



Bachelier en Informatique de Gestion

Projet de Développement Web

Enseignement supérieur économique de type court

Code FWB : 7534 30 U32 D3

Code ISFCE : 4IPW3



= BUROTIX ()

Table des matières

Généralités

- 01. Introduction au web
- 03. Outils
- 05. Frameworks

Côté Client

- 12. Structure HTML
- 13. Formulaire HTML
- 14. Mise en forme CSS
- 15. Adaptabilité
- 17. Javascript
- 18. Framework jQuery
- 19. AJAX

Côté Serveur

- 21. Middleware PHP
- 22. [Traitement du formulaire](#)
- 23. Architecture MVC
- 24. Base de données SQL
- 25. Données XML
- 26. Données JSON



22. Traitement du Formulaire

Transit des données \$_GET et \$_POST

Balises \$_SESSION

Super Globales PHP \$_COOKIES



= BUROTIX ()

Transit des données



= BUROTIX ()

Objectifs des formulaires

- La création d'un formulaire se fait via la balise `<form>`
- Les formulaires sont utilisés pour récolter des informations des utilisateurs.
- Deux problèmes:
 - Comment envoyer les données au serveur ?
 - Comment le serveur traite-t-il les données reçues?



Comment faire transiter les données?

- Deux méthodes :
 - **GET**: envoie les données dans l'URL de la page
 - `https://www.google.com/search?q=developpement+photo`
 - Limité à 255 caractères
 - Paramètres visibles
 - **POST**: envoie les données via la requête HTTP
 - Permet de faire transiter un plus gros nombre de caractères
 - Paramètres invisibles
- Défini avec l'attribut "**method**" de la balise **<form>**
 - `<form method="get" ...`
 - `<form method="post" ...`



Comment traiter les données?

- Il faut envoyer la requête contenant les données du formulaire (via GET ou POST) à un **script** qui pourra les traiter (ex. page contenant du PHP)
- Défini avec l'attribut action

```
<form method="get" action="./register.php" ...
```



Exemple de formulaire

- Déclaration d'un formulaire

```
<form method="get" action="./register.php">
```

```
...
```

```
</form>
```





Balises HTML Form

Rappel



= BUROTIX ()

La balise <input>

- L'élément le plus populaire d'un formulaire est la balise **<input>**.
 - utilisé de nombreuses manières différentes
 - en fonction de la valeur de son attribut **type**.

Prénom:

☐

Pop

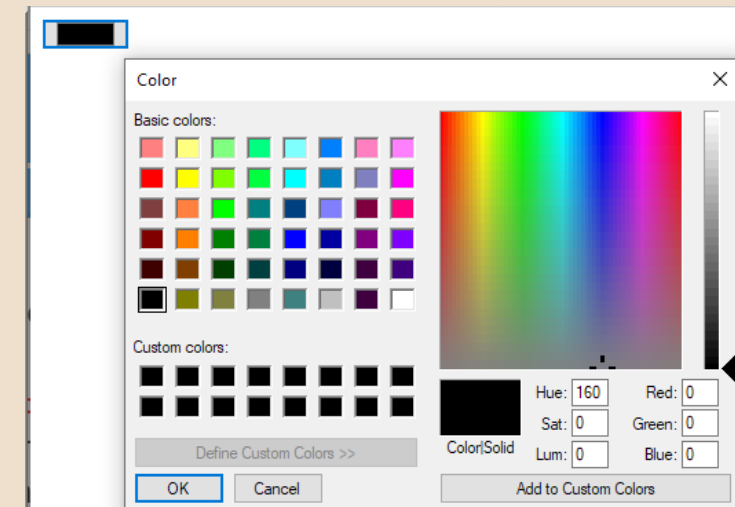
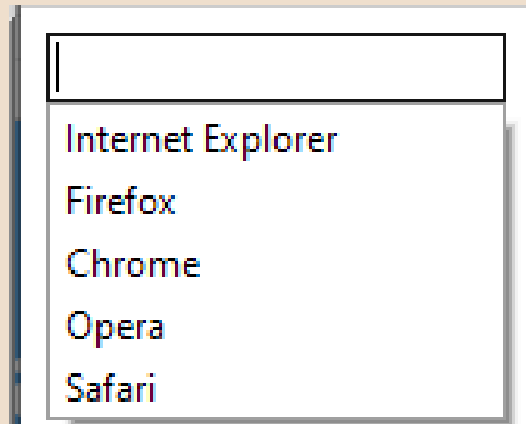
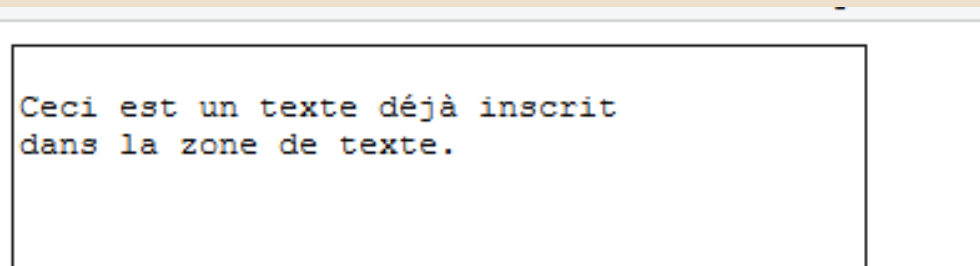
☐

