

## Module 133

### Chapitre 01 - Introduction

#### **Exercice 01-01** [chapitre 2 dans le livre]

Faites fonctionner cet exemple après avoir installé PHP (à partir de WAMP par exemple).

```
<?php
    echo "Hello world !!";
?>
```

#### **Exercice 01-02**

Faites fonctionner cet exemple :

```
<html>
<head>
</head>
<body>
<?php

echo "Hello world !!";

?>
</body>
</html>
```

#### **Exercice 01-03** [chapitre 5 dans le livre]

Trouvez l'erreur dans ce code et corrigez-la :

```
<html>
<head>
</head>
<body>
<?php

$VAR1 = 5;
echo ($var1);

?>
</body>
</html>
```

#### Exercice 01-04

- a) Faites fonctionner l'exemple du cours en comprenant bien les différences.  
b) Modifiez ensuite le code pour avoir chaque expression qui s'affiche sur une nouvelle ligne.

```
<?php
$a = 1;
$b = 7;

$str1 = "hello";
$str2 = " world";

echo $a.$b;
echo $str1.$str2;
echo "Hello $str2";
?>
```

#### Exercice 01-05

Devinez quel va-t-être l'affichage de cette page et testez-la pour vérifier :

```
<?php
$nom1 = "john";
$nom2 = "sixpack";
echo "a) Hello $nom1 <br>";
echo 'b) Hello $nom2 <br>';
echo "c) Hello $nom1.$nom2 <br>";
echo 'd) Hello $nom1.$nom2 <br>';
echo 'e) Hello '.$nom1.$nom2.' <br>';
?>
```

#### Exercice 01-06

Trouvez l'erreur de syntaxe, dans ce code et corrigez-la, puis corrigez l'affichage :

```
<?php
$nom1 = "john";
$nom2 = "sixpack";
echo 'Hello '$nom1.$nom2' <br>';
?>
```

**Exercice 01-07** [chapitre 6 dans le livre]

Faites fonctionner l'exemple suivant en expliquant pourquoi chaque point est exécuté ou non :

```
<?php
$a = '13';
$b = 13;

if ( $a == $b )
    echo "a) Nous avons la même valeur <br>";
if ( $a === $b )
    echo "b) Nous avons la même valeur et le même type. <br>";

if ( "12" === "14" )
    echo "c) ne serra pas affiché <br>";

if ( $a !== $b )
    echo "d) Nous n'avons pas (la même valeur ou) le même type <br>";

if ( "100" == "1e2" )
    echo "e) Pourquoi donc ? <br>";

if ( 0 == "un" )
    echo "f) Pourquoi encore ? <br>";

if ( 0 == "1+1" )
    echo "g) Pourquoi pas ? <br>";

if ( "2" == 1+1 )
    echo "h) Ça calcule ? <br>";
?>
```

**Exercice 01-08**

Devinez quel va-t-être l'affichage de cette page et testez-la pour vérifier :

```
<?php
    $a = true;
    $b = false;
    $c = $a && $b;
    $d = $a || $b;
    if ( $c )
    {
        echo "A-";
    }
    if ( $d )
        echo "B-";
    if ( $c && $d )
    {
        echo "C-";
    }
    if ( $c || $d )
        echo "D-";
?>
```

**Exercice 01-09**

Faites un petit programme qui affiche les nombre de 1 à 20 sur une ligne :

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,

**Exercice 01-10**

Faites un petit programme qui affiche tous les livrets de 1 à 12, un par ligne.  
Autrement dit une page avec :

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
[...]
12 x 10 = 120
12 x 11 = 132
12 x 12 = 144
```

**Exercice 01-11** [chapitre 8 dans le livre]

Créez une fonction "echoLn()" qui permet d'écrire une chaîne de caractère suivi automatiquement d'un retour à la ligne.

Cet exemple :

```
echoLn("bonjour Alf") ;  
echoLn("bonjour Olgi") ;
```

Doit avoir une page avec ce résultat :

