Programmation Web Serveur

Bilan intermédiaire 1 - Sessions

D'après les cours de Philippe Renevier

Fabien Givors

Université de Nice Sophia Antipolis

Département Informatique

fabien.givors@unice.fr

Qu’avons-nous vu jusqu’à présent

BILAN INTERMÉDIAIRE

EN RÉSUMÉ...

L'utilisateur veut voir la page http://www.i3s.unice.fr/~fgivors/

1. Tape l'adresse dans son navigateur, valide

2. Le navigateur fait une requête DNS pour résoudre le nom
   `www.i3s.unice.fr`

3. Le serveur DNS répond au navigateur que ce nom a l'IP
   `134.59.130.2`

4. Le navigateur fait une requête HTTP à `134.59.130.2` pour demander la page `~fgivors/`

5. Le serveur
   a) reçoit la demande
   b) identifie le script à exécuter en fonction de la page demandée
   c) exécute le script PHP afin de générer du code HTML
   d) renvoie le code HTML généré au navigateur via HTTP

6. Le navigateur reçoit le code HTML et l'affiche

EXEMPLE

```
<?php
// calcul préalable
$date = "<p>Nous sommes le";
$date = $date . date("d/m/Y"). "</p>";
?>
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD
   XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xh
   tml11.dtd">

<html
   xmlns="http://www.w3.org/1999/xhtml"
   >

<head><title>HTML avec PHP</title>
</head>
<body>
<h1>HTML + PHP</h1>
<?php
   echo $date;
?>

</body></html>
```

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD
   XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xh
   tml11.dtd">

<html
   xmlns="http://www.w3.org/1999/xhtml"
   >

<head><title>HTML avec PHP</title>
</head>
<body>
<h1>HTML + PHP</h1>
<p>Nous sommes le 02/10/2014</p>

</body></html>
```

LES VARIABLES

- Typage dynamique ; non déclaratif
- Pour connaître le type :
  - gettype( ) (retourne une chaîne)
  - Fonctions qui retournent un booléen : is_array( ), is_float( ), is_int( ), is_object( ) et is_string( ).
- isset : pour savoir si elle(s) existe(nt) :
  - isset
    - retourne vrai ou faux
    - permet de savoir si la variable « nom » existe
  - isset($variable)
  - isset($v1, $v2, $v3)
- unset pour détruire une variable

CHAÎNES DE CARACTÈRES

- `$chaine = "une chaîne de caractère";`

- `strlen($chaine)` pour connaître la taille
- `string` **`substr`** `( string $string , int $start [, int $length ] )` pour avoir une partie de la chaîne (entre `$start` et `$start+$length`)
- `int` **`strpos`** `( string $string , mixed $recherche)` pour rechercher à l'intérieur d'une chaîne
- **`.`** pour concaténer (ajouter deux chaînes)

- Fonction `echo` pour afficher
- Intégration de variables dans les chaînes (remplacées par leur valeur)

```
echo "Du $boisson, du pain et du fromage!";
echo "Il a goûté plusieurs {$boisson}s";
```

- Des caractères spéciaux : `\n` (retour à ligne) ; `\t` (tabulation) ; etc.
  - Non vus sur le HTML « rendu » (dans le navigateur)

OPÉRATIONS

- Opérateurs arithmétiques :
  - `$a + $b` // Addition de $a et $b
  - `$a – $b` // Soustraction de $b `a $a
  - `$a * $b` // Multiplication de $a et $b
  - `$a / $b` // Division de $a par $b
  - `$a % $b` // $a modulo $b (reste de la division de $a par $b)
  - `$i++;` // incrémenter $i (ajouter 1)
  - `$j = ++$i;` // incrémenter $i puis affecter cette valeur à $j
  - `$k = $i++;` // affecter la valeur de $i à $k puis incrémenter $i
  - `$k--;` // décrémenter $k
- Opérateur de concaténation des chaînes:
  - `$c1 = "Bonjour ";`
  - `$c2 = " le monde";`
  - `$c = $c1." tout ".$c2;` //donne « Bonjour tout le monde » dans $c
  - `$c .= " !";` // donne « Bonjour tout le monde !» dans $c, équivalent à $c = $c . " ! ";

FONCTIONS DE PHP

- PHP dispose de nombreuses fonctions et structures standards.
- http://www.php.net
- Fonctions vues
  - Manipulation de chaînes et de tableaux,
  - Calculs : floor
  - Lister des fichiers : glob
  - Etc.

TABLEAUX

- Un tableau est une suite de valeurs référencées par une unique variable.
  - PHP gère dynamiquement la taille des tableaux
  - Les tableaux en PHP peuvent être soit *indicés* soit *associatifs*.
- Créer un tableau (méthodes équivalentes)

