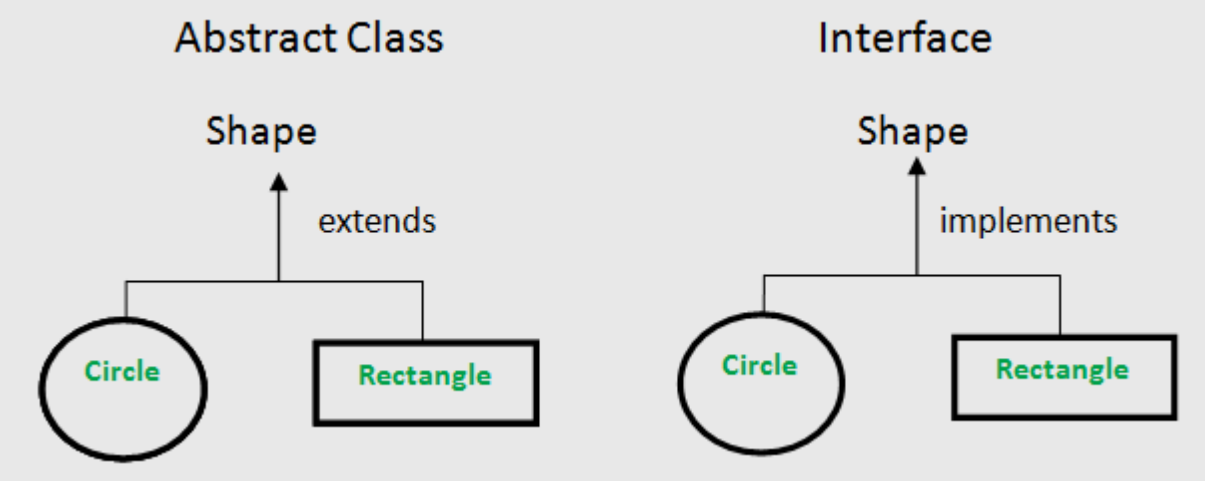
Abstract class vs Interface

في عالم الOOP في PHP، ساعات المطور بيبقى محتاج يحدد قواعد أساسية تمشي عليها الكلاسات اللي هيكتبها بعد كده.

هنا بيظهر سؤال مهم: أستخدم Abstract Class ولا Interface؟

الاتنين بيشبهوا بعض في فكرة إنهم بيحددوا شكل للكلاس، لكن كل واحد فيهم له مميزات واستخدامات مختلفة.

والفرق بينهم هو اللي بيخلي الكود إما منظم وسهل يتوسع، أو معقد وصعب يتعدل.



Abstract Class

كلاس مينفعش نعمل منه Object مباشر.

ممكن يحتوي على: Functions عادية (Concrete).

Functions مجردة (Abstract) لازم الكلاس الابن ينفذها.

الخصائص:

ممكن يحتوي على متغيرات + ثوابت.

ممكن يحتوي على Constructors.

الكلاس يقدر يرث Abstract Class واحد فقط (Single Inheritance)

مثال:

```php
// Rectangle class extending Shape
class Rectangle extends Shape {
    private $width, $height;

    public function __construct($width, $height, $name) {
        parent::__construct($name);
        $this->width = $width;
        $this->height = $height;
    }

    public function area() {
        return $this->width * $this->height;
    }
}

// Using the classes
$rect = new Rectangle(5, 10, "My Rectangle");
echo $rect->getName() . " area: " . $rect->area();

// Output =>  My Rectangle area: 50

```

Interface

الـ Interface عبارة عن عقد (Contract) بيحتوي على تعريف Functions من غير أي تنفيذ.
أي كلاس بيعمل implements للـ Interface لازم ينفذ كل Functions اللي فيه.

الخصائص:

كل Functions فيه بتكون Public بشكل افتراضي.
منفعش يكون فيه متغيرات عادية، لكن ممكن يحتوي على ثوابت فقط.
الكلاس الواحد يقدر يعمل implements لأكتر من Interface (Multiple Inheritance).

مثال:

```php
// Interface Drawable
interface Drawable {
    public function draw();
}

// Interface Movable
interface Movable {
    public function moveTo($x, $y);
}

// Circle class implementing both interfaces
class Circle implements Drawable, Movable {
    private $radius;

    public function __construct($radius) {
        $this->radius = $radius;
    }

    public function draw() {
        echo "Circle has been drawn\n";
    }

    public function moveTo($x, $y) {
        echo "Circle moved to x=$x, y=$y\n";
    }
}

// Using the Circle class
$circle = new Circle(5);
$circle->draw();
$circle->moveTo(10, 20);

/* Output =>
Circle has been drawn
Circle moved to x=10, y=20
*/

```

ايمته نستخدم كل واحد؟

استخدم Abstract Class لو:

- عندك كلاسات ليها سلوك مشترك (Common Behavior).
- محتاج تنفذ شوية دوال عادية (Concrete) بجانب الدوال المجردة.
- محتاج Constructors أو متغيرات.

استخدم Interface لو:

- محتاج تحدد Contract صريح من غير أي تفاصيل تنفيذ.
- عايز تعمل Multiple Inheritance.
- عايز تضمن إن أي كلاس يطبق الـ Interface هينفذ نفس الدوال المطلوبة.

المصادر :

https://www.php.net/manual/en/language.oop5.abstract.php

https://www.w3schools.com/php/php_oop_interfaces.asp

https://www.geeksforgeeks.org/java/difference-between-abstract-class-and-interface-in-java/