```
bonjour Alf  
bonjour Olgi
```

**Exercice 01-12**

Faites un petit programme (un seul fichier) avec :

- a) Une fonction qui prend deux arguments en paramètre et qui retourne le booléen "true" lorsque la somme des deux vaut 1156.
- b) Une fonction qui prend deux arguments en paramètre et qui retourne le booléen "true" lorsque la multiplication des deux vaut 314203.
- c) Une fonction qui prend deux arguments en paramètre et qui retourne le booléen "true" lorsque la somme des deux vaut 1156 **et** multiplication des deux vaut 314203.
- d) Utilisez une boucle et la fonction pour trouver une paire de nombres dont la somme vaut 1156 et le produit vaut 314203. Il existe une solution entière en dessous de 1000 pour les deux chiffres.

**Exercice 01-13**

- a) un petit programme qui affiche les nombres premiers plus petits que 1000.
- b) un petit programme qui affiche les nombres premiers plus petits que 1000, en utilisant une fonction qui sait vérifier si un nombre est premier.
- c) un petit programme qui affiche 1000 nombres premiers.

**Exercice 01-14**

- a) Utilisez une variable globale \$br qui permet de choisir quelle balise de retour à la ligne est utilisée dans la fonction "echoLn()".
- b) Trouvez une deuxième solution au même problème (en utilisant l'autre technique vue dans la présentation en page 21).

**Exercice 01-15**

- a) Créez un "Hello World" en important l'entête et le pied de page html à partir de fichiers séparés.
- b) Utilisez l'encodage UTF-8 pour afficher correctement des accents dans une page. De cette manière vous pouvez utiliser directement les caractères accentués dans votre code.

## Chapitre 02 - Tableau

### Exercice 02-01

Créer un programme qui permette de stocker les jours de la semaine dans un tableau.

- a) Insérer les valeurs dans un premier tableau en utilisant `array()`
- b) Insérer les mêmes valeurs dans un 2<sup>e</sup> tableau en utilisant les crochets.
- c) Afficher le nombre d'éléments qui est stocké dans chaque tableau.
- d) Afficher le contenu du tableau en utilisant une boucle **for**.  
Voici à quoi doit ressembler le résultat :

Indice 0: Lundi  
Indice 1: Mardi  
Indice 2: Mercredi  
Indice 3: Jeudi  
Indice 4: Vendredi  
Indice 5: Samedi  
Indice 6: Dimanche

- e) Afficher le contenu du tableau en utilisant une boucle **while**. (même résultat)
- f) Utilisez une boucle **for** pour afficher le résultat sous forme de table HTML :  
(partie difficile mais importante !)

| Indice | Valeur   |
|--------|----------|
| 0      | Lundi    |
| 1      | Mardi    |
| 2      | Mercredi |
| 3      | Jeudi    |
| 4      | Vendredi |
| 5      | Samedi   |
| 6      | Dimanche |

### Exercice 02-02

a) Créez un tableau associatif dans une variable `$student` avec les champs et les valeurs suivantes :

| Champ      | Valeur  |
|------------|---------|
| first_name | John    |
| last_name  | Sixpack |
| age        | 27      |
| height     | 1.62    |

b) Affichez le contenu de cette variable sous forme de liste textuel, par exemple :

first\_name : John  
last\_name : Sixpack  
age : 27  
height : 1.62

c) Affichez le contenu de cette variable sous forme de table HTML (comme affichée au point a).

### Exercice 02-03

a) Créez un tableau `$guy` avec les champs et les valeurs suivantes :

| Valeur |
|--------|
| Marc   |
| Müller |
| 21     |
