

PHP

February 13, 2024

Entête HTTP

Rappel : Les requêtes HTTP

WEB = service internet basé sur le protocol HTTP

Requête et réponse

- HTTP request
Requête = demande d'un navigateur (le client) au serveur web
- HTTP Response
Réponse = réponse retournée par le serveur

Entête de la requête

Les différents types d'information

- les informations sur la page cible

```
GET /site/ HTTP/1.1  
Host: localhost
```

- les précisions sur le navigateur

```
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;  
rv:34.0) Gecko/20100101 Firefox/34.0  
Accept: text/html,application/xhtml+xml,application/  
xml;q=0.9,*/*;q=0.8  
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3  
Accept-Encoding: gzip, deflate
```

- Le type de connexion

```
Connection: keep-alive
```

Entête de la requête

Les différents types d'information

- les informations sur la page cible

```
GET /site/ HTTP/1.1  
Host: localhost
```

- les précisions sur le navigateur

```
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;  
rv:34.0) Gecko/20100101 Firefox/34.0  
Accept: text/html,application/xhtml+xml,application/  
xml;q=0.9,*/*;q=0.8  
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3  
Accept-Encoding: gzip, deflate
```

- Le type de connexion

```
Connection: keep-alive
```

Entête de la requête

Les différents types d'information

- les informations sur la page cible

```
GET /site/ HTTP/1.1  
Host: localhost
```

- les précisions sur le navigateur

```
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;  
rv:34.0) Gecko/20100101 Firefox/34.0  
Accept: text/html,application/xhtml+xml,application/  
xml;q=0.9,*/*;q=0.8  
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3  
Accept-Encoding: gzip, deflate
```

- Le type de connexion

```
Connection: keep-alive
```

Exploitation de l'entête

- Ces informations permettent d'adapter la réponse du script
- Elles peuvent être récupérées à l'aide de la fonction **getallheaders()**
- On retrouve notamment:
 - ▶ le type du navigateur (User-Agent)
 - ▶ des formats des fichiers acceptés (Accept),
 - ▶ la langue (Accept-Language).

Exemple

```
$header = getallheaders();  
if (strpos($header["User-Agent"], "Firefox") > 0) {  
    print("Vous utilisez Firefox");  
}  
elseif (strpos($header["User-Agent"], "MSIE") > 0) {  
    print("Vous utilisez Internet Explorer");  
}  
else {  
    print("Navigateur non reconnu");  
}
```

Exploitation de l'entête

- Ces informations permettent d'adapter la réponse du script
- Elles peuvent être récupérées à l'aide de la fonction **getallheaders()**
- On retrouve notamment:
 - ▶ le type du navigateur (User-Agent)
 - ▶ des formats des fichiers acceptés (Accept),
 - ▶ la langue (Accept-Language).

Exemple

```
$header = getallheaders();  
if (strpos($header["User-Agent"], "Firefox") > 0) {  
    print("Vous utilisez Firefox");  
}  
elseif (strpos($header["User-Agent"], "MSIE") > 0) {  
    print("Vous utilisez Internet Explorer");  
}  
else {  
    print("Navigateur non reconnu");  
}
```


L'entête de la réponse

Les différents types d'information

- le code réponse

```
HTTP/1.1 200 OK
```

- le serveur distant

```
Date: Sat, 07 Feb 2015 10:55:21 GMT  
Server: Apache/2.4.7 (Ubuntu)  
X-Powered-By: PHP/5.5.9-1ubuntu4.5
```

- le contenu

```
Vary: Accept-Encoding  
Content-Encoding: gzip  
Content-Length: 18163
```

- la connexion

```
Keep-Alive: timeout=5, max=100  
Connection: Keep-Alive  
Content-Type: text/html // utilisé par le navigateur  
pour l'interprétation
```

L'entête de la réponse

Les différents types d'information

- le code réponse

```
HTTP/1.1 200 OK
```

- le serveur distant

```
Date: Sat, 07 Feb 2015 10:55:21 GMT  
Server: Apache/2.4.7 (Ubuntu)  
X-Powered-By: PHP/5.5.9-1ubuntu4.5
```

- le contenu

```
Vary: Accept-Encoding  
Content-Encoding: gzip  
Content-Length: 18163
```

- la connexion

```
Keep-Alive: timeout=5, max=100  
Connection: Keep-Alive  
Content-Type: text/html // utilisé par le navigateur  
pour l'interprétation
```

L'entête de la réponse

Les différents types d'information

- le code réponse

```
HTTP/1.1 200 OK
```

- le serveur distant

```
Date: Sat, 07 Feb 2015 10:55:21 GMT  
Server: Apache/2.4.7 (Ubuntu)  
X-Powered-By: PHP/5.5.9-1ubuntu4.5
```

- le contenu

```
Vary: Accept-Encoding  
Content-Encoding: gzip  
Content-Length: 18163
```

- la connexion

```
Keep-Alive: timeout=5, max=100  
Connection: Keep-Alive  
Content-Type: text/html // utilisé par le navigateur  
pour l'interprétation
```

L'entête de la réponse

Les différents types d'information

- le code réponse

```
HTTP/1.1 200 OK
```

- le serveur distant

```
Date: Sat, 07 Feb 2015 10:55:21 GMT  
Server: Apache/2.4.7 (Ubuntu)  
X-Powered-By: PHP/5.5.9-1ubuntu4.5
```

- le contenu

```
Vary: Accept-Encoding  
Content-Encoding: gzip  
Content-Length: 18163
```

- la connexion

```
Keep-Alive: timeout=5, max=100  
Connection: Keep-Alive  
Content-Type: text/html // utilisé par le navigateur  
pour l'interprétation
```

la fonction Header()

La fonction PHP **header** permet de modifier l'entête.

Attention

Elle doit être placée avant le premier affichage de la page

Exemples de **CE QUI NE MARCHE PAS**

```
Bonjour
<?php
header("Location: http://www.google.com");
?>
```

```
<?php
print("Bonjour");
header("Location: http://www.google.com");
?>
```

la fonction Header()

La fonction PHP **header** permet de modifier l'entête.

Attention

Elle doit être placée avant le premier affichage de la page

Exemples de **CE QUI NE MARCHE PAS**

```
Bonjour
<?php
header("Location: http://www.google.com");
?>
```

```
<?php
print("Bonjour");
header("Location: http://www.google.com");
?>
```

la fonction Header()

La fonction PHP **header** permet de modifier l'entête.

