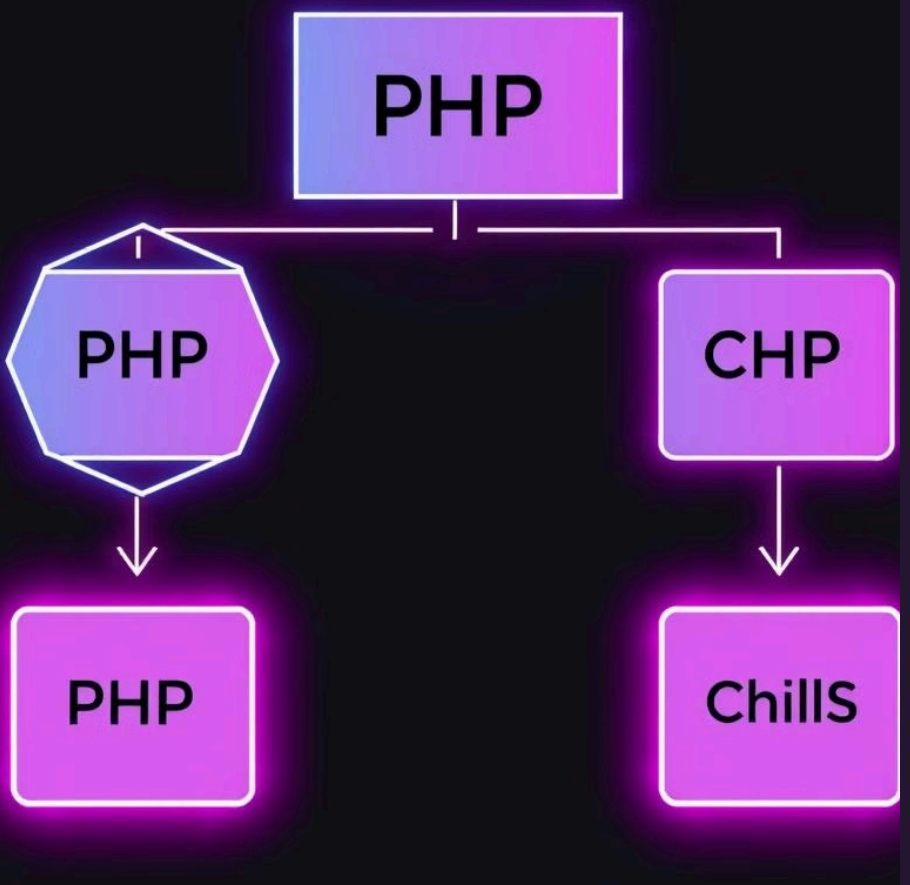
Private

Accessible only within the same class, preventing direct external access.

3

Protected

Accessible within the same class and its subclasses, enabling inheritance and controlled access.



Inheritance in PHP

1

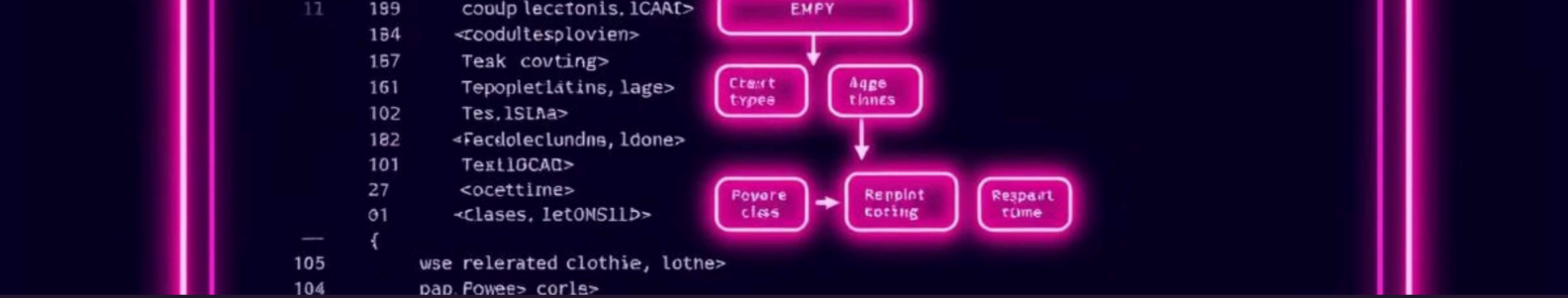
Parent Class

Defines the base properties and methods inherited by child classes.

2

Child Class

Extends the parent class, inheriting its properties and methods and potentially adding new ones.



Practical Examples and Use Cases

1

User Accounts

Representing users with properties like username, password, and roles.

2

Database Interactions

Encapsulating database operations within objects, simplifying database access.

3

E-commerce

Creating classes for products, orders, and customers, streamlining online shopping.

Thank You!

Thank you for your attention. I hope this presentation has provided a solid understanding of object-oriented programming in PHP. Feel free to ask any questions.