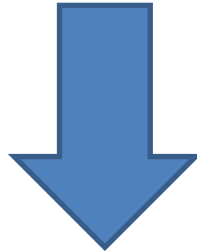




**Les superglobale des formulaires :
requêtes et données**

Rappel d'une donnée associée en développement

Paramètre = Attribut = Clé = Variable/Constante = Argument



Valeur

Par défaut, une valeur est nul

Exemples :

- param = val
- attr = val ou attr(val)
- var = val
- var[clé] = val

Chaque paramètre a un TYPE à définir : integer, double, string, boolean, pour les plus courants. Un langage à « *typage faible* » reconnaît par défaut le type du paramètre sans forcément le déclarer.



Les superglobale :

- ➔ `$_GET ?`
- ➔ `$_POST ?`
- ➔ `$_SESSION ?`
- ➔ `$_COOKIE ?`
- ➔ `$_REQUEST ?`



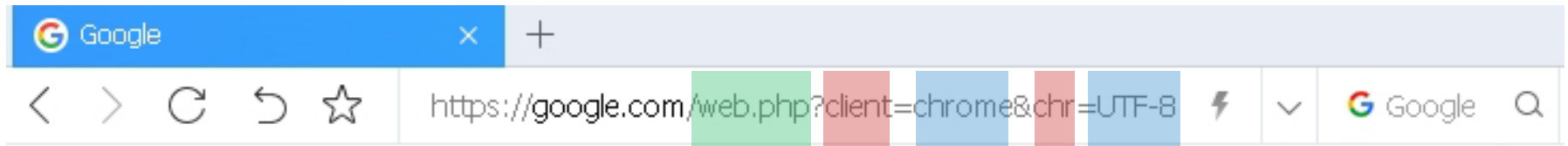
Les superglobales sont des tableaux !
Pour tester une superglobale :

```
<?php
    echo "<pre>";
    var_dump($_GET);
    var_dump($_POST);
    var_dump($_SERVER);
    //etc...
    die(); //on arrête le script
?>
```

\$_GET

Traite les données reçues depuis une URL

- ➔ Avec **\$_GET** je connais le nom de la variable et son contenu : elle est prédéfinie côté dev.
- ➔ Exemple : depuis un lien `<a href>` j'affiche dans mon application une partie sélectionnée :
site.php?article=page1



Dans cet exemple la page **web.php** contient dans l'URL les paramètres :

- client
- chr

Pour afficher les données sur la page **web.php** :

echo \$_GET["client"]; ➔ affiche sur le navigateur la valeur **chrome**

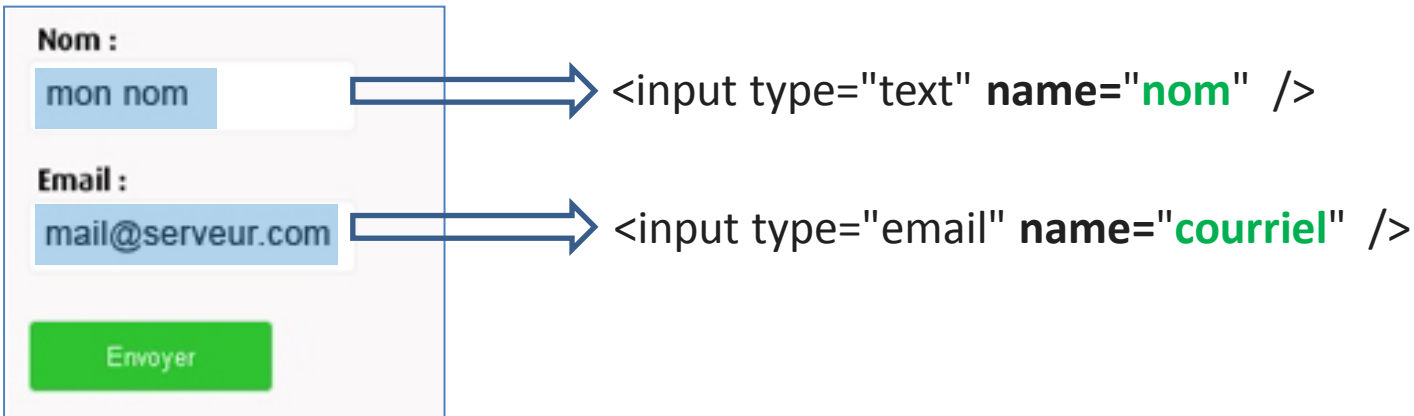
echo \$_GET["chr"]; ➔ affiche sur le navigateur la valeur **UTF-8**

\$_POST

Traite les données expédiées depuis un formulaire

- ➔ Avec **\$_POST** je connais le nom de la variable, pas son contenu : elle est indéfinie.
- ➔ Exemple : depuis la balise `<form action=>`, ou `$.post` (jQuery), ou encore `$.ajax` (jQuery), j'assigne un élément avec le paramètre **name=** qui sera le nom de la variable à envoyer.