| 1.52   |

b) à partir de la variable `$guy`, extraire les 4 variables suivantes et les afficher :

- `$first_name`
- `$last_name`
- `$age`
- `$height`

Exemple d'affichage :

Prénom : Marc  
Nom : Müller  
Age : 21  
Taille : 1.52

### Exercice 02-04

Créez une variable contenant la chaîne de caractère "Un poisson nage dans l'eau sans se noyer".

a) À partir de cette variable, affichez automatiquement un mot par ligne.

Résultat attendu :

Un  
poisson  
nage  
dans  
l'eau  
sans  
se  
noyer

b) Triez les mots avant de les afficher :

Résultat attendu :

Un  
dans  
l'eau  
nage  
noyer  
poisson  
sans  
se

### Exercice 02-05

Créer un programme qui permette de stocker les jours de la semaine dans un tableau.

Utilisez une boucle **foreach** pour afficher le résultat sous forme de table HTML :

| Indice | Valeur   |
|--------|----------|
| 0      | Lundi    |
| 1      | Mardi    |
| 2      | Mercredi |
| 3      | Jeudi    |
| 4      | Vendredi |
| 5      | Samedi   |
| 6      | Dimanche |



### Exercice 02-06

- Créer une structure de données pour représenter le tableau suivant à l'aide de PHP.
- Créer une fonction qui permet d'afficher la structure de donnée dans une page web
- Créer une fonction qui permet d'afficher n'importe quel tableau à 2 dimensions

| No | Nom      | Couleur | Energie [kcal/100g] |
|----|----------|---------|---------------------|
| 1  | Carotte  | Orange  | 40                  |
| 2  | Courges  | Orange  | 45                  |
| 3  | Navets   | Blanc   | 37                  |
| 4  | Poireaux | Vert    | 39                  |

Résultat attendu :

|         |        |    |
|---------|--------|----|
| carotte | orange | 40 |
| courge  | orange | 45 |
| navet   | blanc  | 37 |
| poireau | vert   | 39 |

### Exercice 02-07

- Reprenez l'exercice précédent en ajoutant
- un entête à chaque colonne (étiquettes)
  - l'indice du tableau

Résultat attendu :

| no | nom     | couleur | energie |
|----|---------|---------|---------|
| 1  | carotte | orange  | 40      |
| 2  | courge  | orange  | 45      |
| 3  | navet   | blanc   | 37      |
| 4  | poireau | vert    | 39      |

## Chapitre 03

### Paramètres URI avec \$\_GET[]

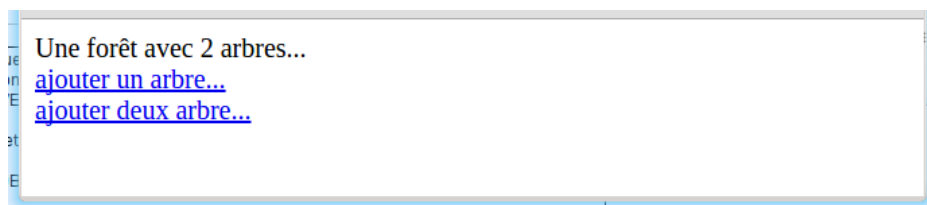
#### Exercice 03-01 [livre chapitres 9, 10, 11]

Créez une petite page web avec un compteur d'arbres.

a) Avec un lien pour aller sur une page avec un arbre de plus :



b) Avec un lien pour aller sur une page avec deux arbres de plus :



c) Corrigez dynamiquement les fautes de pluriel (1 arbre → 2 arbres → 3 arbres → ...).

### Exercice 03-02 [livre : chapitres 9, 10]

Créez une petite page web avec deux compteurs :

a) un compteur d'arbres, et un lien pour ajouter un arbre

b) un compteur de branche par arbre, et un lien pour ajouter une branche par arbre

Affichez aussi le nombre total de branches...



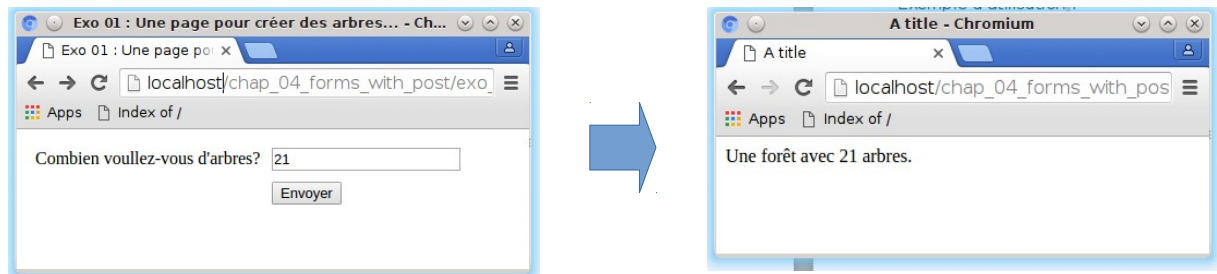
c) Corrigez les fautes de pluriel (arbre et branche).

## Chapitre 04

### Formulaires en PHP avec \$\_POST[]

#### Exercice 04-01 [livre : chapitres 11, 12]

Créez une page en *HTML* avec un formulaire demandant le nombre d'arbres désirés. Une fois validé, l'utilisateur est redirigé sur une page avec ce nombre d'arbres. Exemple d'utilisation :



#### Exercice 04-02

- a) Protégez une page par un mot de passe.