```
// gestion à la main des indices // gestion par php des indices
$tab[0] = " élément 1 ";         $tab[] = "élément 1 "; // $tab[0]
$tab[1] =  "élément 2 ";         $tab[] = "élément 2 "; // $tab[1]
$tab[2] = 120;                   $tab[] = 120;          // $tab[2]

// initialisation avec la fonction array
$tab = array ("élément 1 ", "élément 1 ", 120);
// avec des clefs (tableau associatif)
$mes = array ("Vertigo" => "Hitchcock","Sacrifice" => "Tarkovski",
                          "Alien" => "Scott");
```

FONCTIONS LIÉES AUX TABLEAUX

- Fonction **count** pour savoir le nombre d'élément dans un tableau
- Fonction **sort** pour trier

- Tableau dans une chaîne : {$...}

```
$chaine = "Une banane est
{$fruits["banane"]}.";
```

CONDITION : IF / ELSE

- La structure la plus courante est le if ... else.

```
if (expression) // Bloc { } si expression est vraie.
else            // Bloc { } si expression est fausse (optionnel)
// Ici le script continue.
```

- Exemple

```
$marquePluriel = ""; // chaîne vide
if ($nbJour > 1)
{
  $marquePluriel = "s"; // s'il y a plus qu'un jour
}
$texte = "<p>Il y a $nbJour jour$marquePluriel</p>";
```

  - Si $NbJour vaut 1, $texte vaut " Il y a 1 jour "
  - Si $NbJour vaut 2, $texte vaut " Il y a 2 jours "

EXPRESSIONS LOGIQUES ET COMPARAISONS

```
($a && $b)     // ET logique.
($a || $b )    // OU logique.
($a xor $b)    // Ou exclusif.
(!$a)          // NOT.

($a == $b)     // Vrai si $a est égal à $b.
($a != $b)     // Vrai si $a est différent de $b.
($a < $b)      // Vrai si $a est inférieur à $b.
($a > $b)      // Vrai si $a est supérieur à $b.
($a <= $b)     // Vrai si $a est inférieur ou égal à $b.
($a >= $b)     // Vrai si $a est supérieur ou égal à $b.

if( ($a < $b) && ($a < $c) ) {  /* … */ }
```

FOREACH (1/2)

- C'est un moyen simple de passer en revue un tableau
- Deux syntaxes
- À chaque itération, la valeur de l'élément courant est assignée à *$value* et le pointeur interne à la liste (tableau) est avancé d'un élément
- ce qui fait qu'à la prochaine itération, on accédera à l'élément suivant.

```
foreach ($tableau as $value) {
    /*
   $value prend successivement toutes les valeurs
contenues dans $tableau
   toutes les instructions faites dans ce bloc sont
exécutées pour
   chaque valeur
   */
}
```

FOREACH (2/2) : EXEMPLE

- Supposons que le script php soit exécuté dans le dossier « politique »

```
// listes des fichiers .news dans le dossier courant
$fichiers = glob("*.news");
$texte = "<p>Le dossier contient les fichiers : ";
foreach($fichiers as $f)
// $f prendra successivement toutes les valeurs (ici
// les noms des fichiers) contenues dans la
// liste/tableau $fichiers
{
   $texte = $texte . "$f , ";
}
```

DÉROULEMENT DE L'EXEMPLE