Attention

Elle doit être placée avant le premier affichage de la page

Exemples de **CE QUI NE MARCHE PAS**

```
Bonjour
<?php
header("Location: http://www.google.com");
?>
```

```
<?php
print("Bonjour");
header("Location: http://www.google.com");
?>
```

Utilisation

Demande de redirection

```
<?php
header("Location: http://www.google.com");
?>
```

Désactivation de la mise en cache (proxy et client)

```
<?php
// HTTP/1.1 la requête devra obligatoirement être validée
// par le serveur
header("Cache-Control: no-cache, must-revalidate");
// force le client à relancer la requête
header("Expires: Sat, 26 Jul 1997 05:00:00 GMT"); // Date
// dans le passé
?>
```


Utilisation

Demande de redirection

```
<?php
header("Location: http://www.google.com");
?>
```

Désactivation de la mise en cache (proxy et client)

```
<?php
// HTTP/1.1 la requête devra obligatoirement être validée
// par le serveur
header("Cache-Control: no-cache, must-revalidate");
// force le client à relancer la requête
header("Expires: Sat, 26 Jul 1997 05:00:00 GMT"); // Date
// dans le passé
?>
```

Utilisation

Pour spécifier que le contenu est du XML

```
<?php  
header ("Content-Type: text/xml");  
?>
```

Pour spécifier que le contenu est du text

```
<?php  
header ("Content-Type: text/plain");  
?>
```

Utilisation

Pour spécifier que le contenu est du XML

```
<?php  
header ("Content-Type: text/xml");  
?>
```

Pour spécifier que le contenu est du text

```
<?php  
header ("Content-Type: text/plain");  
?>
```

Utilisation

Pour spécifier que le contenu est du XML

```
<?php  
header ("Content-Type: text/xml");  
?>
```

Pour spécifier que le contenu est du text

```
<?php  
header ("Content-Type: text/plain");  
?>
```

Les fichiers spéciaux

- Fichiers compressés
- Fichiers csv
- Fichiers flash
- Fichiers PDF
- Fichiers Images

Le principe est d'écrire sur le flux de sortie au format approprié et de spécifier dans l'entête le type de fichier.

Fichiers csv

Le format csv

- les données sont délimitées par des guillemets,
- les colonnes sont séparées par ',' ou ';' ;
- les lignes sont repérées par '\r\n' ;

On utilise la fonction header pour spécifier au navigateur le type du contenu

```
// spécifie le format du contenu au navigateur
header("Content-type: text/x-csv");
// force SaveAs du navigateur + nom fichier
header('Content-Disposition: attachment; filename="list .
    csv"');
$str="'toto','titi'."\r\n"."tata","tutu"."\r\n";
echo $str
```

Fichiers PDF

- Il est possible de générer des fichiers pdf via php
- mais il faut installer une extension (coté serveur)
 - ▶ installer 'fpdf'
 - ▶ l'extension de gestion des PDF doit être autorisée dans le fichier php.ini
 - ▶ D'autres librairies open source existent pour créer des PDFS (mPDF, DOMPDF...)
 - ▶ Attention : créer un pdf n'est pas aussi simple qu'une page web, et doit prévoir tous les cas d'usage possibles du fichier à généré (position du texte, longueur du texte, placement des images...).

Utilisation

Ou une exploitation simple en statique peut être :

Pour spécifier que le contenu est un fichier pdf

```
<?php
// Vous voulez afficher un pdf
header('Content-Type: application/pdf');
// Il sera nommé downloaded.pdf
header('Content-Disposition: attachment; filename="
    downloaded.pdf"');
// Le source du PDF original.pdf
readfile('original.pdf');
?>
```

- Attention : pour les lien vers un pdf dans une page web, le comportement de chaque navigateur diffère suivant les régles mis en place. Un lien vers votre fichier pdf, ouvrira le pdf dans le navigateur, dans un nouvel onglet ou le téléchargera.

Fichiers Images

La création et manipulation d'images :

- basée sur la librairie GD (Gif Draw / Graphic Draw)
- vérifier son installation :
 - ▶ `phpinfo()`
 - ▶ GD Support doit être à enabled

Utilisation :

```
<html>
  <head>
    <title>Notre page de test</title>
  </head>
  <body>
    
  </body>
</html>
```

L'image sera générée par un script php.

Fichiers Images

La création et manipulation d'images :

- basée sur la librairie GD (Gif Draw / Graphic Draw)
- vérifier son installation :
 - ▶ `phpinfo()`
 - ▶ GD Support doit être à enabled

Utilisation :

```
<html>
  <head>
    <title>Notre page de test</title>
  </head>
  <body>
    
  </body>
</html>
```

L'image sera générée par un script php.

Voir gestion des images

Processus de création d'une image

- On spécifie que le contenu est une image (extension png ici)

```
header ("Content-type: image/png"); // ou remplacer  
      'png' par 'jpeg' ou 'gif'
```

- Création de l'image (avec palette de couleur = image indexée):

```
$image = @Imagecreate ($largeur , $hauteur);
```

- Définir les couleurs que vous souhaitez utiliser :

```
$rouge = ImageColorAllocate ($image, 255, 0, 0);  
$blanc = ImageColorAllocate ($image, 255, 255, 255);
```

- "Dessinez" les formes géométriques dans votre image
- Affichez ou sauvegardez le contenu de l'image

```
ImagePng($image) // ou ImageGif / ImageJpeg
```

Voir gestion des images

Processus de création d'une image

- On spécifie que le contenu est une image (extension png ici)

```
header ("Content-type: image/png"); // ou remplacer  
      'png' par 'jpeg' ou 'gif'
```

- Création de l'image (avec palette de couleur = image indexée):

```
$image = @Imagecreate ($largeur , $hauteur);
```

- Définir les couleurs que vous souhaitez utiliser :

```
$rouge = ImageColorAllocate ($image, 255, 0, 0);  
$blanc = ImageColorAllocate ($image, 255, 255, 255);
```

- "Dessinez" les formes géométriques dans votre image
- Affichez ou sauvegardez le contenu de l'image

```
ImagePng($image) // ou ImageGif / ImageJpeg
```

Voir gestion des images

Processus de création d'une image

- On spécifie que le contenu est une image (extension png ici)

```
header ("Content-type: image/png"); // ou remplacer  
'png' par 'jpeg' ou 'gif'
```

- Création de l'image (avec palette de couleur = image indexée):

```
$image = @Imagecreate ($largeur , $hauteur);
```

- Définir les couleurs que vous souhaitez utiliser :

```
$rouge = ImageColorAllocate ($image , 255 , 0 , 0);  
$blanc = ImageColorAllocate ($image , 255 , 255 , 255);
```

- "Dessinez" les formes géométriques dans votre image
- Affichez ou sauvegardez le contenu de l'image

```
ImagePng($image) // ou ImageGif / ImageJpeg
```

Voir gestion des images

Processus de création d'une image

- On spécifie que le contenu est une image (extension png ici)

```
header ("Content-type: image/png"); // ou remplacer  
'png' par 'jpeg' ou 'gif'
```