- b) Utilisez la fonction *require\_once()* pour séparer le html et le php.



### Exercice 04-03 [livre : chapitres 4]

Réalisez un formulaire d'inscription comprenant les champs suivants et renvoyant une page de confirmation.

Champs du formulaire :

- nom
- prénom
- Adresse
- NPA
- Localité
- Une date de naissance
- Un type de client (Privé, Entreprise, VIP) [menu]
- Le sexe [radio box]
- Une demande d'inscription à la newsletter [check box]

Page de confirmation :

- Un message de bienvenue avec la date

Aucune donnée n'est réellement enregistrée, puisque nous n'avons pas de base de données pour l'instant. Seul le visuel de la page est donc généré.

Optionnellement, commencez par une page qui ne vérifie pas que les champs obligatoires soient remplis.

The diagram illustrates the flow from a registration form to a confirmation page. On the left, a browser window titled 'Exo 03 : Formulaire d'inscription - Chr...' shows a form with the following fields: 'Nom\*' (John), 'Prénom\*' (Lennon), 'Adresse', 'NPA', 'Localité', 'Date de naissance\*' (9.8.1940), 'Type de client' (Privé), and radio buttons for 'Homme' and 'Femme'. There is also a checkbox for 'Je désire la newsletter' and an 'Envoyer' button. A blue arrow points to the right, where another browser window shows the confirmation page. The confirmation page has the title 'Exo 03 : Formulaire d'inscription - Chr...' and displays 'Bonjour Lennon John né(e) le 9.8.1940'. Below this, it says 'Nous vous remercions de votre inscription. Vous êtes un client de type privé. Inscription faite le 04.02.2016'.

**Formulaire d'inscription**

Nom\*: John  
Prénom\*: Lennon  
Adresse:  
NPA:  
Localité:  
Date de naissance\*: 9.8.1940  
Type de client: Privé  
● Homme ● Femme  
☐ Je désire la newsletter  
Envoyer

**Bonjour Lennon John né(e)  
le 9.8.1940**

Nous vous remercions de votre inscription.  
Vous êtes un client de type privé.  
Inscription faite le 04.02.2016

### Exercice 04-04

Réalisez un formulaire de vente qui génère une page de résumé de commande. Ce formulaire affiche une petite liste d'articles et de prix. Le résumé de commande affiche le prix par article.

**a)** Créez le site de manière statique. Les noms et les prix des articles seront donc dupliqués à plusieurs endroits.

**b)** Créez le site en vous basant sur un tableau associatif à deux dimensions.

- C'est un exercice difficile, prenez votre temps.
- Vous pouvez vous inspirer fortement de l'exercice 02-07. Seule la génération automatique de l'entête n'est pas à reprendre. Refaites l'exercice si nécessaire.
- Commencez par générer automatiquement le formulaire.
- Générez ensuite automatiquement le reçu.

**c)** Les articles non commandés (quantité 0) ne sont pas affichés dans le résumé.

Exemple d'exécution :

Votre formulaire de commande:

| Intitulé | Prix | Quantité voulue |
|----------|------|-----------------|
| Gants    | 3.5  | 10              |
| Gants XL | 8.5  |                 |
| Echarpe  | 14.5 | 1               |
| Fromage  | 15.5 | 1               |

Rétablir Commander



Résumé de votre commande:

| Intitulé             | Prix | Quantité voulue | Sous-Total |
|----------------------|------|-----------------|------------|
| Gants                | 3.5  | 10              | 35         |
| Echarpe              | 14.5 | 1               | 14.5       |