```
foreach($fichiers as $f)
{
    $texte = $texte . "$f, ";
}
```

Avant la boucle : $texte vaut **"<p>Le dossier contient les fichiers : "**

1ère itération : $f vaut "01.news" ; il reste à parcourir 02.news, 03.news, 04.news et 05.news. $texte vaut :

"<p>Le dossier contient les fichiers : 01.news, "

2ème itération : $f vaut "02.news" ; il reste à parcourir 03.news, 04.news et 05.news. $texte vaut :

"<p>Le dossier contient les fichiers : 01.news, 02.news, "

...

5ième itération: $f vaut "05.news" ; il ne reste plus rien à parcourir

$texte vaut **"<p>Le dossier contient les fichiers : 01.news, 02.news, 03.news, 04.news, 05.news, "**

Fin de la boucle : $texte vaut

"<p>Le dossier contient les fichiers : 01.news, 02.news, 03.news, 04.news, 05.news, "

FOREACH ET INDICE / CLEF

```
foreach ($tableau as $clef => $valeur) {
    /*
   $valeur prend successivement toutes les valeurs
contenues dans $tableau

   Dans le même temps, $clef correspond à l'indice (ou
la clef) de cette valeur dans le tableau

   Toutes les instructions faites dans ce bloc sont
exécutées pour chaque couple (clef, valeur)
   */
}
```

PASSAGE DE PARAMÈTRE DANS L'URL

- Passage de valeur d'une page à une autre
- Ajout à la fin de l'url
  - http://serveur.domaine/chemin/page?variable=valeur
  - Une variable dans une url se définit par
    - Un nom
    - Une valeur (après égal)
  - Plusieurs variables possibles : séparation par des & entre les variables

RÉCUPÉRER LA VALEUR DANS PHP

- $_GET
  - Variable super globale
  - Un tableau associatif des valeurs passées au script courant via les paramètres d'URL.
- Les clefs sont les noms des variables de l'url
- Les valeurs sont définies après le signe = (et avant le &)

```
$val = "valeur par défaut";
if ( isset($_GET["variable"])  )
{
  $val = $_GET["variable"];
  // faire des tests sur les valeurs…
  // conversion
}
```

RETOUR SUR LE TP 2

DÉROULEMENT DU TP : PAR ITÉRATION
1. ON LIT UN FICHIER « BRUT »

```php
/** permiere partie : liste des fichiers dans le dossier politique **/
$fichiers = glob("news/politique/*.*");

// indice courant
$indice = 0;

// nombre de fichier
$nbFichiers = count($fichiers);

// message en cas d'absence de fichier
$article = "<div class='article'>il n'y a pas d'article</div>";

// $fichiers contient la liste des fichiers sous forme de tableau
// on affiche l'article courant

if (($indice >= 0) && ($indice < $nbFichiers))
    {
    $article = "<div class='article'>".file_get_contents($fichiers[$indice])."</div>";
    }
else {
    $article = "<div class='article'>article inexistant</div>";
}
```

2. ON DÉSIGNE UN FICHIER (CATÉGORIE FIXE)

```php
/** permiere partie : liste des fichiers dans le dossier politique **/
$fichiers = glob("news/politique/*.*");

// indice courant
$indice = 0;

// nombre de fichier
$nbFichiers = count($fichiers);

// recuperation s'il y a de l'article en cours
if (isset($_GET["indice"])) {
    $indice = intval($_GET["indice"]);
}
```

3. UN PAS VERS LA NAVIGATION : UN LIEN SUIVANT (TOUJOURS LE MÊME LIEN, ENFIN PRESQUE)

```php
// message en cas d'absence de fichier
$article = "<div class='article'>il n'y a pas d'article</div>";
$suivant = "";

// $fichiers contient la liste des fichiers sous forme de tableau
// on affiche l'article courant

if (($indice >= 0) && ($indice < $nbFichiers))
    {
    $article = "<div class='article'>".file_get_contents($fichiers[$indice])."</div>";

    // peut-il y avoir un article suivant ?
    if ($nbFichiers > 1)
        {
        $indice_suivant = ($indice + 1) % $nbFichiers;
        $suivant = "<div class=\"suivant\"><a href=\"?indice=$indice_suivant\">article
        suivant</a></div>";
        }
    }
else {
    $article = "<div class='article'>article inexistant</div>";
}
```

4, « PERSONNALISATION » DU LIEN

```
// peut-il y avoir un article suivant ?
if ($nbFichiers > 1)
    {
    $indice_suivant = ($indice + 1) % $nbFichiers;

    // nom du lien = nom du fichier
    // on verra plus tard avec la fonction file...
    $nomarticle = explode("/",$fichiers[$indice_suivant]);
    $nomarticle = $nomarticle[count($nomarticle)-1];