- Création de l'image (avec palette de couleur = image indexée):

```
$image = @Imagecreate ($largeur , $hauteur);
```

- Définir les couleurs que vous souhaitez utiliser :

```
$rouge = ImageColorAllocate ($image , 255 , 0 , 0);  
$blanc = ImageColorAllocate ($image , 255 , 255 , 255);
```

- "Dessinez" les formes géométriques dans votre image
- Affichez ou sauvegardez le contenu de l'image

```
ImagePng($image) // ou ImageGif / ImageJpeg
```

Voir gestion des images

Processus de création d'une image

- On spécifie que le contenu est une image (extension png ici)

```
header ("Content-type: image/png"); // ou remplacer  
      'png' par 'jpeg' ou 'gif'
```

- Création de l'image (avec palette de couleur = image indexée):

```
$image = @Imagecreate ($largeur , $hauteur);
```

- Définir les couleurs que vous souhaitez utiliser :

```
$rouge = ImageColorAllocate ($image , 255 , 0 , 0);  
$blanc = ImageColorAllocate ($image , 255 , 255 , 255);
```

- "Dessinez" les formes géométriques dans votre image
- Affichez ou sauvegardez le contenu de l'image

```
ImagePng($image) // ou ImageGif / ImageJpeg
```

Exemple :

```
<?php
header ("Content-type: image/png");
$image = @imagecreate (150, 150);
$noir = ImageColorAllocate ($image, 0, 0, 0);
$blanc = ImageColorAllocate ($image, 255, 255, 255);
ImageFilledRectangle ($image, 10, 10, 20, 20, $blanc);
ImagePng ($image);
?>
```

ATTENTION : le premier appel de **ImageColorAllocate** définit la couleur de fond de l'image (pour les images indexées).

Rappel sur les formats d'images :

- JPEG : très bonne compression pour des images complexes (photos)
- GIF : très bonne compression pour des images simples, gère la transparence (tout ou rien).
- PNG : très bonne compression générale, gestion du niveau de transparence.
- WBMP : Wireless Application Protocol Bitmap Format (format monochrome pour applications sans fil)
- WEBP : Nouveau format poussé par Google (proche du jpg dans sa structure)

Attention

- `imagecreate` : crée une image indexée (256 couleurs possibles)
- `imagecreatetruecolor` autant de couleur que possible
- En pratique
 - ▶ utilisez `imagecreatetruecolor`
 - ▶ si vous voulez une image de couleurs indexées faire un appel de `imagetruecolortopalette()` avant l'affichage
- sinon `imagecolorresolve` : Retourne l'index de la couleur donnée, ou la plus proche possible

Quelques fonctions pour dessiner

<code>imagepixel</code>	Dessine un pixel
<code>imageline</code>	Dessine une ligne
<code>imagerectangle</code>	dessine un rectangle aux coordonnées spécifiées
<code>imagepolygon</code>	dessine un polygone dans l'image
<code>imagerotate</code>	fait tourner l'image d'un angle en degrés
<code>imagescale</code>	Met à l'échelle une image en utilisant une hauteur et une largeur fournies
<code>imagefttext</code>	Écrit du texte dans une image avec la police courante FreeType 2
<code>imagestring</code>	Dessine une chaîne horizontale
<code>imageftttext</code>	Dessine un texte avec une police TrueType suivant une direction
<code>imagecreatefromxxx</code>	xxx=jpeg, png, ... Crée une nouvelle image depuis un fichier ou une URL

Les sessions et les cookies

Les sessions et les cookies

Interêt

- associer des données aux internautes
 - ▶ les cookies
 - ▶ les sessions
- à chaque nouvelle visite l'internaute retrouve des informations personnalisées
- facile à mettre en oeuvre (ne nécessite pas de base de données)

Limite

- ils peuvent être désactivés par l'internaute
- ils sont liés au profil de l'utilisateur : session/machine

Les sessions et les cookies

Interêt

- associer des données aux internautes
 - ▶ les cookies
 - ▶ les sessions
- à chaque nouvelle visite l'internaute retrouve des informations personnalisées
- facile à mettre en oeuvre (ne nécessite pas de base de données)

Limite

- ils peuvent être désactivés par l'internaute
- ils sont liés au profil de l'utilisateur : session/machine

Les cookies

Principes

- des informations sont stockées sur l'ordinateur de l'internaute (dans son profil utilisateur)
- ces informations sont stockées sous forme de texte dans des fichiers
- cloisonnement de données : les informations sont gérées par site visité

Exemple d'exploitation

- ne pas avoir à retaper systématiquement votre identifiant et votre mot de passe sur certains sites ;
- disposer d'un système de panier très pratique dans les boutiques en ligne.
- profilage utilisateur, centres d'intérêt
 - ▶ quelles sont les pages les plus visitées
 - ▶ quels sont les produits consultés

Les cookies

Principes

- des informations sont stockées sur l'ordinateur de l'internaute (dans son profil utilisateur)
- ces informations sont stockées sous forme de texte dans des fichiers
- cloisonnement de données : les informations sont gérées par site visité

Exemple d'exploitation

- ne pas avoir à retaper systématiquement votre identifiant et votre mot de passe sur certains sites ;
- disposer d'un système de panier très pratique dans les boutiques en ligne.
- profilage utilisateur, centres d'intérêt
 - ▶ quelles sont les pages les plus visitées
 - ▶ quels sont les produits consultés

Les cookies

Mise en œuvre côté HTTP :

- ce sont des informations au format texte
- ils sont transmis par l'entête http

Exemple de directive HTTP de création de 'moncookie'

```
Set-Cookie: moncookie=hello; path=/; expires Mon, 09-Dec-2002 13:46:00 GMT
```

Directive HTTP d'envoi du contenu d'un cookie par un navigateur

```
Cookie: moncookie=hello
```

Les cookies

Mise en œuvre côté HTTP :

- ce sont des informations au format texte
- ils sont transmis par l'entête http

Exemple de directive HTTP de création de 'moncookie'

```
Set-Cookie: moncookie=hello; path=/; expires Mon, 09-Dec-2002 13:46:00 GMT
```

Directive HTTP d'envoi du contenu d'un cookie par un navigateur

```
Cookie: moncookie=hello
```

Les cookies

Mise en œuvre coté HTTP :

- ce sont des informations au format texte
- ils sont transmis par l'entête http

Exemple de directive HTTP de création de 'moncookie'

```
Set-Cookie: moncookie=hello; path=/; expires Mon, 09-Dec-2002 13:46:00 GMT
```

Directive HTTP d'envoi du contenu d'un cookie par un navigateur

```
Cookie: moncookie=hello
```

Les cookies

Avec PHP on utilise la fonction: **setcookie()**

```
<?php  
setcookie ("moncookie" ,"hello");  
echo 'cookie créé';  
?>
```

ATTENTION : les cookies doivent être envoyés avant toute autre sortie

Lire un cookie

```
<?php  
echo "valeur contenue dans mon cookie : ".$_COOKIE[ '  
    moncookie '];  
?>
```

Les cookies

Avec PHP on utilise la fonction: **setcookie()**

```
<?php
setcookie ("moncookie" ,"hello");
echo 'cookie créé';
?>
```

ATTENTION : les cookies doivent être envoyés avant toute autre sortie
Lire un cookie

```
<?php
echo "valeur contenue dans mon cookie : ".$_COOKIE[ '
    moncookie '];
?>
```

La variable superglobale :

`$_COOKIE[]`

Contient autant d'éléments que de cookies créés.