| Fromage              | 15.5 | 1               | 15.5       |
| Total de la commande |      |                 | 65         |

## Chapitre 05

### Session - \$\_SESSION[]

#### Exercice 05-01

Reprenez l'exercice 03-01 (compteur d'arbres), en utilisant cette fois les sessions pour mémoriser le nombre d'arbres actuels. Cela permet de revenir sur la page sans perdre l'état du compteur. Inversement, vous pouvez potentiellement rencontrer des difficultés à réinitialiser votre état depuis le navigateur. Dans ce cas, essayez d'effacer les *cookies* liés à votre page.

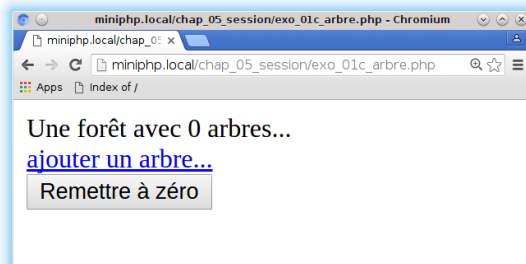
**a)** Avec un lien pour aller sur une page avec un arbre de plus.

Indice : Utilisez toujours `$_GET[]`, mais uniquement pour savoir quelle est l'opération en cours lorsque l'utilisateur arrive sur votre page.

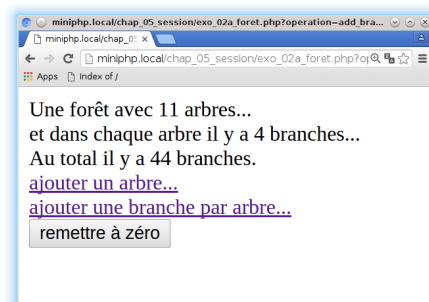
Optionnellement : Trouvez un moyen pour que le rechargement de page n'ait pas d'effet de bord (ajoute un nouvel arbre).

**b)** Ajoutez un lien "remettre à zéro", qui réinitialise le nombre d'arbres pour le remettre à 0.

**c)** Ajoutez un bouton "remettre à zéro" (Petit changement en HTML).



**d)** Ajoutez des branches comme dans l'exercice 03-02 (forêt).



**e)** Peut-on éviter qu'un utilisateur puisse "tricher" en changeant les codes de l'URL ? (Indice : Ici on utilise des boutons au lieu des liens)

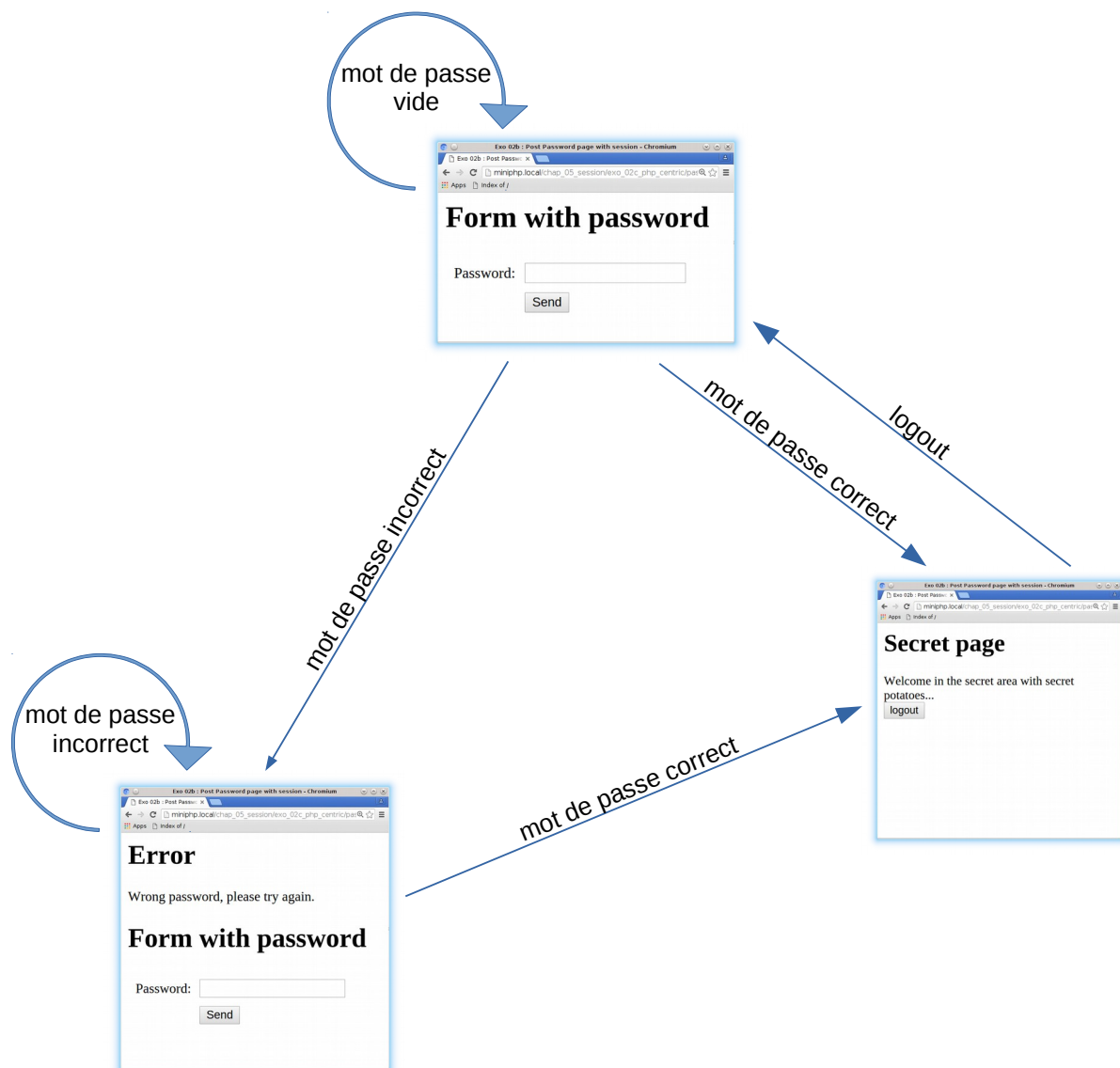
## Exercice 05-02

Reprenez l'exercice 04-02 (mot de passe), en utilisant cette fois les sessions pour mémoriser si l'utilisateur a entré le bon mot de passe.

- a) Créez une seule page PHP qui affiche dynamiquement les cas suivants :
- Un formulaire pour entrer le mot de passe, si aucun mot de passe n'a été entré.
  - Une page secrète si le bon mot de passe a été entré.

b) Affichez en plus un message d'erreur en dessus du formulaire si le mot de passe entré n'est pas le bon.

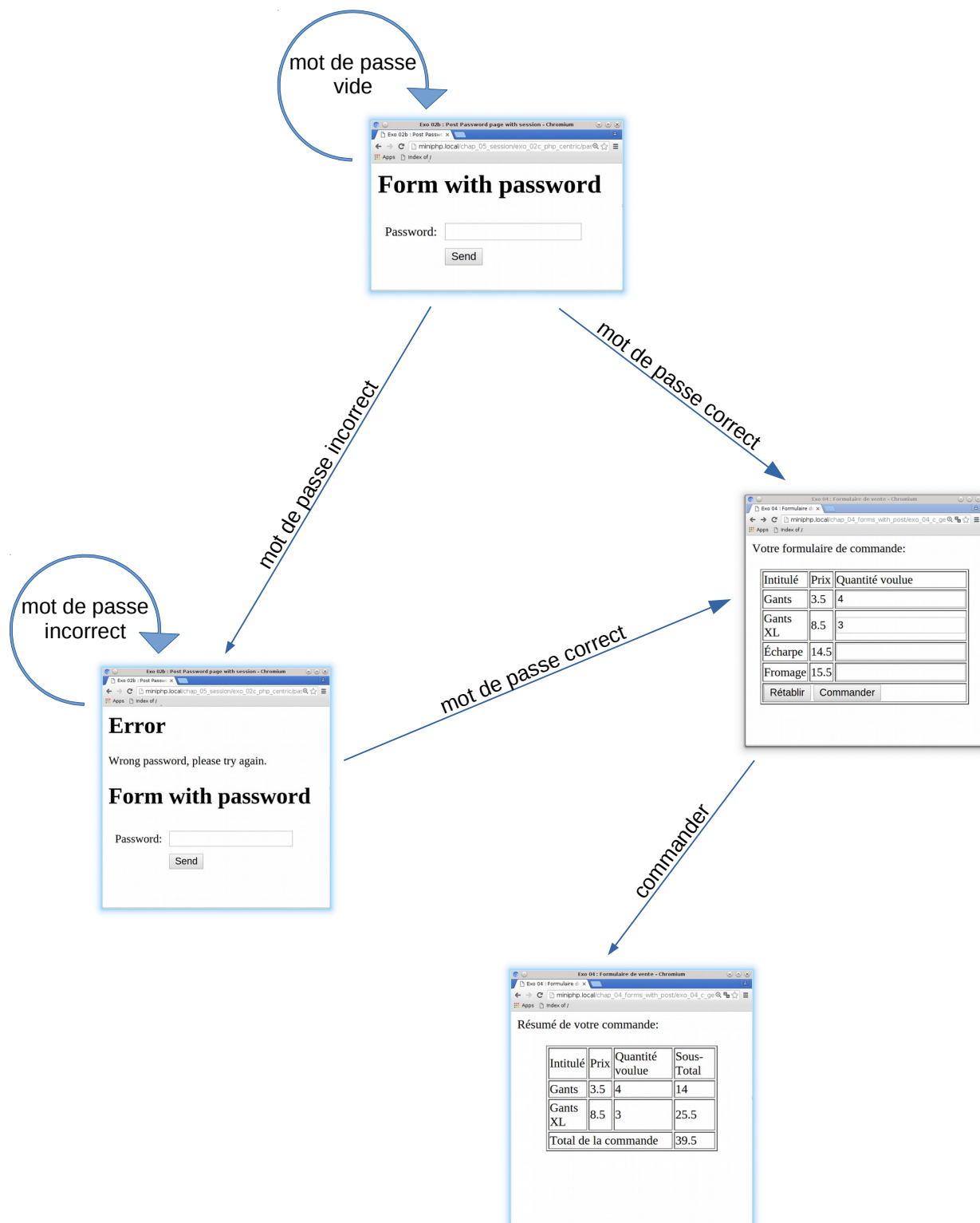
c) Ajoutez un bouton "logout" dans la page secrète pour effacer la session. Optionnellement assurez-vous que vous utilisez la méthode POST pour garantir une meilleur sécurité.





### Exercice 05-03

Cet exercice reprend l'exercice précédent 05-02 (mot de passe avec session).  
La page secrète doit-être remplacée par la page de magasin fait dans l'exercice 04-04.  
Optionnellement, ajoutez un bouton "logout" là où il fait sens.



## Chapitre 06

### XSS avec GET

#### Exercice 06-01

**a)** Créez une page web qui ajoute 1 arbre sur la page de l'exercice **05-01a** sans que l'utilisateur ne s'en rende compte.

Vous pouvez le faire simplement en HTML, ce qui est conseillé.

**b)** Ajoutez 1000 arbres cette fois (commencez par 10).

Utilisez PHP pour répéter l'opération.

**c)** Mettez le compteur à -1000 arbres sur la page de l'exercice **05-01d** (ou toute solution qui utilise un GET avec un nombre d'arbres à définir directement).

Vous pouvez le faire simplement en HTML, ce qui est conseillé.

## Chapitre 07

### Programmation orientée objet

#### Exercice 07-01

Gestion d'un objet Rectangle.

- a) Faire une *classe* Rectangle avec une fonction qui retourne la surface. Utilisez des termes en anglais pour les variables. Il n'y a pas de page affichable pour l'instant.
- b) Créez un formulaire qui permet de créer un Rectangle, et afficher ses propriétés (largeur, hauteur, surface) sur la même page.



