

INITIATION À LA PROGRAMMATION

PARTIE 2 : PHP

Dans ce cours, nous aborderons progressivement et par la pratique les techniques transversales de programmation; par la même occasion, vous découvrirez plusieurs langages : dans la partie 1, le html et le css; dans la partie 2, le php; dans la partie 3, le pascal objet (Delphi); dans la partie 4, le Bash et le Python et dans la partie 5, le C. Après le cours, il ne vous restera plus qu'à choisir lequel de ces langages a votre préférence!

Tous droits réservés - 2012-2019 Christophe VARDON (christophe.vardon@ac-orleans-tours.fr)

1. LE LANGAGE PHP

1.1. Introduction

Le langage PHP est conçu pour créer des pages WEB "dynamiques" pour internet.

Le langage HTML est rarement employé seul; en effet il ne permet que de créer des page WEB statiques, c'est-à-dire qui affichent des informations *sans interaction* avec l'utilisateur. Le langage PHP, en complément d' HTML, permet de créer des pages WEB dynamiques, c'est-à-dire qui affichent des informations *en interaction avec l'utilisateur*. Les pages dynamiques sont de véritables programmes informatiques, qui produisent une page HTML « sur mesure », pour produire une interaction avec l'utilisateur. Ex : le site « PagesJaunes.fr »

1.2 IDE

L'IDE, est l'éditeur, l'outil avec lequel tu vas coder (rédiger) les pages PHP.

Les navigateurs FireFox/Edge/Chrome serviront à visualiser le résultat de ce code.

Certains **IDE** ont des fonctions supplémentaires : débogueur, suivi des variables, exécution pas à pas

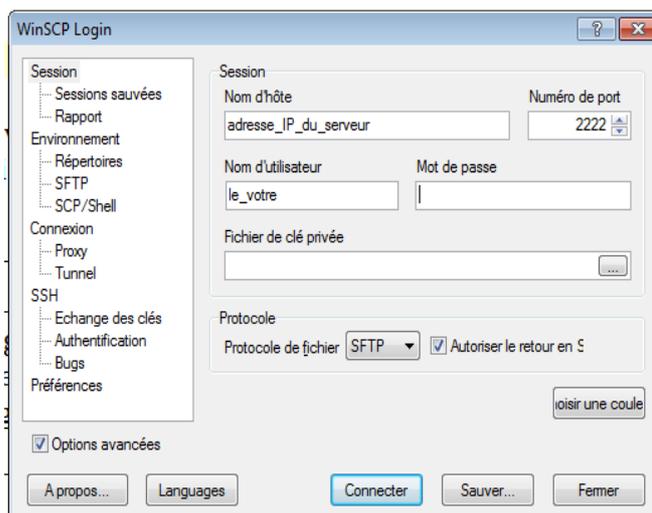
Tu peux utiliser **l'IDE** de ton choix et l'installer sur ton ordinateur ; si tu n'en connaît pas, tu peux commencer avec *Notepad++* (sous Windows) ou *Geany* (sous Linux) ; il en existe plein d'autres comme *SublimeText*, *Eclipse*,... Certains sont payants et certains seront limités à un certain type de langage.

Hébergement

Ton hébergement est disponible à l'adresse _____ dans le dossier /var/www/html/_____

ton nom d'utilisateur est _____ et le mot de passe associé est _____

tu y accèdera avec le client SFTP de ton choix (par exemple "*Winscp*" , « FieZilla », etc..) :



tu visualisera tes pages PHP à l'adresse :

http://_____

1.3 Premier programme : les bases

Dans Notepad+, crée un nouveau fichier nommé index.php , puis recopie le texte ci-contre dans ce fichier	<pre><html> <head> <title>Accueil du serveur GANDALF</title> </head> <body> <h1>ca marche!</h1> </body> </html></pre>	Puis Enregistre le fichier, télécharge-le sur ton hébergement, et ouvre-le depuis "Firefox"
---	---	---

Ce premier programme ne contient aucun code php, ajoutons-le :

Dans Notepad+, modifie le fichier index.php : selon le modèle ci-contre :	<pre><html> <head> <title>Accueil du serveur GANDALF</title> </head> <body> <?php echo " <h1>ca marche!</h1> " ; ?> </body> </html></pre>	Puis Enregistre le fichier, télécharge-le sur ton hébergement, et ouvre-le depuis "Firefox"
--	---	---