Ils sont également dans `$_REQUEST[]`.

Les limites

- Seul le site qui a créé le cookie peut y accéder.
- La taille d'un cookie doit être inférieure à 4 ko.
- Le nombre de cookies créés par un domaine donné est limité à 20.

La variable superglobale :

```
$_COOKIE[]
```

Contient autant d'éléments que de cookies créés.

Ils sont également dans `$_REQUEST[]`.

Les limites

- Seul le site qui a créé le cookie peut y accéder.
- La taille d'un cookie doit être inférieure à 4 ko.
- Le nombre de cookies créés par un domaine donné est limité à 20.

les autres paramètres:

- La date d'expiration : en secondes,
- Le chemin : indique quelle partie du site peut avoir accès au cookie.
- Le domaine : indique quel domaine peut avoir accès au site.
- Un indicateur de sécurité : si sa valeur est 1 , il indique que la valeur du cookie ne peut être transmise que si vous êtes en HTTS (HTTP sécurisé par du SSL).

Attention

Si on ne précise pas la date d'expiration le cookie n'est valable que le temps de la session

les autres paramètres:

- La date d'expiration : en secondes,
- Le chemin : indique quelle partie du site peut avoir accès au cookie.
- Le domaine : indique quel domaine peut avoir accès au site.
- Un indicateur de sécurité : si sa valeur est 1 , il indique que la valeur du cookie ne peut être transmise que si vous êtes en HTTS (HTTP sécurisé par du SSL).

Attention

Si on ne précise pas la date d'expiration le cookie n'est valable que le temps de la session

Introduction

Certains inconvénients des cookies :

- obligation de les gérer en haut du script avant que des données ne soient affichées
- le format de stockage des données = chaîne de caractères, pas de format sophistiqué).

Fonctionnement des sessions

- la fonction **session_start()** : indique au script que vous souhaitez récupérer les variables de session.
- la variable super-globale **\$_SESSION** : contient l'ensemble des variables de session.

Utilisation = utilisation d'un tableau

- `session_start()` ;
`$_SESSION['nom'] = $_GET['nom'] ; // création d'une variable de session`

```
session_start() ;  
$nom = $_SESSION[ 'nom' ] ; // lecture d'une variable de session
```

Fonctionnement des sessions

- la fonction **session_start()** : indique au script que vous souhaitez récupérer les variables de session.
- la variable super-globale **\$_SESSION** : contient l'ensemble des variables de session.

Utilisation = utilisation d'un tableau

- ```
session_start();
$_SESSION['nom'] = $_GET['nom']; // création d'une
 variable de session
```

```
session_start();
$nom = $_SESSION['nom']; // lecture d'une variable de
 session
```

# Fonctionnement des sessions

- la fonction **session\_start()** : indique au script que vous souhaitez récupérer les variables de session.
- la variable super-globale **\$\_SESSION** : contient l'ensemble des variables de session.

## Utilisation = utilisation d'un tableau

- ```
session_start();  
$_SESSION[ 'nom' ] = $_GET[ 'nom' ]; // création d'une  
    variable de session
```

```
session_start();  
$nom = $_SESSION[ 'nom' ]; // lecture d'une variable de  
    session
```

Exemple

```
<?php
session_start();
if (!isset($_SESSION['visite'])) {
    echo "première visite";
    $_SESSION['visite'] = 1;
}
else {
    $_SESSION['visite']++;
    echo "vous avez visité cette page " . $_SESSION['visite'] . "
        fois";
}
?>
```

Remarque

la variable `$_SESSION` (superglobale) est accessible de n'importe quel script (si `session_start()`)

Exemple

```
<?php
session_start();
if (!isset($_SESSION['visite'])) {
    echo "première visite";
    $_SESSION['visite'] = 1;
}
else {
    $_SESSION['visite']++;
    echo "vous avez visité cette page " . $_SESSION['visite'] . "
        fois";
}
?>
```

Remarque

la variable `$_SESSION` (superglobale) est accessible de n'importe quel script (si `session_start()`)

Identifiant de session

La fonction `session_id()`

```
<?php
session_start() ;
echo "Identifiant de session : " . session_id() ;
?>
```

Remarques

- Les variables de session sont stockées sur le serveur web et sont reconnues grâce à l'identifiant unique
- l'identifiant de session est stocké dans un cookie sur votre disque.
- \Rightarrow les cookies doivent être activés
- Il n'y a pas de limitation (taille / nombre)
- On peut imposer/spécifier un ID de session
`session_start(MON_ID_SESSION)` (échange de donnée entre utilisateurs)

Identifiant de session

La fonction `session_id()`

```
<?php
session_start() ;
echo "Identifiant de session : ".session_id() ;
?>
```

Remarques

- Les variables de session sont stockées sur le serveur web et sont reconnues grâce à l'identifiant unique
- l'identifiant de session est stocké dans un cookie sur votre disque.
- ⇒ les cookies doivent être activés
- Il n'y a pas de limitation (taille / nombre)
- On peut imposer/spécifier un ID de session
`session_start(MON_ID_SESSION)` (échange de donnée entre utilisateurs)

Identifiant de session

La fonction `session_id()`

```
<?php
session_start() ;
echo "Identifiant de session : " . session_id() ;
?>
```

Remarques

- Les variables de session sont stockées sur le serveur web et sont reconnues grâce à l'identifiant unique
- l'identifiant de session est stocké dans un cookie sur votre disque.
- \Rightarrow les cookies doivent être activés
- Il n'y a pas de limitation (taille / nombre)
- On peut imposer/spécifier un ID de session
`session_start(MON_ID_SESSION)` (échange de donnée entre utilisateurs)

Identifiant de session

La fonction `session_id()`

```
<?php
session_start() ;
echo "Identifiant de session : " . session_id() ;
?>
```