Exo 02 : Post Password page (PHP centered) - Chromium

miniphp.local/chap\_06\_oop/Exo01b/index.php

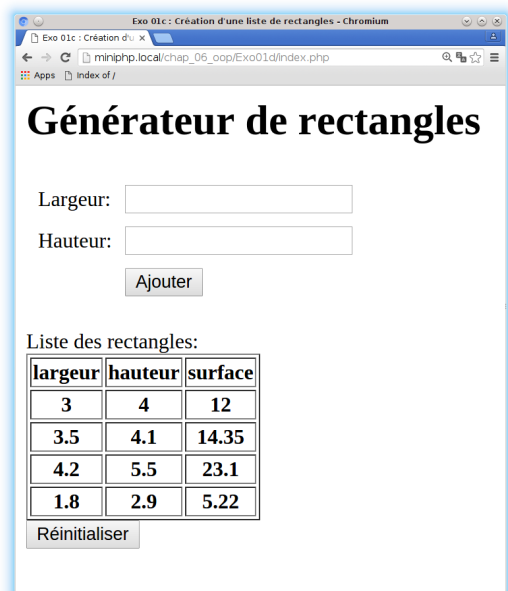
### Générateur de rectangles

Largeur:

Hauteur:

Un rectangle de largeur 111, de hauteur 2, et de surface 222.

- c) Créez un formulaire qui permet d'ajouter un Rectangle, et afficher la liste des rectangle et de ses propriétés (largeur, hauteur, surface) sur la même page, dans une table HTML.
- d) Ajoutez un bouton pour réinitialiser la liste.
- e) La gestion de l'affichage d'un rectangle doit être faite dans la classe *Rectangle*. Déplacez donc la partie du code qui génère les lignes de la table HTML des rectangles dans la classe *Rectangle*.



Exo 01c : Création d'une liste de rectangles - Chromium

miniphp.local/chap\_06\_oop/Exo01d/index.php

### Générateur de rectangles

Largeur:

Hauteur:

Liste des rectangles:

| largeur | hauteur | surface |
|---------|---------|---------|
| 3       | 4       | 12      |
| 3.5     | 4.1     | 14.35   |
| 4.2     | 5.5     | 23.1    |
| 1.8     | 2.9     | 5.22    |

## Exercice 07-02

Gestion d'un objet "Personne".

**a)** Faire une *classe* "Person" avec les attributs suivants :

- first\_name
- last\_name
- email
- zip\_code

Il n'y a pas de page affichable pour l'instant.

**b)** Créez une personne manuellement et affichez le résultat dans une page statique.

La personne:

| prénom | nom      | email                       | npa  |
|--------|----------|-----------------------------|------|
| John   | van Gosh | John.van.gosh@protonmail.ch | 1000 |

**c)** Créez un formulaire qui permet de créer une personne et d'en afficher le résultat.

Left side (Form):

Prénom:

Nom:

Email:

NPA:

Right side (Result):

La personne:

| prénom | nom      | email                       | npa  |
|--------|----------|-----------------------------|------|
| John   | van Gosh | John.van.gosh@protonmail.ch | 1000 |

**d)** Créez un formulaire qui permet d'ajouter une personne, et afficher la liste des personnes.

**e)** Ajoutez un bouton pour réinitialiser la liste.

**f)** La gestion de l'affichage d'une personne doit être faite dans la classe *Person*. Déplacez donc la partie du code qui génère les lignes de la table HTML des personnes dans la classe *Person*.

Prénom:

Nom:

Email:

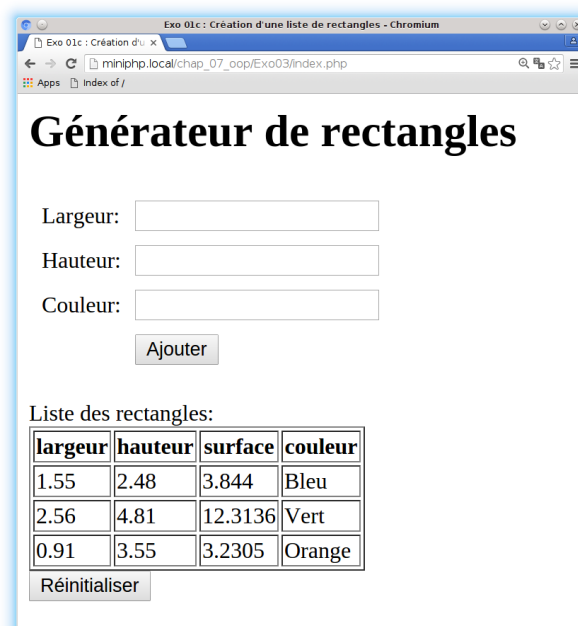
NPA:

| prénom   | nom      | email                  | npa  |
|----------|----------|------------------------|------|
| Jonathan | Gold     | jo.go@loin.io          | 1300 |
| Luméron  | Chermony | Lum.Chermony@youhaa.fr | 1555 |
| Pascale  | di Verde | pa.di.ve@yankex.tv     | 1200 |

## Exercice 07-03

Dans cet exercice nous reprenons l'exercice 07-01d de gestion de rectangles. Nous allons étendre le code actuel pour ajouter la possibilité de gérer des rectangles colorés. Pour ne pas modifier le code existant, nous allons étendre la classe *Rectangle* par la classe *ColorRectangle*, puis ajouter un champ "couleur" dans la création des rectangles.

Assurez-vous que la gestion de l'affichage des rectangles soit bien dans les classes *Rectangle* et *ColorRectangle*.



**Générateur de rectangles**

Largeur:

Hauteur:

Couleur:

Liste des rectangles:

| largeur | hauteur | surface | couleur |
|---------|---------|---------|---------|
| 1.55    | 2.48    | 3.844   | Bleu    |
