

اعتبارسنجی‌های رایج برای رمز عبور در کلاس: Validator

در یک سیستم واقعی، می‌خواهیم پسوورد:

۱. خالی نباشه
۲. حداقل طول کافی داشته باشه (مثلاً حداقل ۸ کاراکتر)
۳. گاهی اوقات شامل ترکیب حرف بزرگ، کوچک، عدد یا علامت خاص باشه (در پروژه‌های پیشرفته‌تر)

مثال: متد password در کلاس Validator

```
class Validator
{
    public static function password($value)
    {
        $value = trim($value);

        // ۱. چک کنیم خالی نباشه
        if (empty($value)) {
            return false;
        }

        // ۲. حداقل ۸ کاراکتر باشه
        if (strlen($value) < 8) {
            return false;
        }

        // ۳. اختیاری: شامل حداقل یک عدد باشه
        if (!preg_match('/[0-9]/', $value)) {
            return false;
        }

        // ۴. اختیاری: شامل حرف بزرگ باشه
        if (!preg_match('/[A-Z]/', $value)) {
            return false;
        }
    }
}
```

```

        return true; // اگر همه شرایط رعایت شد
    }
}

```

چند مثال:

مقدار	نتیجه
123	✗ کمتر از ۸ کاراکتر
password123	✗ بدون حرف بزرگ
Password123	✓ معتبر
PASSWORD123	✓ معتبر (فاصله‌ها حذف می‌شوند با trim)
abcde	✗ ضعیف

عالیه! الان می‌خواهم تابع `preg_match` رو کامل، دقیق و مرحله‌به‌مرحله برات توضیح بدم + بهت یاد بدم چطور چک کنیم رشته‌ای شامل کاراکترهایی مثل @، #، ! و غیره باشه.

preg_match چیست؟ ✓

`preg_match()` تابعی در PHP هست که بررسی می‌کند آیا یک رشته (string) با یک الگوی خاص (regular expression) مطابقت داره یا نه.

ساختار کلی:

```
preg_match("الگو", "رشته‌ای که می‌خوای بررسی کنی");
```

خروجی	معنی
1	یعنی تطابق پیدا شد ✓
0	یعنی تطابق پیدا نشد ✗
False	یعنی اروری در الگو وجود داشت ⚠

✈️ مثال‌های ساده:

مثال ۱: آیا رشته شامل عدد هست؟

خروجی: ۱ (چون عدد داره) `preg_match('/[0-9]/', 'abc123');` //

مثال ۲: آیا حرف بزرگ داره؟

خروجی: ۱ `preg_match('/[A-Z]/', 'helloWorld');` //

🌟 بررسی وجود کاراکترهای خاص مثل %, \$, !, #, @ و غیره

🎯 هدف: بررسی کنیم آیا کاربر حداقل یک کاراکتر خاص وارد کرده یا نه.

✓ الگوی مناسب:

`preg_match('/[\\@\\#\\!\\$\\%\\^\\&*\\(\\)]/', $password);`

🗨 توضیح:

- `[\\@\\#\\!\\$\\%\\^\\&*\\(\\)]` یعنی یک گروه از کاراکترها
- `\\` قبل از هر کاراکتر خاص گذاشته می‌شه تا به درستی تفسیر بشه (چون مثلاً `(` خودش در regex معنی خاصی داره)
- یعنی دنبال یکی از این کاراکترها در رشته می‌گرده `() * & % $! # @ :`

✪ نمونه کامل بررسی رمز عبور شامل کاراکتر خاص:

```
$password = 'Hello@123';  
  
if (preg_match('/[\\@\\#\\!\\$\\%\\^\\&\\*\\(\\)]/', $password))  
{  
    echo " رمز عبور شامل کاراکتر خاص است ✓";  
} else {  
    echo " رمز عبور شامل هیچ کاراکتر خاصی نیست ✗";  
}
```

□ تمرین سریع:

۱. مقدار MyPass123! رو تست کن (✓ چون ! داره)
۲. مقدار mypass123 رو تست کن (✗ چون هیچ کاراکتر خاصی نداره)