

# Transmission de données du client vers le serveur

Peut se faire via :

- URL → méthode GET
  - Taille des données transmises limitée (taille URL < 256, limite du serveur 2000)
  - Paramètres lisibles dans l'URL
  - Utilisation :
    - Transmission de données de page en page (à éviter)
    - Création de lien hypertexte avec paramètres pour réduire le nombre de pages à créer
    - Lecture d'information du serveur
- Formulaire → méthode POST
  - Aucune limite dans la taille des données
  - Paramètre invisibles dans l'URL
  - Utilisation :
    - Envoi de données ou de fichiers
    - Envoi via les formulaires
    - Écriture ou mise à jour d'informations sur le serveur

# Transmission de données du client vers le serveur ► via URL

- Invoquer une url avec des paramètres de la forme :
  - <http://mysite.com/mypage.php?parameterName1=value&parameterName2=value>
    - ? : séparateur entre le nom de la page et le début des paramètres
    - = : séparateur entre le nom du paramètre et sa valeur
    - & : séparateur entre paramètres
- Récupérer, dans la page mypage.php, la liste des paramètres à partir de la variable super globale `$_GET`
  1. Vérifier l'existence de la variable avec la fonction `isset()`
  2. Transtyper la valeur de la variable dans le bon type (`type`)
  3. Contrôler la valeur de la variable par rapport aux conditions métier
  4. Effectuer le traitement quand tous les contrôles sont bons

# Transmission de données du client vers le serveur ► via URL

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width">
<title>Réception de paramètres
dans l'URL</title>
</head>
<body>
  <p>
    <?php
if (isset($_GET['prenom']) AND
isset($_GET['nom']))
  echo 'Bonjour ' .
$_GET['prenom'] . ' ' .
$_GET['nom'] . ' !';
else
  echo 'Il faut renseigner un nom
et un prénom !';
?> </p>
  </body>
</html>
```

Page bonjour.php



- Dans la page index.php :