Rock

Chocolate
Strawberry
Vanilla

Autres balises

- `<textarea rows="10" cols="30">`
- `<input list="browsers">`
`<datalist id="browsers">`
- `<label for="male">Male</label>`
- `<input type="color">`



Soumettre un formulaire



- Une fois rempli, le formulaire est "soumis au serveur", c  d que les donn  es y sont envoy  es.
- Bouton de type “submit”
 - `<button type="submit">S'inscrire</button>`
 - remarque : il existe un codage ancien
 - `<input type="submit" value="S'inscrire">`
- Bouton de type “button”
 - appel d'un code JavaScript = programmation !
 - formulaire non envoy   automatiquement au serveur
 - exemple : un tel bouton peut valider les input, envoyer le form au serveur et afficher "veuillez patienter".



= BUROTIX ()

Exo 01 : Sample "Atomic" Form

- Télécharger et placer dans le WAMP les deux fichiers
 - `exo01_sample_atomic_form.html`
 - `exo01_sample_atomic_form.php`
- Démo de
 - Input type text
 - Input type password
 - Input type radio
 - Input type checkbox
 - Select option
 - Input + Datalist
 - Input type color
 - Input type range
 - Button type submit
 - Button type button




Exo 03 : Sample "Molecular" Form

- Télécharger et placer dans le WAMP les deux fichiers
 - `exo03_sample_atomic_form.html`
 - `exo03_sample_atomic_form.php`
- Démo de
 - Image Upload
 - File Upload



Exo 05 : Catalogue + Form Input

- Créer un formulaire qui permet d'ajouter des éléments dans le tableau HTML.
- Ci-contre un exemple de formulaire.
- fichiers exo :
 - database_catalogue.csv
 - exo05_reading_csv.php



1	produit;prix
2	bière;10
3	sandwich;3
4	verre;5
5	assiette;32
6	couvert;10

Catalogue	
produit	prix
bière	10
sandwich	3
verre	5
assiette	32
<input type="text"/>	<input type="text"/>
<input type="button" value="add"/>	

Super Globales : \$_GET et \$_POST



= BUROTIX ()

\$_GET et \$_POST

- Les variables super globales **\$_GET** et **\$_POST** permettent de récupérer des données envoyées via l'url ou la requête HTTP
- **\$_GET** : url
- **\$_POST** : http



\$_GET et \$_POST

- tableaux associatifs : éléments accessibles via une clé
- URL

`http://localhost/exercice.php?name=jean&age=23`

- PHP

```
<?php  
echo $_GET['name']; // affiche 'jean'  
echo $_GET['age'];  // affiche '23'  
?>
```



Exo 06 : lire le form

- Comment parcourir les données reçues du formulaire ?

```
foreach( $_GET as $key => $val )  
{  
    echo "$key : $val<br/>";  
}
```

- ou

```
print_r($_GET) ;  
var_dump($_GET) ;
```



Exo 07 : login élémentaire et bibliothèque de fonction

- Écrivez un programme qui
 - Lit un login entré par l'utilisateur
 - Évalue si ce login est correct (un seul login hard-codé)
 - Affiche en conséquence un message de bienvenue ou de rejet
 - Organisation du code en fonctions PHP

```
html_head($title="4IPDW")  
html_welcome($text)  
html_login_form($intro="", $value="")  
html_welcome_user($people)  
html_foot()
```

Welcome everybody to 4IPDW world !