    $suivant = "<div class=\"suivant\"><a href=\"?indice=$indice_suivant\">$nomarticle
    </a></div>";
    }
```

5. AJOUT D'UN LIEN « PRÉCÉDENT »

- $precedent initialisé à ""

```
// peut-il y avoir aussi un article precedent ?
if ($nbFichiers > 2)
    {
    // on reboucle a la fin...
    $indice_precedent = ($indice - 1 ) % $nbFichiers;
    // le resultat du % peut etre < 0... securite necessaire
    if ($indice_precedent < 0) $indice_precedent += $nbFichiers;

    // nom du lien = nom du fichier
    // on verra plus tard avec la fonction file...
    $nomarticle = explode("/",$fichiers[$indice_precedent]);
    $nomarticle = $nomarticle[count($nomarticle)-1];

    $precedent = "<div class=\"precedent\"><a href=\"?indice=$indice_precedent\">$nomarticle
    </a></div>";
    }
}
```

Sauvegarde à court terme

SESSION

NOTION DE SESSION

- Le support des sessions de PHP est un moyen de préserver des données entre plusieurs accès. Cela vous permet de créer des applications personnalisées.
- Chaque visiteur accédant à votre page web se voit assigner un identifiant unique, appelé "identifiant de session". Il peut être stocké soit dans un cookie, soit propagé dans l'URL.
- Lorsqu'un visiteur accède à votre site, PHP va vérifier sur demande explicite avec `session_start( )` s'il existe une session du même nom. Si c'est le cas, l'environnement précédemment sauvé sera recréé.
- `session_start( )`
  - La gestion par défaut du numéro de session (identifiant) passe par les cookies...
  - donc `session_start( )` doit être appelé avant toutes sorties

SESSION : $_SESSION

- $_SESSION : tableau contenant toutes les variables de session
- Affection = création ou mise à jour
  - `$_SESSION["style"]="blue.css"` : crée une variable de session « style » qui vaut « blue.css »
  - On peut mettre aussi des tableaux dans une session

```
$_SESSION["pagesvues"]=array("news/France/01.news" =>  true,
                             "news/France/02.news"  => false);
```

- Utilisation = utilisation de la variable