Comme tu le constates : **dans le code php, il faut encadrer les instructions HTML** par la commande "echo"

Info : ce qui différencie "index.php" de "index.html"

- le serveur web "apache" reconnaît l'extension ".php" et fait subir un traitement spécial au fichier pour interpréter le code php inclus dans la page.
- A l'intérieur du fichier, le code "php" commence avec la balise **<?php** et finit avec la balise **?>**

Évidemment, **si on utilise que la commande "echo", le php n'apporte rien de plus que le html !** nous allons ajouter quelques fonctions à notre programme "index.php"

Dans Notepad+, modifie le fichier index.php : selon le modèle ci-contre :	<pre><html> <head> <title>Accueil du serveur GANDALF</title> </head> <body> <?php echo " <h1>ca marche!</h1> <hr> nous sommes le : "; echo date ('d/m/Y'); echo " , il est : "; echo date ('H\h i \m\i\n s\s'); echo "<hr>"; ?> </body> </html></pre>	Puis Enregistre ce fichier, télécharge-le sur ton hébergement, et ouvre-le depuis "Firefox"
--	---	---

Exercice : cherche sur le net des informations sur la commande "date" du langage PHP

Nous avons affiché la date et l'heure, ce qui était impossible avec du "html" simple; mais il est temps de passer à quelque chose de plus consistant...

1.4 Gérer les réponses des formulaires HTML

Le langage **html** permet de créer des formulaires pour proposer des choix à l'internaute, mais pour exploiter la réponse de celui-ci, il faut un langage de programmation; PHP est tout-à-fait adapté à cela.

les formulaires en HTML - récupération de variables en PHP avec \$_GET.

Un formulaire en HTML se traduit par l'affichage d'un élément de saisie pour que l'utilisateur puisse entrer une information ou faire un choix

Exemple boîte de saisie:

Votre nom :

Exemple bouton radio :

Votre qualité :

- Monsieur
 Madame
 Mademoiselle

Exemple cases à cocher :

Vos sports préférés :

- Natation
 Cyclisme
 Ski

Exemple liste de sélection :

A quels jeux avez-vous déjà joué :

Exemple bouton submit :

Envoyer

Exemple de formulaire avec une boîte de saisie et un bouton submit :

contenu de index.php :

<code><?php</code>	Début de la section php (après <body>)
<code>echo "<form action='reponse.php'>";</code>	Début de formulaire; les données seront transmises à reponse.php
<code>echo "<input type='text' name='mareponse'>";</code>	Boîte de saisie mareponse
<code>echo "<input type='submit' value='Envoyer'>";</code>	Le bouton pour valider
<code>echo "</form>";</code>	Fin de formulaire
<code>?></code>	Fin de la section php (avant </body>)

Récupération de la saisie utilisateur dans le script « reponse.php »

<code><?php</code>	Début de la section php (après <body>)
<code>\$nom=\$_GET['mareponse'];</code>	On récupère mareponse dans \$nom
<code>echo "ton nom est : \$nom";</code>	On affiche la variable \$nom
<code>?></code>	Fin de la section php (avant </body>)

Remarque sur les noms de variables

les caractères accentués et les espaces sont interdits dans les noms de variable !!

Cette exercice t' apprend :

- ✓ à récupérer un paramètre fourni par l'utilisateur
- ✓ à passer un paramètre d'une page à une autre

Exercice : Tu dois créer deux page PHP; la première, **index.php**, comporte un formulaire avec une boite de saisie "Entres ton nom : " et un bouton submit "Envoyer"; quand l'utilisateur clique sur "Envoyer", une autre page "reponse.php" est appelée qui affiche "Le nom que tu as donné est : \$nom" (\$nom étant remplacé par la valeur fournie par l'utilisateur).