- **SOURCE** Exemple : *formulaire.html*



Nom :

Email :

Envoyer

`<input type="text" name="nom" />`

`<input type="email" name="courriel" />`

`echo $_POST["nom"];` ➔ affiche sur le navigateur la phrase : `mon nom`

`echo $_POST["courriel"];` ➔ affiche sur le navigateur le mot `mail@serveur.com`

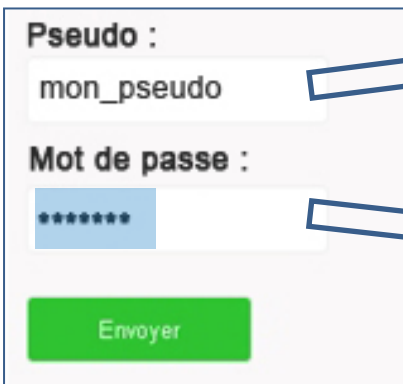
- **CIBLE** Exemple : *validate_form.php*

\$_SESSION

Conserve les données de façon temporaires sur le navigateur

- ➔ Avec **\$_SESSION** je conserve la donnée d'une variable de page en page jusqu'à la fermeture du navigateur (ou de la durée de la session sur PHP, soit 24 min par défaut).
- ➔ Je dois toujours commencer mon script PHP par **session_start();**
- ➔ Exemple : Une application web est protégée par un mot de passe. Si le mot de passe est accepté, alors une session de validation est activée. En plus, je peux afficher sur toutes les pages de mon application le *pseudo* de la personne connectée - sans passer par \$_GET.

• **SOURCE** Exemple : *welcome.html*



• **CIBLE** Exemple : *validate.php*

➔ On conserve le pseudo :

```
$_SESSION["pseudo"] = $_POST["pseudo"] ;
```

➔ On crée une session si le mot de passe est OK :

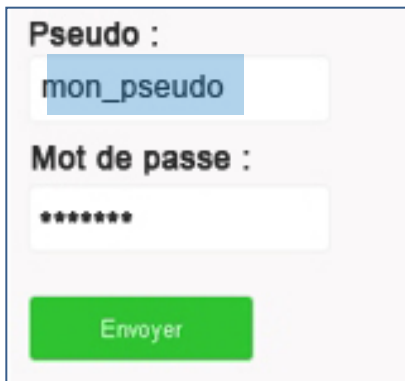
```
if ( $_POST["mdp"] === "secret" ) {  
    $_SESSION["valide"] = true ;  
}
```

\$_SESSION

Conserve les données de façon temporaires sur le navigateur

- ➔ Avec **\$_SESSION** je conserve la donnée d'une variable de page en page jusqu'à la fermeture du navigateur (ou de la durée de la session sur PHP, soit 24 min par défaut).
- ➔ Je dois toujours commencer mon script PHP par **session_start();**
- ➔ Exemple : Une application web est protégée par un mot de passe. Si le mot de passe est accepté, alors une session de validation est activée. En plus, je peux afficher sur toutes les pages de mon application le *pseudo* de la personne connectée - sans passer par \$_GET.

• RÉFÉRENCE DE DÉPART



A screenshot of a web form for login. It has two input fields: 'Pseudo :' with the value 'mon_pseudo' and 'Mot de passe :' with masked characters '*****'. Below the fields is a green button labeled 'Envoyer'.



• AUTRES PAGES PHP

- ➔ Si la session du mot de passe est validée :

```
if ( $_SESSION["valide"] ) {  
    ➔ affiche sur le navigateur le mot mon_pseudo  
    echo $_SESSION["pseudo"] ;  
}
```


\$_REQUEST

\$_GET, \$_POST et \$_COOKIE

→ **\$_REQUEST** est un tableau qui va associer toutes les requêtes de type `$_GET`, `$_POST`, et `$_COOKIE`. Comme les autres variables vus précédemment, c'est une superglobale accessible dans l'ensemble d'une application PHP.

→ Exemple de retour :

```
<?php
    echo "<pre>";
    print_r($_REQUEST);
    die;
?>
```