| 2.56    | 4.81    | 12.3136 | Vert    |
| 0.91    | 3.55    | 3.2305  | Orange  |

## Chapitre 08

### Validation

#### Exercice 08-01

Dans cet exercice nous reprenons l'exercice 07-03, générateur de rectangle.

**a)** Vérifiez en plus que les dimensions du rectangle soient bien des nombres.

**b)** Vérifiez en plus que la couleur du rectangle soit une couleur. Vous pouvez limiter à "rouge", "vert" et "bleu".

## **Chapitre 09**

### **Expressions régulières**

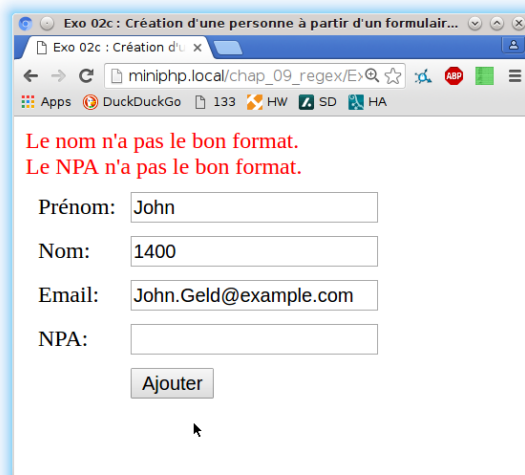
#### **Exercice 09-01**

Tester ces expressions avec un programme PHP :

1. Comment tester si un champ est rempli uniquement avec "Les Arches"?
2. Comment tester si un champ contient "Les Arches"?
3. Comment tester si un champ débute avec "Les Arches"?
4. Définir un motif permettant de tester une entrée dans un champ 'Prénom'. On doit s'assurer que celui-ci commence par une majuscule. Essayez ensuite de trouver une solution qui fonctionne avec un trait d'union (Par exemple "Jean-Marc").
5. Établir un motif qui correspond à ENUM("masculin", "féminin")
6. Établir un motif pour un numéro de téléphone de forme 021 987 65 43
7. Établir un motif permettant de tester une plaque minéralogique suisse
8. Établir un motif pour un URL. (Trouver une solution avec le 'www.' et une sans.)
9. Établir un motif pour une adresse e-mail.
10. Établir un motif pour une couleur composée (par exemple "rouge clair").

## Exercice 09-02

Dans cet exercice nous reprenons l'exercice **07-02f**, générateur de personnes. Vérifiez en plus chaque champ à l'aide d'une expression régulière.



Exo 02c : Création d'une personne à partir d'un formulaire...

miniphp.local/chap\_09\_regex/Exo02c/index.php

Le nom n'a pas le bon format.  
Le NPA n'a pas le bon format.

Prénom:

Nom:

Email:

NPA:

## Exercice 09-03

Dans cet exercice nous reprenons l'exercice **07-03**, générateur de rectangles. Vérifiez en plus chaque champ à l'aide d'une expression régulière. On n'accepte pas les nombres négatifs pour la largeur et la hauteur.



Exo 01c : Création d'une liste de rectangles - Chromium

miniphp.local/chap\_09\_regex/Exo03/index.php

### Générateur de rectangles

Largeur:

Hauteur:

Couleur:

La hauteur n'a pas le bon format.



## **Chapitre 10**

### **Validation client serveur**

#### **Exercice 10-01**

a) Quelle sont les langages utilisés pour la validation du côté client ?

---

---

---

---

---

b) Quelle sont les langages utilisés pour la validation du côté serveur ?

---

---

---

---

---

c) Quelle sont les avantage de procéder aux validations du côté du serveur plutôt que du côté du client ?

---

---

---

---

---

d) Quelle sont les avantage de procéder aux validations du côté du client plutôt que du côté du serveur ?

---

---

---

---

---

e) Citez les technologies qui permettent d'utiliser le même langage du côté du client et du serveur ?

---

---

---

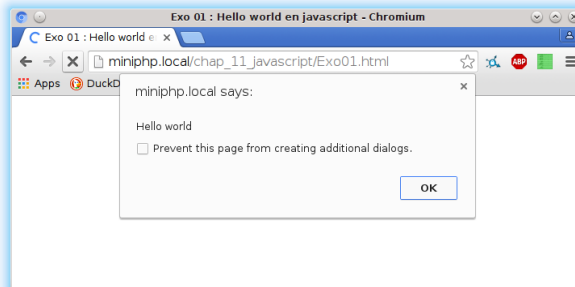
---

---

## Chapitre 11 javascript

### Exercice 11-01

Créez une page qui affiche le message "hello world" en javascript.

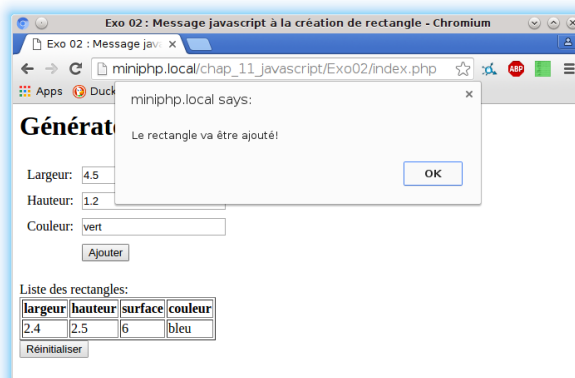


### Exercice 11-02

Reprenez l'exercice de générateur de rectangle couleur (Version Objet : Ex 07-03), et ajoutez un message "Le rectangle va être ajouté" lorsqu'un rectangle est ajouté.