Ci-dessous la page index.php	Ci-dessous la page reponse.php
<p>Entres ton nom : <input type="text"/></p> <p>Envoyer</p>	<p>Le nom que tu as donné est : toto</p>

Correspond à la valeur fournie par l'utilisateur dans index.php

<p>Dans Notepad+, modifie le fichier index.php : selon le modèle ci-contre; à toi de compléter ce qui manque dans la section php</p>	<pre><html> <head> <title>test avec une boite de saisie et un bouton submit</title> </head> <body> <?php (à toi de compléter ce qui manque ici) ?> </body> </html></pre>	<p>Puis Enregistre ce fichier, télécharge-le sur ton hébergement, et ouvre-le depuis "Firefox"</p>
--	--	--

<p>Dans Notepad+, crée le fichier reponse.php : selon le modèle ci-contre :</p>	<pre><html> <head> <title>reponse au formulaire</title> </head> <body> <?php (à toi de compléter ce qui manque ici) ?> </body> </html></pre>	<p>Puis Enregistre ce fichier, télécharge-le sur ton hébergement, et ouvre-le depuis "Firefox"</p>
--	--	--

Cette exercice t' apprend :

- ✓ le rôle des fonctions fopen, fgets, fclose, explode

Exercice : En utilisant la recherche documentaire sur Internet, écris un exemple pour chacune des fonctions PHP présentées ci-dessous : fopen, fgets, fclose, explode.

fopen	fclose
Permet d'ouvrir un fichier en lecture ou en écriture	Permet de fermer de refermer un fichier (et donc de l'enregistrer)

fgets	explode
Permet de lire dans un fichier	Permet de séparer une chaîne de caractères en plusieurs champs

Cette exercice t' apprend :

- ✓ à ouvrir et exploiter un fichier présent sur le disque dur du serveur web (*fopen, fgets, fclose*)
- ✓ à exploiter des chaînes de caractères au format CSV (*explode*)
- ✓ à utiliser des boucles : avec la fonction *while(!feof(\$monfichier))*.

Crée un fichier que tu nommera « **test.txt** » ; dans ce fichier, écris le texte indiqué dans le cadre de droite =>

Tu dois créer *ou* copier ce fichier dans le même dossier que "index.php".

Montargis, Loiret,45200
Gien,Loiret,45500
Bourges,Cher,18000
Tours,Indre-et-Loire,37000

Note : la partie en violet est *une boucle*

<p>Dans Notepad+, recopie le fichier index.php : selon le modèle ci-contre;</p>	<pre><html> <head> <title>test de lecture d'un fichier texte</title> </head> <body> <?php echo "Base de données :"; \$monfichier = fopen("test.txt", "r"); while(!feof(\$monfichier)) { \$ville = fgets(\$monfichier); \$champs = explode(",", \$ville); echo "La ville de \$champs[0] est située dans le \$champs[1] ; son code postal est \$champs[2]
 "; } fclose(\$monfichier); ?> </body> </html></pre>	<p>Puis Enregistre ce fichier, télécharge-le sur ton hébergement, et ouvre-le depuis "Firefox"</p>
--	--	--

Comme tu peux le remarquer lors de l'exécution du programme, la partie (en violet) comprise entre "*while(!feof(\$monfichier)) {*" et "*}*" est exécutée autant de fois qu'il y a de lignes dans le fichier "test.txt" ; c'est ce qu'on appelle *une boucle*.

En anglais "*while*" signifie "*tant que..*".

On ne sort de la boucle que quand une condition de sortie est remplie.

Dans notre cas la condition est "*tant que la fin du fichier \$monfichier n'est pas atteinte*", donc quand la fin du fichier est atteinte, le programme est sorti de la boucle et a exécuté la suite du programme.

La **boucle ne traite donc qu'une ligne du fichier à la fois** et elle va se répéter autant de fois qu'il y a de lignes dans fichier.

Cette exercice t' apprend :

- ✓ à ouvrir et exploiter un fichier présent sur le disque dur du serveur web (*fopen, fgets, fclose*)
- ✓ à exploiter des chaînes de caractères au format CSV (*explode*)
- ✓ à utiliser des **boucles** : avec la fonction *while(!feof(\$monfichier))*.

Exercice : tu dois écrire une page web qui affiche le contenu d'une base donnée sous la forme d'un fichier CSV présent sur le disque dur du serveur.

