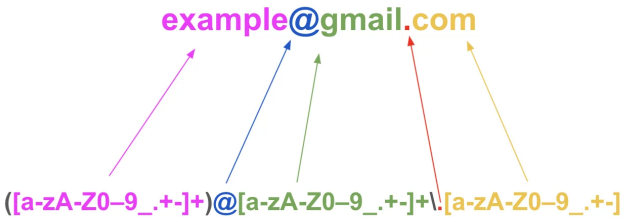
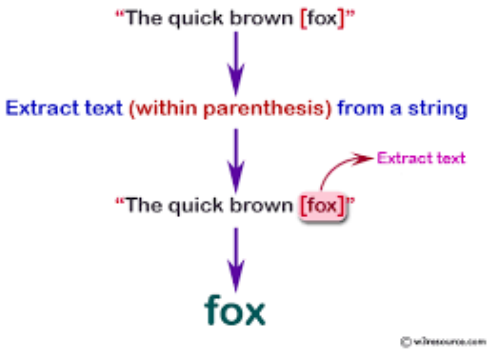
Les Expression Régulière (Regex - REGular EXpression)

Les expressions régulières (regex) sont des motifs utilisés pour effectuer des recherches et des manipulations de chaînes de caractères. Elles sont extrêmement puissantes pour valider les entrées de formulaire, rechercher des motifs spécifiques dans des textes, et bien plus encore. Voici un guide pour comprendre et utiliser les regex en PHP.



1. Utilisation de base

PHP fournit plusieurs fonctions pour travailler avec les expressions régulières. Les plus couramment utilisées sont `preg_match`, `preg_match_all`, et `preg_replace`.



2. Exemple de validation avec regex

Prenons quelques exemples courants de validation avec regex.

a. Validation d'un nom

Un nom ne doit contenir que des lettres (majuscules ou minuscules) et des espaces :

```php
$name = "Jean Dupont";
if (preg_match("/^[a-zA-Z-' ]*$/", $name)) {
    echo "Nom valide";
} else {
    echo "Nom invalide";
}
```

b. Validation d'un email

L'adresse email doit être au format standard :

```php
$email = "exemple@domaine.com";
if (preg_match("/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]
    +\.[a-zA-Z]{2,}$/", $email)) {
    echo "Email valide";
} else {
    echo "Email invalide";
}
```

c. Validation d'un numéro de téléphone

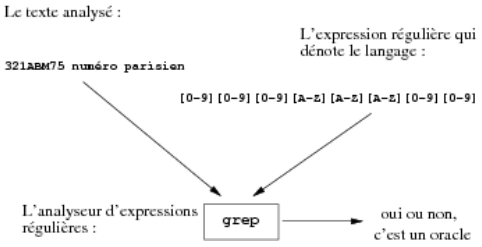
Un numéro de téléphone français typique (format 10 chiffres) :

```php
$phone = "0612345678";
if (preg_match("/^0[1-9][0-9]{8}$/", $phone)) {
    echo "Numéro de téléphone valide";
} else {
```

```
    echo "Numéro de téléphone invalide";
}
```

3. Structure des expressions régulières

Les expressions régulières peuvent sembler complexes au premier abord, mais elles suivent une structure logique.



Composants de base

- `^` : Début de la chaîne
- `$` : Fin de la chaîne
- `.` : Tout caractère (sauf le caractère de nouvelle ligne)
- `*` : 0 ou plusieurs répétitions du caractère précédent
- `+` : 1 ou plusieurs répétitions du caractère précédent
- `?` : 0 ou 1 répétition du caractère précédent
- `\d` : Un chiffre (équivalent à `[0-9]`)
- `\w` : Un caractère alphanumérique (équivalent à `[a-zA-Z0-9_]`)
- `[abc]` : Un des caractères a, b, ou c
- `[^abc]` : Tout caractère sauf a, b, ou c
- `(x|y)` : x ou y

Groupes et Répétitions

- `(abc)` : Groupe les caractères abc
- `a{2,3}` : a apparaissant entre 2 et 3 fois
- `a{3,}` : a apparaissant 3 fois ou plus

4. Exemple de script complet

Voici un exemple complet de validation de formulaire en utilisant des regex dans PHP.

Exemples concrets

	Exemple	Regex à ajouter
Numéro de téléphone français	0612344556 , +33655667789	^((\+)33|0)[1-9](\d{2}){4}$
Numéro de SIREN français	36252187900034	^[0-9]{14}$
Numéro de SIRET français	362521879	^[0-9]{9}$

```php
<?php
function test_input($data) {
   return htmlspecialchars(stripslashes(trim($data)));
}
$nameErr = $emailErr = $phoneErr = "";
$name = $email = $phone = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Validation du nom
    if (empty($_POST["name"])) {
        $nameErr = "Le nom est requis";

    } else {
        $name = test_input($_POST["name"]);
        if (!preg_match("/^[a-zA-Z-' ]*$/", $name)) {
            $nameErr = "Seules les lettres et les
            espaces sont autorisés";
        }
    }
    // Validation de l'email
    if (empty($_POST["email"])) {
        $emailErr = "L'email est requis";

    } else {
        $email = test_input($_POST["email"]);
        if (!preg_match("/^[a-zA-Z0-9._%+-]
        +@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/", $email)) {
            $emailErr = "Format d'email invalide";
        }
    }
```

```
    // Validation du numéro de téléphone
    if (empty($_POST["phone"])) {
        $phoneErr = "Le numéro de téléphone est
        requis";
    }

    else {
        $phone = test_input($_POST["phone"]);
        if (!preg_match("/^0[1-9][0-9]{8}$/", $phone)) {
            $phoneErr = "Numéro de téléphone
            invalide";
        }
    }
    if (empty($nameErr) && empty($emailErr)
        && empty($phoneErr)) {
        echo "Nom : $name<br>";
        echo "Email : $email<br>";
        echo "Téléphone : $phone<br>";

    } else {
        echo "Erreurs dans le formulaire :<br>";
        echo $nameErr . "<br>";
        echo $emailErr . "<br>";
        echo $phoneErr . "<br>";
    }
}
?>
```

Et voici le formulaire HTML correspondant :

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Formulaire de Validation</title>
</head>
<body>
    <form action="process_form.php" method="post">
        <label for="name">Nom :</label>
```

```
        <input type="text" id="name" name="name"
        required><br><br>
        <label for="email">Email :</label>
        <input type="email" id="email" name="email"
        required><br><br>
        <label for="phone">Téléphone :</label>
        <input type="text" id="phone" name="phone"
         required><br><br>
        <input type="submit" value="Envoyer">
    </form>
</body>
</html>
```

Conclusion

Les expressions régulières sont des outils puissants pour la validation des formulaires et bien d'autres manipulations de chaînes de caractères. En les maîtrisant, vous pouvez écrire des scripts PHP robustes et sécurisés.