Remarques

- Les variables de session sont stockées sur le serveur web et sont reconnues grâce à l'identifiant unique
- l'identifiant de session est stocké dans un cookie sur votre disque.
- \Rightarrow les cookies doivent être activés
- Il n'y a pas de limitation (taille / nombre)
- On peut imposer/spécifier un ID de session
`session_start(MON_ID_SESSION)` (échange de donnée entre utilisateurs)

Identifiant de session

La fonction `session_id()`

```
<?php  
session_start();  
echo "Identifiant de session : ".session_id();  
?>
```

Remarques

- Les variables de session sont stockées sur le serveur web et sont reconnues grâce à l'identifiant unique
- l'identifiant de session est stocké dans un cookie sur votre disque.
- \Rightarrow les cookies doivent être activés
- Il n'y a pas de limitation (taille / nombre)
- On peut imposer/spécifier un ID de session
`session_start(MON_ID_SESSION)` (échange de donnée entre utilisateurs)

Durée d'une session

Durée

- Par défaut, une session est détruite quand l'internaute ferme son navigateur
- Peut être maintenue par

```
session_set_cookie_params(time() + 604800);
```
- La fonction `session_set_cookie_params()` doit être appelée avant chaque appel à `session_start()`

Durée d'une session

Durée

- Par défaut, une session est détruite quand l'internaute ferme son navigateur
- Peut être maintenue par

```
session_set_cookie_params(time() + 604800);
```

- La fonction `session_set_cookie_params()` doit être appelée avant chaque appel à `session_start()`

Durée d'une session

Durée

- Par défaut, une session est détruite quand l'internaute ferme son navigateur
- Peut être maintenue par

```
session_set_cookie_params( time() + 604800 );
```
- La fonction `session_set_cookie_params()` doit être appelée avant chaque appel à `session_start()`

Programmation objet

PHP Objet

Mieux depuis la version 5

Avantages

- la modélisation ;
- la maintenance ;
- la relecture ;
- l'évolutivité.

les nouvelles extensions sont fournies sous forme de classes (ex: PDO, XML, SOAP, etc.)

Les classes

Déclaration

```
class Rectangle {  
  // Attributs  
  public $longueur = null;  
  public $largeur = null;  
  public $couleur = "rouge";  
  
  function perimetre() {  
    if ($this->longueur!=null && $this->largeur!=null)  
      {  
        return (2*$this->longueur+2*$this->largeur);  
      }  
  }  
  
  function surface() {  
    if ($this->longueur!=null &&  
    $this->largeur!=null) {  
      return ($this->longueur*$this->largeur);  
    }  
  }  
}
```

Initialisation des attributs

- Uniquement avec une valeur constante
- Interdit

```
public $date = date();  
public $id = "Paul"."Dupont";
```

Les objets

- Instanciation

```
$rect = new Rectangle();
```

- Accès aux attributs et méthodes

```
$rect->longueur = 3;  
$rect->largeur = 5;  
$tmp = $rect->perimetre();
```

- Accès aux attributs à l'aide de variables

```
$rect = new Rectangle(3,5);  
$attr = "longueur";  
$rect->$attr = 4;
```

Création dynamique d'attributs

```
$monrect = new Rectangle() ;  
$monrect->toto = 1 ;
```

- L'objet \$monrect contient désormais l'attribut \$toto.
- Attention l'attribut n'est pas ajouté à la classe
- Il est possible d'ajouter dynamiquement une méthode à une classe (cf `runkit_method_add`).

var_dump

La fonction `var_dump()` permet de visualiser les attributs d'une classe

```
$rect = new Rectangle();  
$rect->toto=4;  
print("<pre>");  
var_dump($rect);  
print("</pre>");
```

Résultat:

```
object(Rectangle)#1 (4) {  
    ["longueur"]=> NULL  
    ["largeur"]=> NULL  
    ["couleur"]=> string(5) "rouge"  
    ["toto"]=> int(4)  
}
```

Organisation du code

- Pour créer une instance la définition de la classe doit être incluse dans le script
- On crée un fichier par classe
- On définit la fonction "__autoload()" suivante en début du script

```
<?php
function __autoload($class_name) {
    require_once("class/".$class_name.".php");
}
$obj1 = new MaClasse1();
$obj2 = new MaClasse2();
?>
```

- à chaque création d'une nouvelle instance la fonction __autoload() est appelée automatiquement par l'interpréteur (si nécessaire).

Organisation du code

- Pour créer une instance la définition de la classe doit être incluse dans le script
- On crée un fichier par classe
- On définit la fonction "__autoload()" suivante en début du script

```
<?php
function __autoload($class_name) {
    require_once("class/".$class_name.".php");
}
$obj1 = new MaClasse1();
$obj2 = new MaClasse2();
?>
```

- à chaque création d'une nouvelle instance la fonction __autoload() est appelée automatiquement par l'interpréteur (si nécessaire).

Organisation du code

- Pour créer une instance la définition de la classe doit être incluse dans le script
- On crée un fichier par classe
- On définit la fonction "__autoload()" suivante en début du script

```
<?php
function __autoload($class_name) {
    require_once("class/".$class_name . '.php');
}
$obj = new MaClasse1();
$obj2 = new MaClasse2();
?>
```

- à chaque création d'une nouvelle instance la fonction __autoload() est appelée automatiquement par l'interpréteur (si nécessaire).

Organisation du code

- Pour créer une instance la définition de la classe doit être incluse dans le script
- On crée un fichier par classe
- On définit la fonction "__autoload()" suivante en début du script

```
<?php
function __autoload($class_name) {
    require_once("class/".$class_name.".php");
}
$obj1 = new MaClasse1();
$obj2 = new MaClasse2();
?>
```

- à chaque création d'une nouvelle instance la fonction __autoload() est appelée automatiquement par l'interpréteur (si nécessaire).

Conversion

d'objet en tableau

```
class Test {  
    public $a = 1;  
    public $b = 2;  
    function hello {}  
}  
$test = new Test();  
$tab = (array) $test;
```

de tableau en objet

```
$tab = array("a"=>1, "b"=>2);  
$obj = (object) $tab;  
var_dump($obj);
```

Créer un objet avec uniquement les attributs.

Conversion

d'objet en tableau

```
class Test {  
    public $a = 1;  
    public $b = 2;  
    function hello {}  
}  
$test = new Test();  
$tab = (array) $test;
```

de tableau en objet

```
$tab = array("a"=>1, "b"=>2);  
$obj = (object) $tab;  
var_dump($obj);
```

Créer un objet avec uniquement les attributs.