Le fichier CSV sera appelé "users.txt" que tu dois créer/copier dans le même dossier que "index.php". Un exemple de fichier est donné ici à droite; les trois champs : nom, prénom, date de naissance sont séparés par des virgules ==>

```
jean,dupont,12/12/2000
benoit,martin,01/03/1995
frodon,sacquet,02/02/1866
yann,solo,25/10/2046
```

Les valeurs fournies par le fichier users.txt seront récupérées avec les fonctions *fopen, fgets, fclose, explode*

Ci-dessous, ce que doit afficher la page **index.php**

Bases de données :

Monsieur jean dupont est né le 12/12/2000

Monsieur benoit martin est né le 01/03/1995

Monsieur frodon sacquet est né le 02/02/1866

Monsieur frodon sacquet est né le 02/02/1866

Correspond aux données
contenues dans « users.txt »

Dans Notepad+,
modifie le fichier
index.php :
selon le modèle
ci-contre; à toi de
compléter ce qui
manque dans la
section **php**

```
<html>
<head>
<title>test avec une boite de saisie et un bouton submit</title>
</head>
<body>
<?php      (à toi de compléter ce qui manque ici)
```

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

Puis Enregistre
ce fichier,
télécharge-le
sur ton
hébergement,
et ouvre-le
depuis
"Firefox"

Correction de l'exercice :

index.php :

```
<?php
echo "Base de données :";
$monfichier = fopen("users.txt", "r");
while(!feof($monfichier)) {
    $compte = fgets($monfichier);
    $champs = explode(" ", $compte);
    echo "Monsieur $champs[0] $champs[1] est né le $champs[2] ";
    echo "<br>";
}
fclose($monfichier);
?>
```

Évaluation		Formulaire HTML	
Nom :	Prénom :	Note :	

1. Par quels balises un formulaire HTML est-il délimité ? (1 point)

2. Indiquez le code source correspondant à l'élément visuel suivant (2 points) :

Élément visuel	Code source correspondant
Prénom : <input type="text"/>	

3. Le code source suivant comporte une **erreur de syntaxe**; indiquez laquelle. (1 point)

```
echo "Indiquez votre prénom : <input type='text' name='prénom'>";
```

4. Le code source suivant comporte une **erreur de programmation**; indiquez laquelle. (1 point)

```
echo "<form action='reponse.php'>";
echo "Indiquez votre nom : <input type='text' name='nom'>";
echo "</form>";
```

5. Quelle fonction le code source suivant remplit-il ? (2 points)

```
$myname=$_GET['nom'];
```

6. Indiquez l'aspect visuel du résultat de ce code source (3 points)

Code source	Élément visuel correspondant
<pre>while(\$i<4) { echo \$i; echo "
"; \$i++; }</pre>	

Cette exercice t' apprend :

- ✓ A ouvrir et exploiter un fichier présent sur le disque dur du serveur web
- ✓ A utiliser des fonctions de **tests** (if ... else ...)

Exercice : tu dois écrire deux pages web qui serviront à vérifier si un utilisateur est bien autorisé à se connecter au site web.
La liste des utilisateurs sera contenue dans un fichier appelé "users.txt" que tu devra créer/copier dans le même dossier que "index.php". Un exemple de fichier est donné ici à droite

```
jean,dupont  
benoit,martin  
frodon,sacquet  
yann,solo
```

Les valeurs fournies par l'utilisateur devront donc être comparées aux valeurs contenues dans le fichier avant de l'autoriser ou non à entrer sur le site.

Ci-dessous la page index.php	Ci-dessous la page reponse.php
<p>Indiquez votre nom : <input type="text"/></p> <p>Indiquez votre prénom : <input type="text"/></p> <p>Envoyer</p>	<p>Merci jean dupont, vous êtes autorisé à entrer</p> <p>(ou bien)</p> <p>Désolé jean dupont, vous êtes inconnu</p> <p>Correspond à la valeur fournie par l'utilisateur dans index.php</p>

<p>Dans Notepad+, modifie le fichier index.php : selon le modèle ci-contre; à toi de compléter ce qui manque dans la section php</p>	<pre><html> <head> <title>test avec une boite de saisie et un bouton submit</title> </head> <body> <?php (à toi de compléter ce qui manque ici) ?> </body> </html></pre>	<p>Puis Enregistre ce fichier, télécharge-le sur ton hébergement, et ouvre-le depuis "Firefox"</p>
--	--	--

