

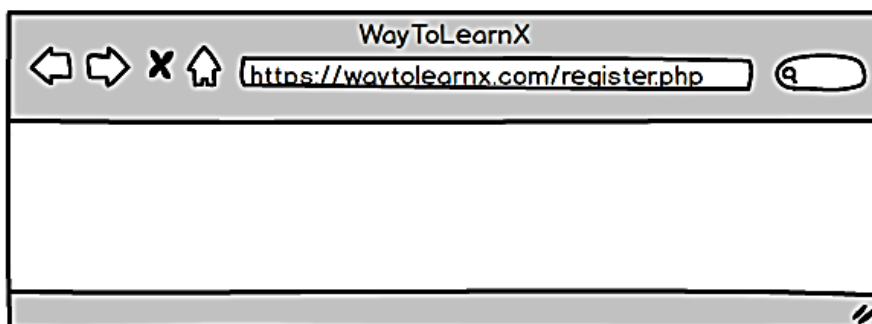
# Méthode Get Et Post

## Différence entre GET et POST dans PHP :

GET	POST
Dans le cas d'une requête GET, seule une quantité limitée de données peut être envoyée car les données sont envoyées dans l'en-tête.	En cas de requête POST, une grande quantité de données peut être envoyée car les données sont envoyées dans le corps.
La requête GET n'est pas sécurisée car les données sont exposées dans la barre d'URL.	La requête POST est sécurisée car les données ne sont pas exposées dans la barre d'URL.
La requête GET est plus efficace et utilisée plus que POST.	La requête POST est moins efficace et utilisée moins que GET.
Les paramètres restent dans l'historique du navigateur car ils font partie de l'URL.	Les paramètres ne sont pas enregistrés dans l'historique du navigateur.
Les requêtes GET sont ré-exécutées mais ne peuvent pas être soumises au serveur si le code HTML est stocké dans la mémoire cache du navigateur.	Le navigateur prévient généralement l'utilisateur que les données devront être soumises à nouveau.
Seuls les caractères ASCII sont autorisés.	Pas de restrictions. Les données binaires sont également autorisées.
GET est moins sécurisé que POST car les données envoyées font partie de l'URL. Donc, il est enregistré dans l'historique du navigateur et les journaux du serveur en texte brut.	POST est un peu plus sûr que GET car les paramètres ne sont pas stockés dans l'historique du navigateur ou dans les journaux du serveur Web.
La méthode GET ne doit pas être utilisée lors de l'envoi de mots de passe ou d'autres informations sensibles.	Méthode POST utilisée lors de l'envoi de mots de passe ou d'autres informations sensibles.
La méthode GET est visible par tout le monde (elle sera affichée dans la barre d'adresse du navigateur) et limite le nombre d'informations à envoyer.	Les variables de méthode POST ne sont pas visibles dans l'URL.
Peut être mis en cache	Ne peut être mis en cache

## Définition du méthode GET

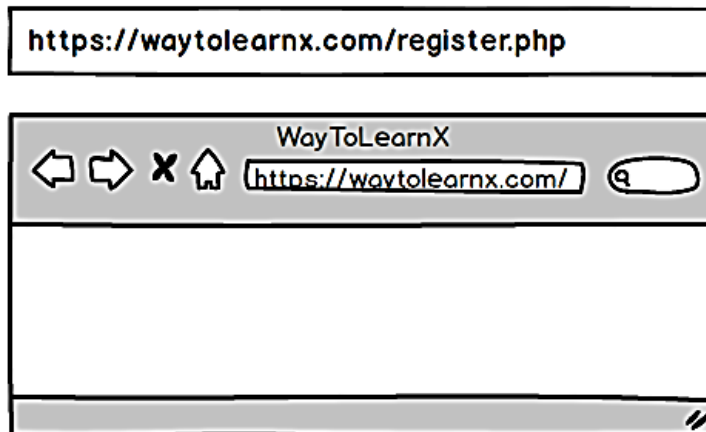
<https://waytolearnx.com/register.php?name=alex&age=22>



Les paramètres sont envoyés dans l'URL d'une requête GET:

1. GET/register.php?name=alex&age=22

## Définition du méthode POST



Les paramètres sont envoyés dans le corps du message HTTP pour une requête POST:

1. POST/register.php HTTP/1.1
2. Hôte: www.waytolearnx.com
3. name=alex&age=22

Comme nous le savons, en cas de requête POST, les données d'origine sont envoyées dans le corps du message.

### Conclusion :

Les deux méthodes GET et POST sont utilisées pour transférer des données du client au serveur avec le protocole HTTP. La différence clé entre les méthodes POST et GET est que GET transporte le paramètre dans la chaîne d'URL, tandis que POST transporte le paramètre dans le corps du message, ce qui le rend plus sûr le transfert des données du client au serveur avec le protocole http.

### Les variables :

- Une variable ne doit pas commencer par un chiffre.

```
// Déclaration et règles
$var=1; // $var est à 1
$Var=2; // $ Var est à 2
$_toto='Salut'; // Ok
$3petitscochons=5; // Invalide : commence par un chiffre
```

### Les Chaines de caractères :

Une chaîne est une séquence de caractères. PHP travaille en ASCII soit 256 caractères, mais ne supporte pas encore le format Unicode, prévu dans la version 5. Il n'y a pas de limite théorique pour la taille de la chaîne.

On distingue trois syntaxes pour utiliser un chaîne

- Les guillemets simples `'...'` (apostrophes) : Comme en shell, tous les caractères inclus dans la chaîne sont sortis tels quels sans interprétation. Si vous devez afficher un guillemet simple, il faudra l'échapper : `\'`
- Les guillemets doubles `"..."` : Certaines séquences de caractères sont interprétées et les variables sont substituées (remplacées par leur valeur).
- HereDoc : Identique aux HereScripts en Shell. Le texte saisi jusqu'à un délimiteur spécifié est placé dans la variable.

Voici le tableau issu de la documentation PHP des séquences pouvant être utilisés avec les guillemets doubles.

Séquence	Valeur
<code>\n</code>	Nouvelle ligne (linefeed, LF ou 0x0A (10) en ASCII)
<code>\r</code>	Retour à la ligne (carriage return, CR ou 0x0D (13) en ASCII)
<code>\t</code>	Tabulation horizontale (HT ou 0x09 (9) en ASCII)
<code>\\</code>	Antislash
<code>\\$</code>	Caractère \$
<code>\"</code>	Guillemets doubles
<code>\[0-7]{1,3}</code>	Une séquence de caractères qui permet de rechercher un nombre en notation octale.
<code>\x[0-9A-Fa-f]{1,2}</code>	Une séquence de caractères qui permet de rechercher un nombre en notation hexadécimale.

## Comment se connecter à une base de données :

```
Mysql_connect(« localhost », « root » ; « »);
```

```
Mysql_select_db(« nom de la base »);
```