Constructeur

Constructeur = methode `__construct()`

```
class Rectangle {  
    ...  
    function __construct($longueur, $largeur) {  
        $this->longueur = $longueur;  
        $this->largeur = $largeur;}  
    ...  
}
```

- Attention il n'y a pas de polymorphisme en PHP
- donc un seul constructeur possible
- peut être contourné en utilisant les tableaux ou en exploitant les fonctions `func_num_args()` et `func_get_args()`;

Constructeur

Constructeur = methode `__construct()`

```
class Rectangle {  
    ...  
    function __construct($longueur, $largeur) {  
        $this->longueur = $longueur;  
        $this->largeur = $largeur;  
    }  
    ...  
}
```

- Attention il n'y a pas de polymorphisme en PHP
- donc un seul constructeur possible
- peut être contourné en utilisant les tableaux ou en exploitant les fonctions `func_num_args()` et `func_get_args()`;

Exemple de constructeur avec différents nb d'arguments

```
public function __construct()  
{  
    $ctp = func_num_args();  
    $args = func_get_args();  
    switch($ctp)  
    {  
        case 5:  
            $this->fonction1($args[0], $args[1], $args[2],  
                            $args[3], $args[4]);  
            break;  
        case 2:  
            $this->fonction2($args[0], $args[1]);  
            break;  
        case 3:  
            $this->fonction3($args[0], $args[1], $args[2]);  
            break;  
        default:  
            break;  
    }  
}
```

Destructeur

destructeur = methode `__destruct()`

```
class Rectangle {  
    ...  
    function __destruct() {  
        print("Bye bye <br/>");  
    }  
    ...  
}
```

REMARQUE :

- il ne peut y avoir qu'un seul destructeur
- `unset($objet)` appelle le destructeur
- Les objets sont détruits automatiquement en fin de script

Destructeur

destructeur = methode `__destruct()`

```
class Rectangle {  
    ...  
    function __destruct() {  
        print("Bye bye <br/>");  
    }  
    ...  
}
```

REMARQUE :

- il ne peut y avoir qu'un seul destructeur
- `unset($objet)` appelle le destructeur
- Les objets sont détruits automatiquement en fin de script

Destructeur

destructeur = methode `__destruct()`

```
class Rectangle {  
    ...  
    function __destruct() {  
        print("Bye bye <br/>");  
    }  
    ...  
}
```

REMARQUE :

- il ne peut y avoir qu'un seul destructeur
- `unset($objet)` appelle le destructeur
- Les objets sont détruits automatiquement en fin de script

Destructeur

destructeur = methode `__destruct()`

```
class Rectangle {  
    ...  
    function __destruct() {  
        print("Bye bye <br/>");  
    }  
    ...  
}
```

REMARQUE :

- il ne peut y avoir qu'un seul destructeur
- `unset($objet)` appelle le destructeur
- Les objets sont détruits automatiquement en fin de script

Les fonctions magiques

Php réserve les noms de fonctions commençant par '___' :

Exemple : `__construct`, `__destruct`, `__toString`.

- La fonction `__toString` La méthode magique `__toString()` va être appelée dès que l'on va traiter un objet comme une chaîne de caractères

```
class Rectangle {
    const NOM = "Super Rectangle";
    ...
    function __toString() {
        return (self::NOM." [".$this->longueur."x".
            $this->largeur."]");
    }
    ...
}
```

```
$rect = new Rectangle(3,5);
print($rect); // affiche : Super Rectangle [3x5]
```

Les fonctions magiques

Php réserve les noms de fonctions commençant par '___' :

Exemple : `__construct`, `__destruct`, `__toString`.

- La fonction `__toString` La méthode magique `__toString()` va être appelée dès que l'on va traiter un objet comme une chaîne de caractères

```
class Rectangle {
    const NOM = "Super Rectangle";
    ...
    function __toString() {
        return (self::NOM." [".$this->longueur."x".
            $this->largeur."]");
    }
    ...
}
```

```
$rect = new Rectangle(3,5);
print($rect); // affiche : Super Rectangle [3x5]
```

Les fonctions magiques

Les fonctions `__set()` et `__get()` sont appelées lorsqu'on écrit/lit sur un attribut non existant.

Exemple :

```
function __set($nom, $val) {
    if ($nom=="dim" && count($val)==2) {
        list ($this->longueur, $this->largeur) = $val;
    }
}
function __get($nom) {
    if ($nom=="dim") {
        return array($this->longueur, $this->largeur);
    }
}
```

```
$rect = new Rectangle(3,5);
$rect->dim = array(2,3);
$dim = implode("x",$rect->dim); // retour la chaîne à
    partir du tableau
echo "Surface d'un rectangle de $dim = ";
echo $rect->surface();
```

Les fonctions magiques

Les fonctions `__set()` et `__get()` sont appelées lorsqu'on écrit/lit sur un attribut non existant.

Exemple :

```
function __set($nom, $val) {
    if ($nom=="dim" && count($val)==2) {
        list ($this->longueur, $this->largeur) = $val;
    }
}
function __get($nom) {
    if ($nom=="dim") {
        return array($this->longueur, $this->largeur);
    }
}
```

```
$rect = new Rectangle(3,5);
$rect->dim = array(2,3);
$dim = implode("x", $rect->dim); // retour la chaîne à
    partir du tableau
echo "Surface d'un rectangle de $dim = ";
echo $rect->surface();
```

Les fonctions magiques

Les fonctions `__set()` et `__get()` sont appelées lorsqu'on écrit/lit sur un attribut non existant.

Exemple :

```
function __set($nom, $val) {
    if ($nom=="dim" && count($val)==2) {
        list ($this->longueur, $this->largeur) = $val;
    }
}
function __get($nom) {
    if ($nom=="dim") {
        return array($this->longueur, $this->largeur);
    }
}
```

```
$rect = new Rectangle(3,5);
$rect->dim = array(2,3);
$dim = implode("x", $rect->dim); // retour la chaîne à
    partir du tableau
echo "Surface d'un rectangle de $dim = ";
echo $rect->surface();
```


Héritage

```
class B extends A {  
}
```

Exemple

```
class Carre extends Rectangle {  
    public $cote = null;  
    const NOM = "Carré";  
    function __construct($cote) {  
        parent::__construct($cote, $cote); // l'appel du  
        constructeur du parent n'est pas fait par défaut  
        $this->cote = $cote;  
    }  
}
```

Remarque : Il n'y a pas d'héritage multiple

Visibilité des méthodes et attributs

public, protected, private

- public : utilisable à partir de n'importe quelle partie du programme
- protected : limité à la classe elle-même, aux classes héritières, et aux classes parentes
- private : uniquement réservé à la classe

Mot clé final

- Devant la définition de la classe, il interdit qu'elle soit étendue
- Devant la définition d'une méthode, il interdit que cette méthode soit surchargée.

Visibilité des méthodes et attributs

public, protected, private

- public : utilisable à partir de n'importe quelle partie du programme
- protected : limité à la classe elle-même, aux classes héritières, et aux classes parentes
- private : uniquement réservé à la classe

Mot clé final

- Devant la définition de la classe, il interdit qu'elle soit étendue
- Devant la définition d'une méthode, il interdit que cette méthode soit surchargée.