```
echo " <link rel=\"stylesheet\" type=\"text/css\"
href=\"{$_SESSION["style"]}\" />";
if ( $_SESSION["pagesvues"]["news/France/02.news"] ) { /* … */
}
```

- Test d'existence : isset
  - `if  ( isset($_SESSION["style"]) )` …
- Effacement : unset
  - `unset($_SESSION["style"])`

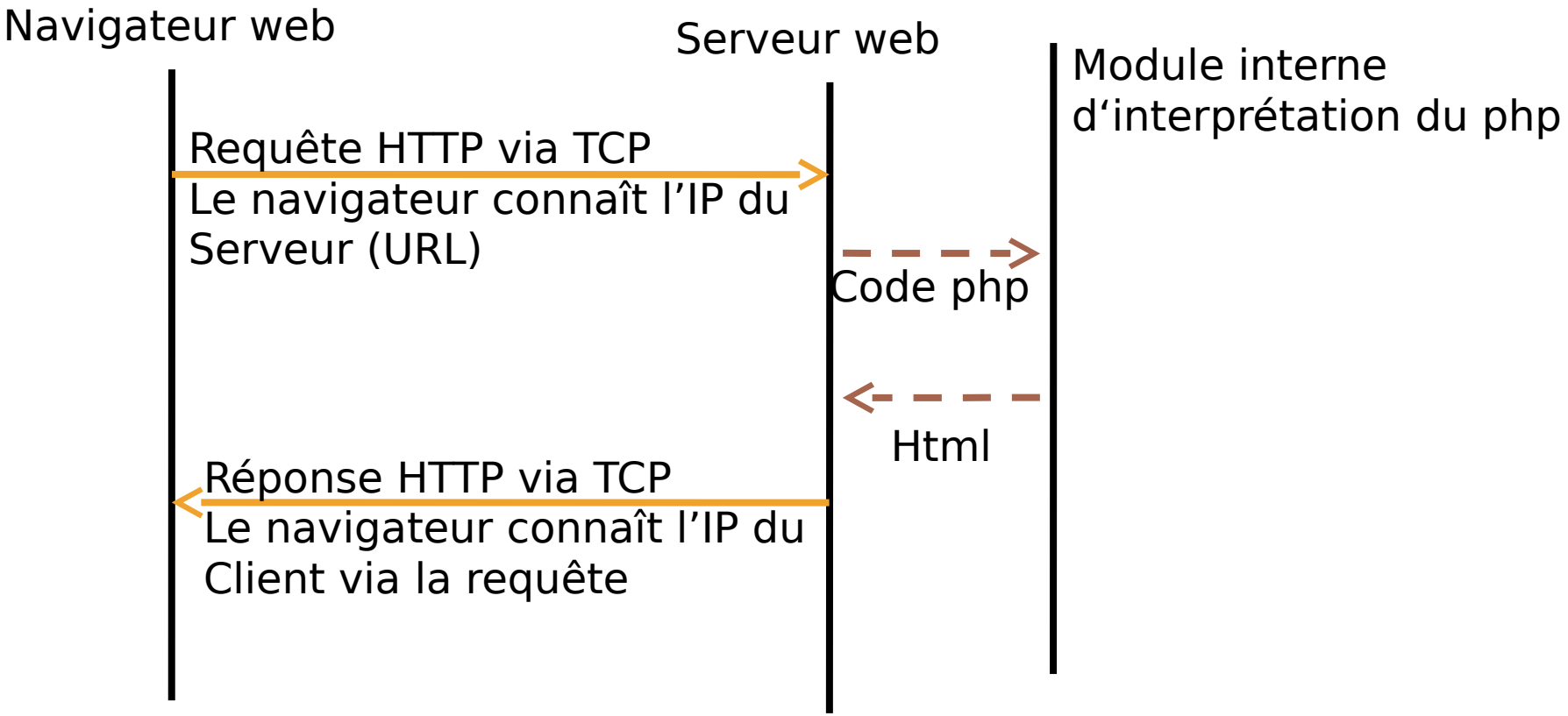
PERSONNALISATION DES SESSIONS SUR UN MÊME SERVEUR

- Fonction session_name
  - SANS PARAMETRE : permet de savoir le nom de la session courante
  - Avec une chaîne de caractères (au moins une lettre) en paramètre ET AVANT session_start : permet de commencer une session spécifique
- Vous permet de faire des sessions distinctes sur www-mips…
- A mettre avant toutes ouvertures de session !
  - Sinon : une seule session pour tous les sites d’un serveur…

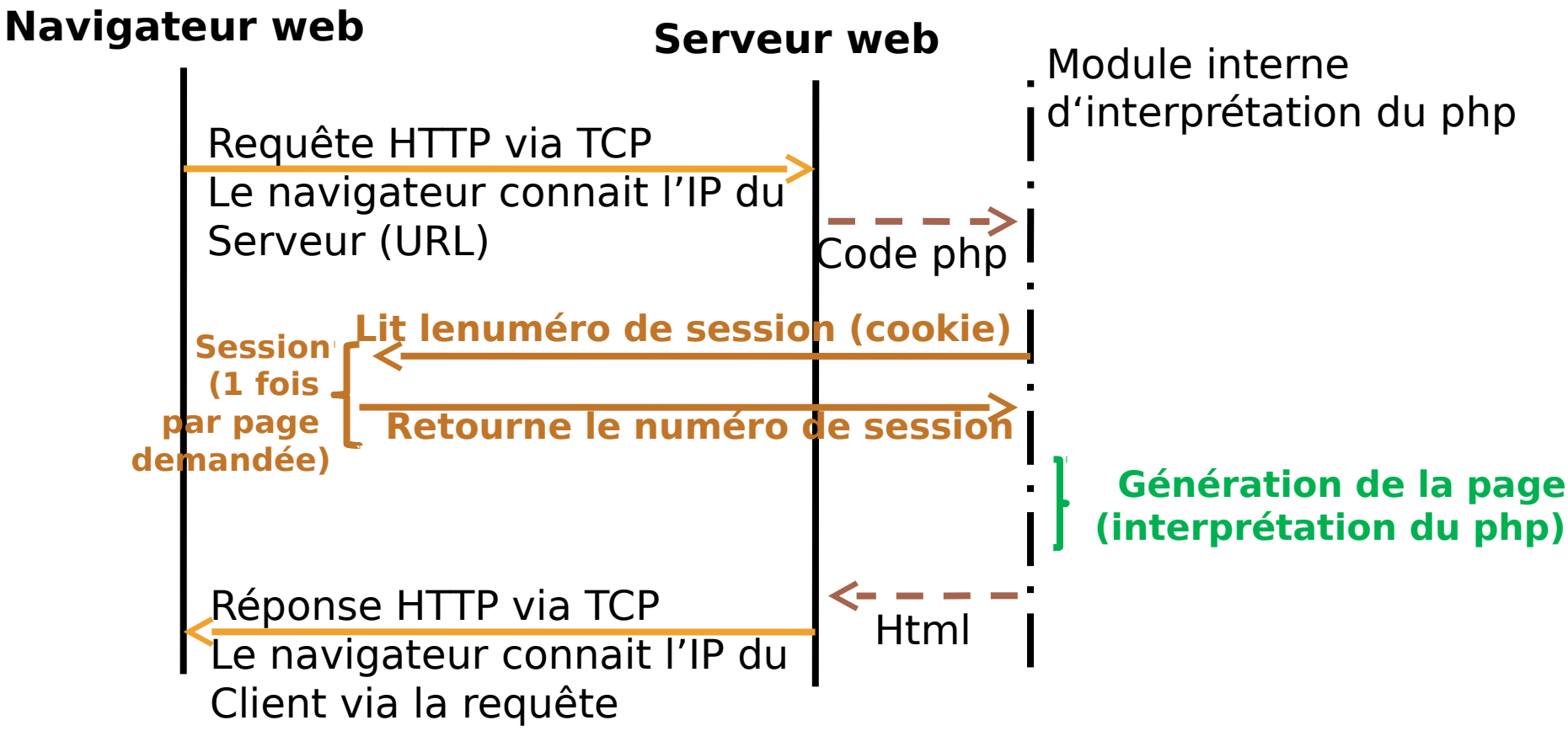
DÉTRUIRE UNE SESSION