Votre login :

Welcome everybody to 4IPDW world !

Go to hell
Votre login :

Welcome everybody to 4IPDW world !

Hello Reda. Do what you want.

Exo 11 : Catalogue + Insertion CSV

- A la suite des exercices précédents, écrivez le code PHP correspondant au formulaire.
- Fonction : Le script PHP insère les données dans le fichier CSV, puis réaffiche le tableau.
- fichiers exo :
 - database_catalogue.csv
 - exo11_catalogue.php

Catalogue	
produit	prix
bière	10
sandwich	3
verre	5
assiette	32
couvert	10
pain	1.5
vin	5
whisky	50
cognac	40
<input type="button" value="add"/>	



Super Globales : \$_SESSION



= BUROTIX ()

Définition

- En informatique et en télécommunication, une **session** est une période délimitée pendant laquelle un système informatique ou "serveur" est en **communication** avec et réalise des **opérations** pour un "client" - un usager, un logiciel ou un autre système.



session_start()

- Crée une session ou restaure celle trouvée sur le serveur, via l'identifiant de session passé dans une requête **GET**, **POST** ou par un cookie.



session_unset()

- Détruit toutes les données associées à la session courante.
- Si **session_unset()** n'est pas explicitement lancée, alors la session sera détruite automatiquement après un "time-out".



\$_SESSION

- La variable super globale **\$_SESSION** permet d'enregistrer des données coté serveur sur un utilisateur pendant une session
- **\$_SESSION**
 - est **unique** pour chaque session
 - ne dure **que le temps** de la session
 - s'utilise comme un tableau associatif



isset (\$var)

- Détermine si la variable **\$var** est définie
- return : boolean



Exo 21 : log-in, log-out

- Ecrivez un programme php qui permet à l'utilisateur de se logger, en saisissant son nom dans un input field.
- Ensuite, écrivez un programme php qui permet à l'utilisateur de se délogger.
- Le login de l'utilisateur est enregistré dans une variable **`$_SESSION`**.

Votre nom :

log in

Bonjour Alfred

log out



Exo 24 : log-in et identification

- Reprenez le formulaire HTML de l'exercice 06-21 et ajoutez-y un mot de passe.
- Vérifier si le login et le mot de passe sont corrects et si l'utilisateur peut être connecté.
- Tous les login et mot de passe sont stockés de manière non cryptée dans un fichier CSV.
- fichiers exo :
 - database_login.csv
 - exo24_login_password_session.php



Super Globales : \$_COOKIE



= BUROTIX ()

Qu'est-ce qu'un cookie?

- Un petit fichier que le serveur web place sur l'ordinateur de l'utilisateur.
- Application populaire:
 - Conserver l'identité d'un utilisateur de page en page
- En PHP, vous créez, lisez, modifiez et supprimez vos propres cookies.



Créer des cookies avec PHP

```
setcookie( name, value, expire, path,  
          domain, secure, httponly );
```

- **name** : nom du cookie
- **value** : valeur du cookie
- **expire** : date-heure d'expiration du cookie
 - souvent calculé ainsi : date-heure courante + durée de validité
 - si **expire** vaut **0** ou est omis, alors le cookie expirera à la fin de la session.
- **path** = **"/"** : le répertoire en dessous duquel le cookie est disponible
 - dans ce cours : à mettre à **"/"** à des fins de test
- A placer **AVANT** la balise **<html>**



Créer des cookies avec PHP : exemple

```
$name = "user";  
$value = "John Doe";  
$expire = time() + (86400 * 30);  
setcookie( $name, $value, $expire, "/" );
```

- Ce code crée un cookie nommé "user" avec la valeur "John Doe", qui expirera après 30 jours (86400 * 30).