Les interfaces

Une interface permet de créer des classes, en spécifiant les méthodes publiques qu'une classe doit implémenter, sans impliquer la complexité et les détails de l'implémentation des méthodes.

Définition d'un interface :

```
interface MonInterface {  
    public function methode1 ();  
    public function methode2 ();  
    // etc.  
}
```

Toutes les méthodes de l'interface doivent être implémentées

Les interfaces

Une interface permet de créer des classes, en spécifiant les méthodes publiques qu'une classe doit implémenter, sans impliquer la complexité et les détails de l'implémentation des méthodes.

Définition d'un interface :

```
interface MonInterface {  
    public function methode1 ();  
    public function methode2 ();  
    // etc.  
}
```

Toutes les méthodes de l'interface doivent être implémentées

Exemple :

```
interface ObjetGeom {
    public function aCotesEgaux ();
}
```

...

```
class Rectangle implements ObjetGeom{
    ...
    function aCotesEgaux () {
        if ($this->largeur==$this->longueur) {
            return true;
        }
        return false;
    }
}
```

```
class Carre extends Rectangle {
    ...
    function aCotesEgaux () { // surcharge de l'interface
        return true;
    }
}
```

Exemple :

```
interface ObjetGeom {  
    public function aCotesEgaux ();  
}
```

...

```
class Rectangle implements ObjetGeom{  
    ...  
    function aCotesEgaux () {  
        if ($this->largeur==$this->longueur) {  
            return true;  
        }  
        return false;  
    }  
}
```

```
class Carre extends Rectangle {  
    ...  
    function aCotesEgaux () { // surcharge de l'interface  
        return true;  
    }  
}
```

Itérateurs d'objets

Ils permettent de parcourir les attributs d'un objet.

Exemple :

```
$rect = new Rectangle(3,5);  
foreach ($rect as $attr => $val) {  
  print (" $attr = $val<br/>");  
}
```

Affiche :

```
longueur = 3  
largeur = 5  
couleur = rouge
```


Itérateurs d'objets

Ils permettent de parcourir les attributs d'un objet.

Exemple :

```
$rect = new Rectangle(3,5);  
foreach ($rect as $attr => $val) {  
  print (" $attr = $val<br/>");  
}
```

Affiche :

```
longueur = 3  
largeur = 5  
couleur = rouge
```

Iterateur

Il est possible de redéfinir le parcours des attributs en implémentant l'interface "Iterator".

- `rewind()` : Revient au début de la liste
- `current()` : Valeur de l'élément en cours
- `key()` : Valeur de la clé de l'élément en cours
- `next()` : Passer à l'élément suivant
- `valid()` : Retourne false lorsque vous êtes à la fin de la ligne (true sinon)

Copie d'objet vs Clonage d'objet

```
<?php
$objet = new Classe;
?>
```

\$objet ne contient pas l'objet lui-même

- mais son identifiant unique.
- Quand on passe un objet en paramètre, on passe son identifiant.
- Par exemple on n'est pas obligé de passer la référence de l'objet pour modifier les "valeurs" de l'objet dans la fonction.

```
<?php
$copie_d_objet = $objet;
$copie_d_objet->attr1='newVal' ; // modifie la valeur de
    l'attribut de $objet
?>
```

Copie d'objet vs Clonage d'objet

```
<?php  
$objet = new Classe;  
?>
```

\$objet ne contient pas l'objet lui-même

- mais son identifiant unique.
- Quand on passe un objet en paramètre, on passe son identifiant.
- Par exemple on n'est pas obligé de passer la référence de l'objet pour modifier les "valeurs" de l'objet dans la fonction.

```
<?php  
$copie_d_objet = $objet;  
$copie_d_objet->attr1='newVal' ; // modifie la valeur de  
    l'attribut de $objet  
?>
```

Copie d'objet vs Clonage d'objet

```
<?php
$objet = new Classe;
?>
```

\$objet ne contient pas l'objet lui-même

- mais son identifiant unique.
- Quand on passe un objet en paramètre, on passe son identifiant.
- Par exemple on n'est pas obligé de passer la référence de l'objet pour modifier les "valeurs" de l'objet dans la fonction.

```
<?php
$copie_d_objet = $objet;
$copie_d_objet->attr1='newVal' ; // modifie la valeur de
    l'attribut de $objet
?>
```

Copie d'objet vs Clonage d'objet

```
<?php
$objet = new Classe;
?>
```

\$objet ne contient pas l'objet lui-même

- mais son identifiant unique.
- Quand on passe un objet en paramètre, on passe son identifiant.
- Par exemple on n'est pas obligé de passer la référence de l'objet pour modifier les "valeurs" de l'objet dans la fonction.

```
<?php
$copie_d_objet = $objet;
$copie_d_objet->attr1='newVal' ; // modifie la valeur de
    l'attribut de $objet
?>
```

Copie d'objet vs Clonage d'objet

```
<?php  
$objet = new Classe;  
?>
```

\$objet ne contient pas l'objet lui-même

- mais son identifiant unique.
- Quand on passe un objet en paramètre, on passe son identifiant.
- Par exemple on n'est pas obligé de passer la référence de l'objet pour modifier les "valeurs" de l'objet dans la fonction.

```
<?php  
$copie_d_objet = $objet;  
$copie_d_objet->attr1='newVal' ; // modifie la valeur de  
    l'attribut de $objet  
?>
```

Copie d'objet vs Clonage d'objet

```
<?php
$objet = new Classe;
?>
```

\$objet ne contient pas l'objet lui-même

- mais son identifiant unique.
- Quand on passe un objet en paramètre, on passe son identifiant.
- Par exemple on n'est pas obligé de passer la référence de l'objet pour modifier les "valeurs" de l'objet dans la fonction.

```
<?php
$copie_d_objet = $objet;
$copie_d_objet->attr1='newVal' ; // modifie la valeur de
    l'attribut de $objet
?>
```


Copie d'objet vs Clonage d'objet

Pour copier tous les attributs et valeurs dans un nouvel objet unique : **on le clone**

```
<?php
$clonage_d_objet = clone $objet;
$clonage_d_objet->attr1='newVal' ; // ne modifie pas la
    valeur de l'attribut de $objet
?>
```

ATTENTION

Lorsqu'un objet est cloné, si des attributs sont des références à d'autres variables ont aura une copie de ces références.

Copie d'objet vs Clonage d'objet

Pour copier tous les attributs et valeurs dans un nouvel objet unique : **on le clone**

```
<?php
$clonage_d_objet = clone $objet;
$clonage_d_objet->attr1='newVal' ; // ne modifie pas la
    valeur de l'attribut de $objet
?>
```

ATTENTION

Lorsqu'un objet est cloné, si des attributs sont des références à d'autres variables ont aura une copie de ces références.