<p>Dans Notepad+, crée le fichier reponse.php : selon le modèle ci-contre :</p>	<pre><html> <head> <title>reponse au formulaire</title> </head> <body> <?php \$nom=\$_GET['prenom'] . "," . \$_GET['nom']; \$autorise=0; \$monfichier = fopen("users.txt", "r"); while(!feof(\$monfichier)) { \$compte = fgets(\$monfichier); if (\$compte==\$nom) {\$autorise=1;} } fclose(\$monfichier); if (\$autorise==1) {echo "Merci \$nom, vous êtes autorisé à entrer "}; else {echo "Désolé \$nom, vous êtes inconnu "}; ?> </body> </html></pre>	<p>Puis Enregistre ce fichier, télécharge-le sur ton hébergement, et ouvre-le depuis "Firefox"</p>
--	---	--

Correction de l'exercice :

index.php :

```
<?php
echo "<form action='reponse.php'>";
echo "Indiquez votre nom : <input type='text' name='nom'>";
echo "<br>";
echo "Indiquez votre prénom : <input type='text' name='prenom'>";
echo "<br>";
echo "<input type='submit' value='Envoyer'>";
echo "</form>";
?>
```

reponse.php :

```
<?php
$nom=$_GET['prenom'] . " ," . $_GET['nom'];
$autorise=0;
$monfichier = fopen("users.txt", "r");
while(!feof($monfichier)) {
    $compte = fgets($monfichier);
    if ($compte==$nom) {$autorise=1;}
}
fclose($monfichier);
if ($autorise==1) {echo "Merci $nom, vous êtes autorisé à entrer "};
else {echo "Désolé $nom, vous êtes inconnu "};
?>
```

Évaluation		Formulaire HTML	
Nom :	Prénom :	Note :	

Exercice : tu dois écrire deux pages web qui serviront à **vérifier si un utilisateur est bien autorisé à se connecter au site web.**

La liste des utilisateurs sera contenue dans un fichier appelé "users.txt" que vous devez créer/copier dans le même dossier que "index.php". Un exemple de fichier est donné ici à droite

```
jean,dupont
benoit,martin
frodon,sacquet
yann,solo
```

Les valeurs fournies par l'utilisateur devront donc être comparées aux valeurs contenues dans le fichier avant de l'autoriser ou non à entrer sur le site.

Ci-dessous la page index.php	Ci-dessous la page reponse.php
<p>Indiquez votre nom : <input type="text"/></p> <p>Indiquez votre prénom : <input type="text"/></p> <p><input type="button" value="Envoyer"/></p>	<p>Merci jean dupont, vous êtes autorisé à entrer</p> <p>(ou bien)</p> <p>Désolé jean dupont, vous êtes inconnu</p>

Correspond à la valeur fournie par l'utilisateur dans index.php

<p>Dans Notepad+, modifie le fichier index.php : selon le modèle ci-contre; à toi de compléter ce qui manque dans la section php</p>	<pre><html> <head> <title>test avec une boite de saisie et un bouton submit</title> </head> <body> <?php (à toi de compléter ce qui manque ici) ?> </body> </html></pre>	<p>Enregistre ce fichier, télécharge-le sur ton hébergement dans "Devoirs", et ouvre-le depuis "Firefox"</p>
--	--	---

<p>Dans Notepad+, crée le fichier reponse.php : selon le modèle ci-contre :</p>	<pre><html> <head> <title>reponse au formulaire</title> </head> <body> <?php (à toi de compléter ce qui manque ici) ?> </body> </html></pre>	<p>Puis Enregistre ce fichier, télécharge-le sur ton hébergement dans "Devoirs", et ouvre-le depuis "Firefox"</p>
--	--	--

Évaluation		Formulaire HTML	
Nom :	Prénom :	Note :	

Exercice : tu dois écrire deux pages web qui serviront à fournir à un utilisateur le programme TV en fonction de la chaîne qu'il a sélectionnée.

La liste des programme TV est fournie par le fichier "programme-tv.txt" qui est présent (dans le dossier "iso")