```
// Lors d'une déconnexion par exemple, il faut détruire la session
// Initialisation de la session.
// Si vous utilisez un autre nom
// session_name("autrenom")
session_start();

// Détruit toutes les variables de session
$_SESSION = array();
// Si vous voulez détruire complètement la session, effacez également
// le cookie de session.
// cela détruira la session et pas seulement les données de session !

if (isset($_COOKIE[session_name()]))  {
    setcookie(session_name(), '', time()-42000, '/');
}
// Finalement, on détruit la session.
session_destroy();
```

REQUÊTE HTTP POUR UNE PAGE WEB



REQUÊTE HTTP POUR UNE PAGE WEB AVEC SESSION



SESSION SANS COOKIE : SID EN PARAM GET

- Dans php.ini
  - session.use_cookies = 0 (par défaut à 1)
- string session_id ([string $id])
  - session_id() pour obtenir ou forcer le numéro de session.
  - Si *id* est utilisé, il fixe le numéro de session. A utiliser avant session_start( ).
- La constante SID peut aussi être utilisée directement pour écrire le numéro de session dans les URLs

- Reconnu automatiquement si le SID est écrit en premier :

```
echo '<br /><a href="page2.php?' . SID . '">page 2</a>';
```

- Sinon, il faut utiliser un nom de variable et utiliser session_id( ) :

```
echo '<br /><a href="page2.php?
session_id=' . SID . '">page 2</a>';
…
session_id($_GET["session_id"]);
```

PARTAGE DE SESSION

- Utilisation de `session_id ("id de session")` avant `session_start`
- Chaque exécution de la page accédera à la même session
  - Attention au accès concurrent...
  - ... mais facile de faire une page où tout le monde poste : démo live !

REDIRECTION

- Fonction `header("Location: $url");`
- $url : localisation de la page de redirection
  - Chemin local
  - Chemin absolu (http://...)
- Écrit les Headers de la réponse HTTP
- **À faire avant d'écrire du texte**
  - i.e. avant d’être dans le document html
  - Le moindre espace compte

La semaine prochaine

Test de 15 minutes en début de cours

sur tout ce qui a été vu jusqu'à présent

N'arrivez pas en retard.