Copie d'objet vs Clonage d'objet

Pour copier tous les attributs et valeurs dans un nouvel objet unique : **on le clone**

```
<?php
$clonage_d_objet = clone $objet;
$clonage_d_objet->attr1='newVal' ; // ne modifie pas la
    valeur de l'attribut de $objet
?>
```

ATTENTION

Lorsqu'un objet est cloné, si des attributs sont des références à d'autres variables ont aura une copie de ces références.

SimpleXML

SimpleXML

- est une extension de PHP
- permet de construire , lire, manipuler les documents XML
- gère la représentation arborescente du document
- offre les fonctionnalités de base les plus utiles

Si insuffisant utiliser DOM (Document Object Model)

Création d'un document XML

À partir d'une chaîne de caractère

```
$str= '<polygone><point><x>2</x><y>3</y></point><point><x>2</x><y>3</y></point></polygone>' ;  
$poylgone = new SimpleXMLElement( $str );
```

ou

```
$str= '<polygone><point><x>2</x><y>3</y></point><point><x>2</x><y>3</y></point></polygone>' ;  
$poylgone = simplexml_load_string( $str );
```

Exporter et sauvegarder

La méthode **asXML** transforme la structure en chaîne de caractères

Exemple: Affichage

```
echo $poylgone->asXML() ;
```

Exemple: enregistrement dans un fichier

```
$poylgone->asXML('fichier_polygone.xml') ;
```

Lire un fichier xml existant

```
<?php
header ("Content-Type: text/xml");
$maquette=simplexml_load_file('maquette.xml');
echo $maquette->asXML();
?>
```


Parcours

Accéder aux noeuds : ->

```
<?php
$maquette=simplexml_load_file('maquette.xml');
// accès au 1er noeud formation
$noeud_formation = $maquette->formation ;
echo $noeud_formation->libelle ,
?>
```

Remarque

- `$maquette->formation` retourne la première formation
- `$maquette` représente la racine. Dans notre cas la balise 'formations'

Parcours

Accéder aux noeuds : ->

```
<?php
$maquette=simplexml_load_file('maquette.xml');
// accès au 1er noeud formation
$noeud_formation = $maquette->formation ;
echo $noeud_formation->libelle ,
?>
```

Remarque

- **\$maquette->formation** retourne la première formation
- **\$maquette** représente la racine. Dans notre cas la balise 'formations'

Parcours

Avec **foreach**

```
// parcours de chaque semestre de formation
foreach ( $noeud_formation->semestre as $semestre)
{
    echo '<hr/>' ;
    //parcours chaque UE de semestre
    foreach ( $semestre->UE as $UE)
    { echo $UE->libelle . '<br/>' ;}
}
?>
```

Rappel

- Les chaînes sont stockées en UTF8 par SimpleXML
- Utilisez la fonction `utf8_decode()` pour les convertir en latin-1 si l'encodage de votre page html est latin-1

Parcours

Avec **foreach**

```
// parcours de chaque semestre de formation
foreach ( $noeud_formation->semestre as $semestre)
{
    echo '<hr/>' ;
    //parcours chaque UE de semestre
    foreach ( $semestre->UE as $UE)
    { echo $UE->libelle . '<br/>' ;}
}
?>
```

Rappel

- Les chaînes sont stockées en UTF8 par SimpleXML
- Utilisez la fonction `utf8_decode()` pour les convertir en latin-1 si l'encodage de votre page html est latin-1

Parcours

de tous les nœuds avec la méthode **children**

```
foreach($noeud_formation->children() as $nom_element=>
    $valeur_element)
echo $nom_element . '<br/>';
```

- On affiche tous les noms des éléments de l'élément formation
- Pas uniquement les semestres

Parcours

Accès aux attributs

- les attributs sont stockés dans un tableau associatif
- On accède à un attribut donné par : `$nom_noeud['nom_attribut']`
- ou à la liste des attributs par : `$nom_noeud->attributes`

Exemple

```
foreach ( $noeud_formation->semestre as $semestre )
{
    echo 'Semestre ' . $semestre['ID'];
}
```

```
foreach($UE->attributes() as $nom_attribut =>
    $val_attribut) {
    echo $nom_attribut . '=' . $val_attribut . "<br/>";
}
```

[► Voir exemple](#)

Parcours

Accès aux attributs

- les attributs sont stockés dans un tableau associatif
- On accède à un attribut donné par : `$nom_noeud['nom_attribut']`
- ou à la liste des attributs par : `$nom_noeud->attributes`

Exemple

```
foreach ( $noeud_formation->semestre as $semestre)
{
echo 'Semestre '. $semestre['ID'];
}
```

```
foreach($UE->attributes() as $nom_attribut =>
    $val_attribut) {
    echo $nom_attribut . '=' . $val_attribut . "<br/>";
}
```

► Voir exemple

Parcours

Accès aux attributs

- les attributs sont stockés dans un tableau associatif
- On accède à un attribut donné par : `$nom_noeud['nom_attribut']`
- ou à la liste des attributs par : `$nom_noeud->attributes`

Exemple

```
foreach ( $noeud_formation->semestre as $semestre)
{
echo 'Semestre '. $semestre['ID'];
}
```

```
foreach($UE->attributes() as $nom_attribut =>
    $val_attribut) {
    echo $nom_attribut. '=' . $val_attribut. '<br/>';
}
```

[► Voir exemple](#)

Construction d'un document XML

Création d'un fichier avec la balise root obligatoire (ici 'classe');

```
$racine = new SimpleXMLElement('<classe/>');
```

Ajout d'éléments enfants

```
$racine->addChild('eleve', 'Paul Dupont');  
$racine->addChild('eleve', 'Eric Durant');
```

Représentation textuelle

```
echo $racine->asXML();
```

Construction d'un document XML

À partir d'une chaîne de caractères

```
$str= '<polygone><point><x>2</x><y>3</y></point>  
<point><x>2</x><y>3</y></point></polygone>' ;  
$poylgone = new SimpleXMLElement($str);
```

Xpath

On utilise la méthode **xpath()** qui retourne un tableau.

Exemple :

```
$maquette=simplexml_load_file('maquette.xml');  
foreach ($maquette->xpath('//libelle') as $lib)  
{ echo $lib . '<br/>';}
```

Affiche la valeur textuelle de tous les libellés.

Autres fonctionnalités

Ajouter un attribut

```
$nom_attribut = 'couleur' ;  
$val_attribut = 'rouge' ;  
$polygone->addAttribute($nom_attribut , $val_attribut) ;
```

Supprimer un attribut

```
Unset($polygone[ 'couleur' ]) ;
```