- Créer une fonction JavaScript nommée *messageRectangleSent()* permettant d'afficher "Le rectangle va être ajouté". Créez cette fonction dans un fichier séparé.
- Ajouter un événement 'onClick' au bouton du formulaire et associer cet événement *onClick()* à la fonction *messageRectangleSent()*.

Pour l'instant il n'est pas possible d'utiliser un appel à *confirm()* pour pouvoir changer d'avis avant d'envoyer le rectangle en formulaire. On se contentera donc d'un message simple :



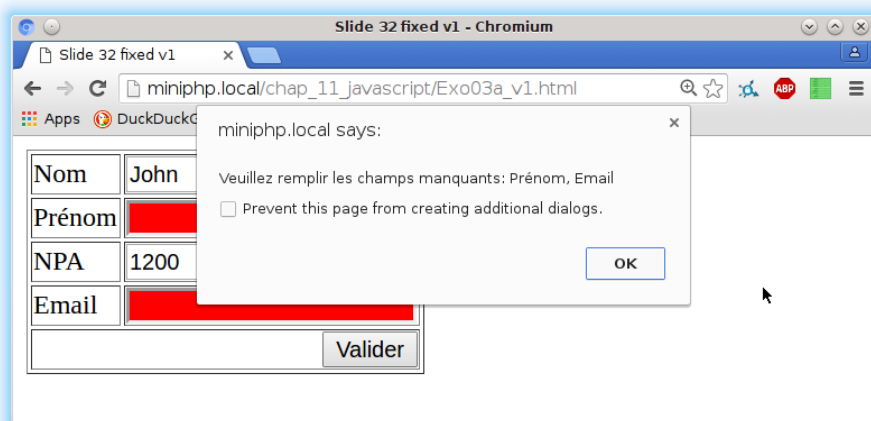
### Exercice 11-03

Reprenez l'exemple du *Slide 25* et ajoutez la validation client pour les autres champs.

**a)** Mettez les champs manquant en rouge et affichez un message d'erreur avec la liste des champs manquants.

Vérifiez qu'il est possible de changer les champs remplis et vide d'une fois à l'autre. Autrement dit, il doit-être possible que certains champs en rouge deviennent blanc et réciproquement.

**b)** [Avancé] Le code résultat se répète pour chaque champ. En utilisant les tableaux, trouvez comment générer automatiquement les différentes lignes, et le *javascript*. Cette partie est difficile, car il faudra générer du *javascript* en PHP.



## Exercice 11-04

Reprenez l'exercice de générateur de rectangle couleur (Ex 09-03), et ajoutez la validation du côté du client. Du côté client on ne vérifie que si le champ est rempli ou non.

Vérifiez que la validation du côté serveur fonctionne toujours en essayant d'insérer des données existantes mais non valides.

The screenshot shows a web browser window titled "Exo 04 : Création d'une liste de rectangles avec vérification client - Chromium". The address bar shows "miniphp.local/chap\_11\_javascript/Exo04/index.php". A message box from "miniphp.local" says: "miniphp.local says: Veuillez remplir les champs manquants: hauteur" with an "OK" button. The form has the following fields:

- Largeur: 1.5
- Hauteur: [redacted]
- Couleur: orange

Below the fields is an "Ajouter" button. At the bottom, there is a table titled "Liste des rectangles:" with the following data:

| largeur | hauteur | surface | couleur    |
|---------|---------|---------|------------|
| 2.3     | 1.7     | 3.91    | bleu clair |

There is also a "Réinitialiser" button at the bottom left.

## Exercice 11-05

Reprenez l'exercice de gestion de personnes (Ex 09-02), et ajoutez la validation du côté du client. Du côté client on ne vérifie que si le champ est rempli ou non.

Vérifiez que la validation du côté serveur fonctionne toujours en essayant d'insérer des données existantes mais non valides.

The screenshot shows a web browser window titled "Exo 05 : Création d'une personne à partir d'un formulaire avec vérification js - Chromium". The address bar shows "miniphp.local/chap\_11\_javascript/Exo05/index.php". A message box from "miniphp.local" says: "miniphp.local says: Veuillez remplir les champs manquants: email, NPA" with an "OK" button. The form has the following fields:

- Prénom: Jean-Ma
- Nom: van Geld
- Email: [redacted]
- NPA: [redacted]

Below the fields is an "Ajouter" button. At the bottom, there is a table with the following data:

| prénom | nom     | email                    | npa  |
|--------|---------|--------------------------|------|
| John   | Sixpack | John.Sixpack@example.com | 1200 |

There is also a "Réinitialiser" button at the bottom left.