Ci-dessous la page index.php	Ci-dessous la page reponse.php												
<p>PROGRAMME TV D'AUJOURDHUI</p> <p>Choisissez une chaine :</p> <p> <input type="radio"/> TF1 <input type="radio"/> France 2 <input type="radio"/> France 3 <input type="radio"/> Canal+ <input type="radio"/> La 5ème </p> <p style="text-align: right; margin-right: 50px;"><input type="button" value="Envoyer"/></p>	<p>Programmes de TF1</p> <table border="1"> <thead> <tr> <th>Heure</th> <th>programme</th> </tr> </thead> <tbody> <tr> <td>00:11</td> <td>2 minutes pour la Syrie</td> </tr> <tr> <td>00:12</td> <td>Infô-Afrique</td> </tr> <tr> <td>00:23</td> <td>Lost : les disparus</td> </tr> <tr> <td>01:05</td> <td>Lost : les disparus</td> </tr> <tr> <td>01:47</td> <td>Lost : les disparus</td> </tr> </tbody> </table> <p style="text-align: center;">retour à la page d'accueil</p>	Heure	programme	00:11	2 minutes pour la Syrie	00:12	Infô-Afrique	00:23	Lost : les disparus	01:05	Lost : les disparus	01:47	Lost : les disparus
Heure	programme												
00:11	2 minutes pour la Syrie												
00:12	Infô-Afrique												
00:23	Lost : les disparus												
01:05	Lost : les disparus												
01:47	Lost : les disparus												

Correspond aux valeurs fournies par le fichier programme-tv.txt

Correspond à la valeur fournie par l'utilisateur dans index.php

<p>Dans Notepad+, modifie le fichier index.php : selon le modèle ci-contre; à toi de compléter ce qui manque dans la section php</p>	<pre><html> <head> <title>Programme TV aujourd'hui</title> </head> <body> <?php (à toi de compléter ce qui manque ici) ?> </body> </html></pre>	<p>Puis Enregistres ce fichier, télécharge-le sur ton hébergement, et ouvre-le depuis "Firefox"</p>
--	---	---

<p>Dans Notepad+, crée le fichier reponse.php : selon le modèle ci-contre :</p>	<pre><html> <head> <title>Le programme de <?php (à toi de compléter ce qui manque ici) ?> </title> </head> <body> <?php (à toi de compléter ce qui manque ici) ?> </body> </html></pre>	<p>Puis Enregistres ce fichier, télécharge-le sur ton hébergement, et ouvre-le depuis "Firefox"</p>
--	---	---

3. ANNEXE : QUESTIONNAIRE DE LECTURE

Remarque : les notions ci-dessous sont **des savoirs de bases** ; cela signifie que tu devras pouvoir répondre à ces questions sans t'aider d'une documentation.

1. Que fait le serveur Web quand on lui demande une page avec l'extension `'.php'` ? (page 3.)

2. Dans un fichier avec l'extension `'.php'`, il peut y avoir des parties écrites en code `'html'` et des parties écrites en code `'php'` ; pour les distinguer, le code `'php'` doit commencer et terminer par deux balises spécifiques : quelles sont ces balises ? (page 3.)

3. Dans un programme en `'php'`, on doit très souvent intégrer des commandes `'html'`, par exemples :

```
<p align='right'> ceci est un paragraphe </p>
```

Par quoi cette commande doit-elle être encadrée en `'php'` ; écris ci-dessous le code résultant :

4. Expliques ce qu'est un `'formulaire html'` : (page 4.)

5. Par quels balises un formulaire HTML est-il délimité ? (page 4.)

6. Dans un `'formulaire html'`, il y a une balise `'input'` qui doit être présente dans tous les cas, sans quoi le formulaire serait inutilisable ; de quel type de balise `'input'` s'agit-il ? (page 4.)

7. Quels sont les caractères interdits dans les noms de variable ? (faire l'exercice 1. en annexe 4)

8. Combien y-a-t-il de variables dans le code de la page 4 ; comment ces variables sont-elles nommées ?
Même question pour le code de la page 7.

9. Quelle est la fonction `'php'` qui permet de faire un test ? (page 11.) ; donnes un exemple :

10. Qu'est-ce qu'une boucle de programme ? Qu'est-ce que la `'condition de sortie'` ? (page 7.)

4. ANNEXE : EXERCICES

1. Dans la liste suivantes de 10 variables, rayer les noms de variable incorrects :

réponse	ma_reponse	nom	prénom	i
age	monâge	mon nom	a	adresse

2. Regardes la boucle ci-dessous ; quelle est sa condition de sortie ?

```
while( a < 6 ) { a++ ; }
```

3. Regardes la boucle ci-dessus ; qu'est-ce qui va s'afficher dans la page web ?

4. Complète le code pour que la valeur de 'a' s'affiche à chaque itération de la boucle :

```
while( a < 6 ) { a++ ; _____ }
```

5. Une fois le code complété, qu'est-ce qui va s'afficher dans la page web ?