```
<a
href="bonjour.php?nom=Dupont
&prenom=Jean">Dis-moi
bonjour !</a>
```

Dis-moi bonjour !



Bonjour Jean Dupont !

# Transmission de données du client vers le serveur ► via formulaire

- Construire un formulaire HTML
  - Renseigner dans la balise form les attributs
    - action : la page cible chargée de traiter les données reçues
    - method : la methode http utilisée pour envoyer les données
      - GET : Transmission par URL (très peu utilisée) ➔ Récupération en php via \$\_GET
      - POST : Transmission des données dans le body de la requête ➔ Récupération en php via \$\_POST
  - Créer les éléments de formulaire pour la saisie des données
    - Le nom des champs correspond au nom des variables obtenues dans la variable super globale \$\_GET ou \$\_POST
    - Renseigner les balises html de validation des données saisies
  - Créer le bouton de type submit pour envoyer les données du formulaire

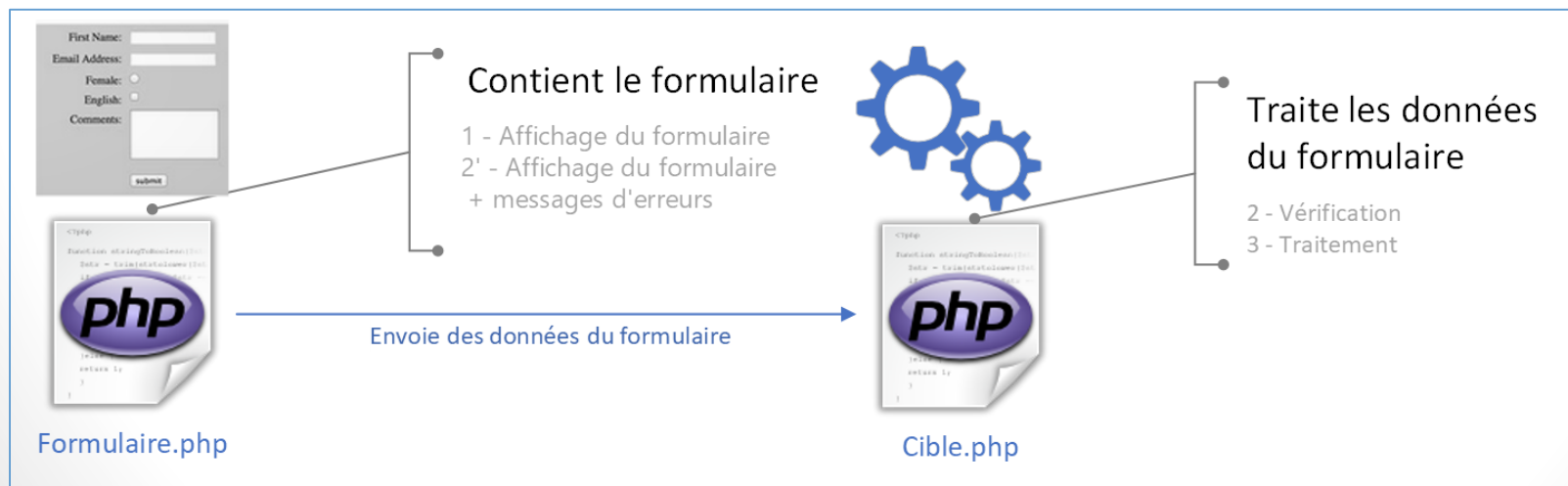
# Transmission de données du client vers le serveur ► via formulaire

- Construire la page cible chargée de traiter les données
  1. Vérifier l'existence des variables du formulaire avec la fonction `isset()`
  2. Récupérer les variables envoyées par le formulaire dans `$_GET` ou `$_POST`
  3. Transtyper la valeur des variables dans le bon type (`type`)
  4. Contrôler la valeur des variables par rapport aux conditions métier
  5. Si tous les contrôles sont bons alors effectuer le traitement métier prévue
  6. Sinon réafficher le formulaire avec les messages d'erreurs et de préférence les anciennes bonnes valeurs saisies (ergonomie)