Lire un cookie

- super globale `$_COOKIE`
 - Lire la valeur du cookie, par ex.
`echo $_COOKIE["user"];`
 - Associative array !
- `isset()`
 - Savoir si le cookie est défini



Modifier un cookie

- On définit à nouveau le cookie avec les nouvelles valeurs.

```
setcookie(name, value, expire, "/" );
```

- **name** : nom du cookie (inchangé)
 - **value** : nouvelle valeur du cookie
 - **expire** : nouvelle date-heure d'expiration du cookie, etc.
- Exemple

```
setcookie( "user", "Alex Porter" );
```



Supprimer un cookie

- On définit à nouveau le cookie, mais avec une date d'expiration dans le passé.

```
setcookie( name, value, expire, "/" );
```

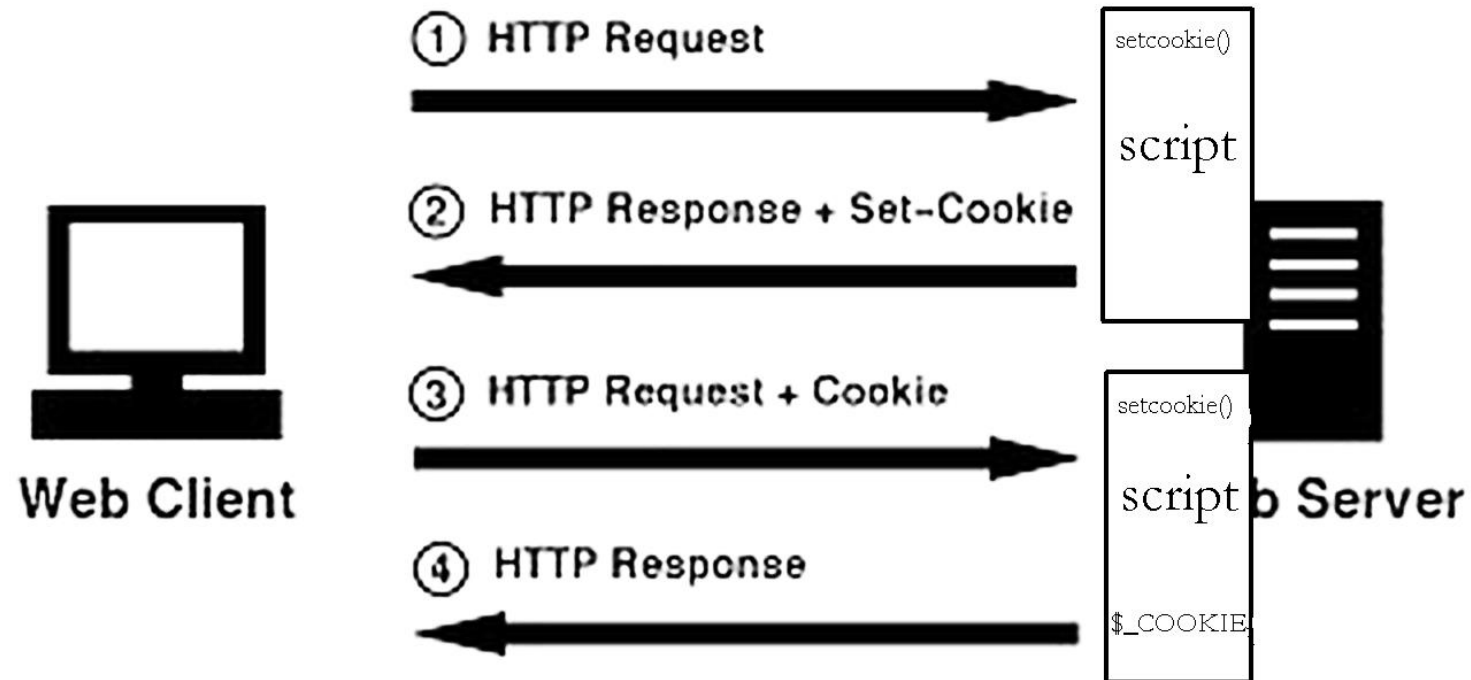
- *expire* : valeur dans le passé, par ex. *1*

- Exemple

```
setcookie( "user", "John Doe", 1, "/" );
```



Architecture des cookies



Exo 27 : log-in, log-out

- Ecrivez un programme php qui permet à l'utilisateur de se logger, en saisissant son nom dans un input field.
- Ensuite, écrivez un programme php qui permet à l'utilisateur de se délogger.
- Le login de l'utilisateur est enregistré dans une variable `$_COOKIE`.
- fichier exo :
 - `exo27_login_cookie.php`

Votre nom :

Bonjour Alfred

