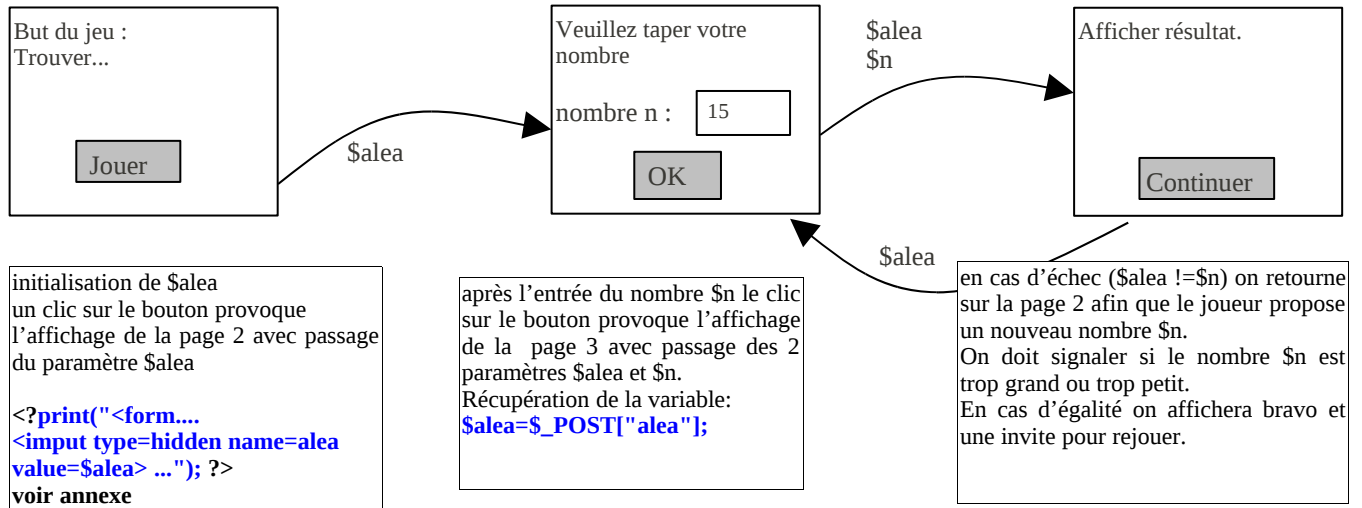


- Objectifs :**
- s'initier à la programmation **côté serveur**
 - interfacier les langages Html et Php
 - réaliser des pages web interactives

But : Réaliser un jeu où le joueur doit deviner un nombre (\$alea) compris entre 0 et 100. Le nombre à deviner est aléatoirement initialisé à chaque partie.

On propose de travailler avec 3 nouvelles pages; la première sera accessible depuis un lien placé dans le menu des TP précédents.



On développe côté serveur : vos scripts doivent donc être enregistrés dans votre compte sur la machine serveur. La première page recherchée par le serveur HTTP est `index.htm`, `index.html` ou `index.php`

1) Sans variable stockée sur la machine cliente

- 1.1) Réaliser la première page php et s'assurer du lien vers une seconde page avec passage de l'argument.
- 1.2) Produire le code de la seconde page et vérifier le passage des 2 paramètres vers une troisième page.
- 1.3) Donner l'algorithme correspondant à la page 3 puis le coder.
- 1.4) Modifier la page 3 afin d'indiquer au joueur en combien de fois il a trouvé le nombre \$alea.
La page 1 devra donc initialiser une variable *compteur* à 0 et la page 2 ou 3 l'incrémenter.

2) Transparence du paramètre avec *cookie* ou *session*

- 2.1) Pour tricher le joueur peut afficher le source et lire \$alea. On décide donc de stocker \$alea dans un *cookie*.
Les pages 2 et 3 doivent donc aller chercher la variable dans le *cookie*.
Vous pouvez vérifier la présence du cookie sur votre station voir l'aide sur la fonction `setcookie ()`
- 2.2) Le cookie étant sur la machine *cliente*, le joueur peut l'afficher; on décide d'ouvrir une session et de mémoriser \$alea comme variable de session : les variables de session sont enregistrées sur la machine *serveur*.
Dans chaque script lisant une variable de session, il est nécessaire de continuer la session avec `session_start`.
D'autre part les fonctions de session doivent être appelées avant tout envoi d'entête.
voir l'aide sur les sessions et les fonctions `session_start ()`, `session_register ()` et `session_destroy ()`.

3) Vérification du formulaire de saisie en javascript

- Réaliser une fonction javascript qui envoie le nombre n au serveur seulement si le champ n'est pas vide.
Justifier le choix du javascript pour cette fonction.

Critères d'évaluation :

- Degré d'autonomie dans la recherche d'informations et dans la mise au point
- Avancement du travail
- Respect de l'algorithme
- Qualité du programme et du compte-rendu.

AIDE avec le navigateur dans l'URL <http://s.flament.iris.rostand/aide> (proxy 172.30.200.254 port 3128)

Annexe

Contenu du premier fichier index.php.

- On retrouve la fonction aleatoire vue en cours.

Cette partie est située en début de fichier pour faciliter les questions avec les cookies et les sessions qui doivent être envoyés dans l'en-tête donc avant tout caractère.

- Pour afficher ou envoyer du HTML on utilise echo " salut ", echo 'salut', ou print(" toto ");

Dans le source, on distingue 3 variantes pour utiliser echo. Un seul echo pour le formulaire est possible.

- Le formulaire doit transmettre à la page " saisie.php " la valeur d'une variable php, on peut donc placer tout le formulaire dans le code PHP avec un echo.

```
<?php
```

```
function aleatoire($min,$max)
{
    $temps=explode(" ",microtime());
    $usec=$temps[0];
    srand($usec*1000000);
    $mystere = rand($min,$max);
    return $mystere;
}
```

```
$alea=aleatoire(0,100);
```

```
?>
```

```
<HTML>
```

```
<HEAD><TITLE> Rostand Jeux </TITLE> </HEAD>
```

```
<BODY BGCOLOR=white >
```

```
But: Trouver un nombre compris <br> entre 0 et 100
```

```
<br>
```

```
<?
```

```
echo '<form action="saisie.php" method=post>';
echo "<input type=hidden name=\"alea\" value=$alea>";
echo "<input type=hidden name=compteur value=0>";
echo "<input type=submit value=Jouer>";
echo "</form>";
```

```
?>
```

```
</BODY>
```

```
</HTML>
```

La seconde page " saisie.php " affiche un formulaire permettant de proposer un nombre.

Dans un premier temps il est conseillé d'afficher les variables dans chaque page.

Avant le traitement d'une variable, il faut la récupérer dans le tableau \$_POST, \$_GET, \$_COOKIE selon la méthode utilisée.

```
$alea=$_POST["alea"];
```

```
echo " variable: alea=$alea ";
```

Démo sur:

<http://btsirinfo.free.fr/jeu1/jeu1exo1.php>

```
Enregistrer un cookie avec une durée de vie d'une heure:
```

```
setcookie(" alea ",$alea,time()+3600);
```

```
?>
```

```
<!DOCTYPE HTML PUBLIC.....
```

```
lire un cookie $a=$_COOKIE[" alea "];
```

```
Enregistrer une variable de session:
```

```
session_start();
```

```
$_SESSION["alea"]=$alea;
```

```
?>
```

```
<!DOCTYPE HTML PUBLIC....
```

```
Lire une variable de session:
```

```
session_start();
```

```
$alea=$_SESSION[" alea "];
```

```
?>
```

```
<!DOCTYPE HTML PUBLIC....
```