- Il est possible que la page formulaire joue aussi le rôle de la page cible ➔ la même page affiche le formulaire et réalise le traitement
  - La variable `$_SERVER['PHP_SELF']` permet de trouver l'url de la page courante

# Transmission de données du client vers le serveur ► via formulaire

## Solution de deux pages différentes

- ☺ Simplicité à mettre en œuvre
- ☹ Difficulté à afficher le formulaire une 2<sup>ème</sup> fois avec les messages d'erreurs et le contenu saisi
- Technique de redirection (instruction header)
- Utilisation des techniques de transmission de données entre deux invocations HTTP
  - Paramètres dans l'URL
  - Sessions ou cookies



# Transmission de données du client vers le serveur ► via formulaire

## Solution d'une page unique avec formulaire auto invoquant

- ☺ Facilité à afficher le formulaire une 2<sup>ème</sup> fois avec les messages d'erreurs et le contenu saisi
  - Ergonomie de l'utilisateur
- ☹ Algorithme plus complexe à mettre en œuvre

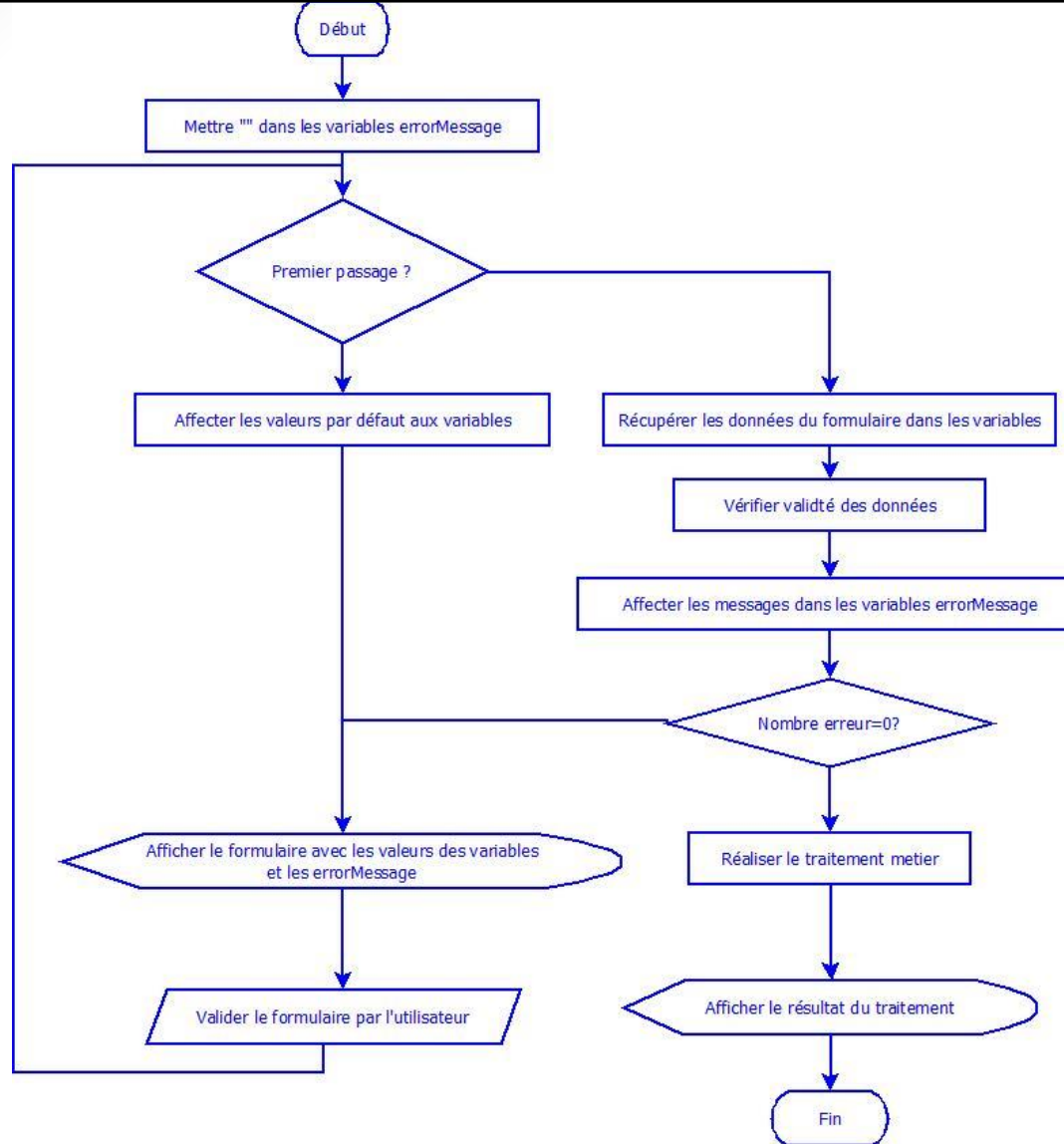


Formulaire.php



Contient le formulaire et son traitement

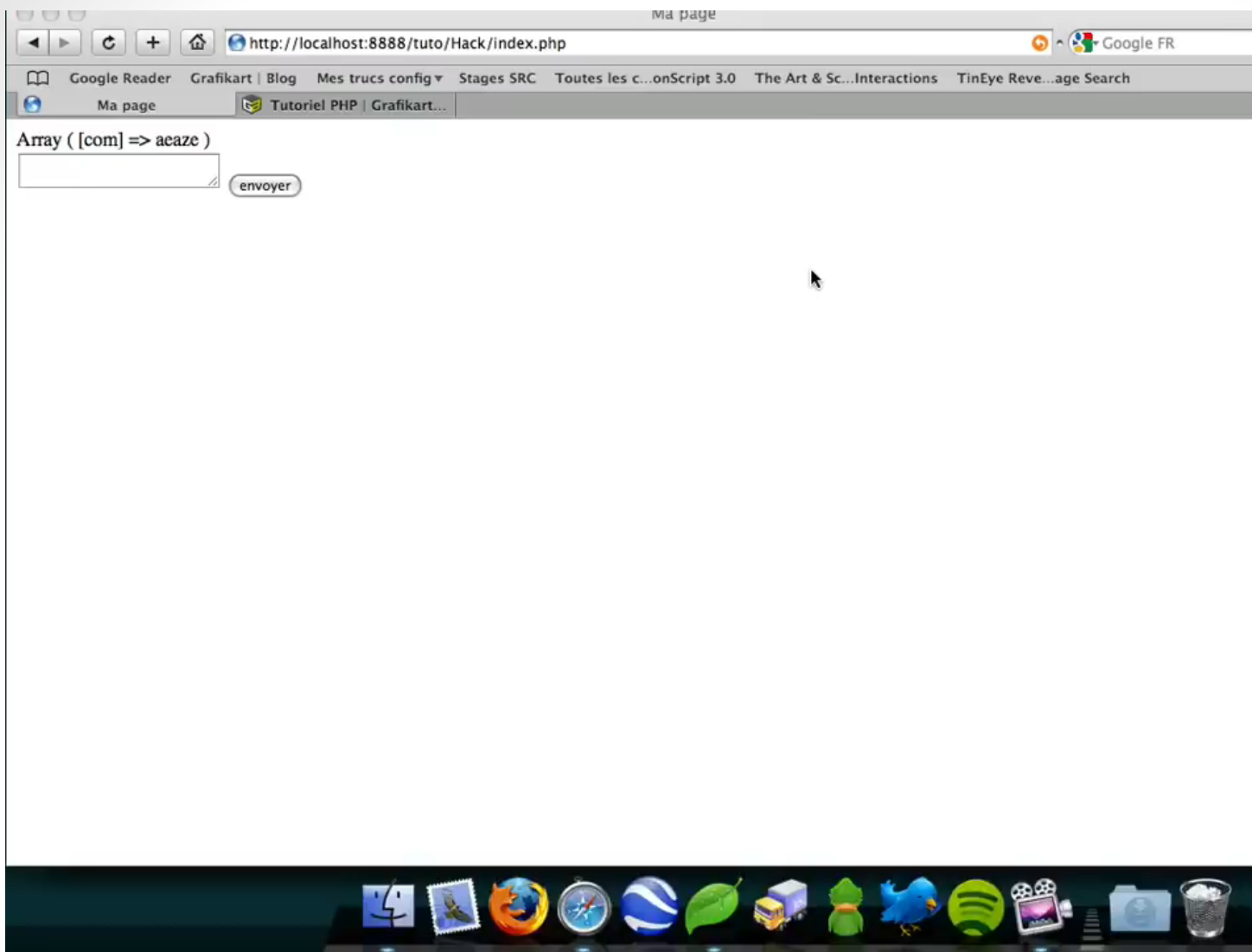
- 1 - Affichage du formulaire
- 2 - Vérification des données  
réaffichage du formulaire en cas d'erreur avec les messages
- 3 - Traitement si pas d'erreur et affichage du résultat



## Transmission de données du client vers le serveur ► via formulaire

Algorithme de la solution d'une page unique avec formulaire auto invoquant





**Transmission de données du client vers le serveur ► faille XSS**

Exercices de l'Activité A2 sur l'espace Moodle

# PAUSE ACTIVITÉ PRATIQUE