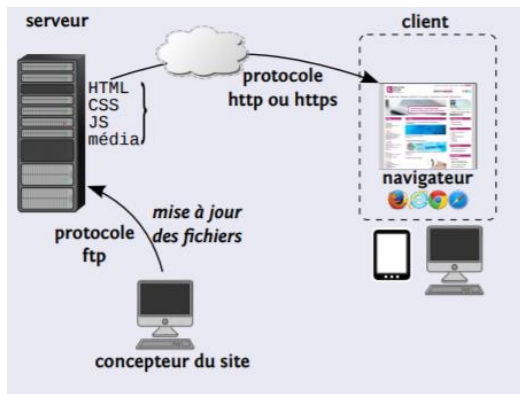


Notes du Cours 1

PHP

Modèle client-serveur :



- Les fichiers sont situés sur un serveur
- Le client y accède par le réseau. Il utilise le protocole HTTP ou HTTPS.
- Le développeur dispose d'un accès spécifique pour déposer les fichiers : protocole FTP, accès authentifié

FTP : File Transfert Protocol

Générer des pages côté serveur :

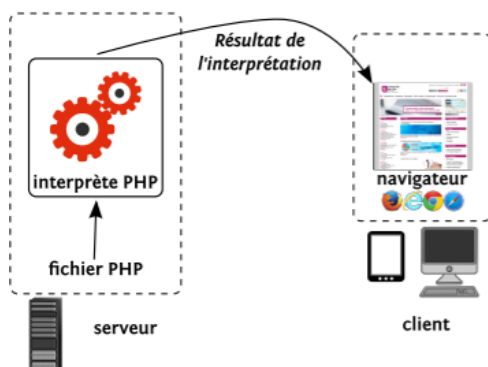
Une page écrite directement en (X)HTML est statique.

Les pages doivent évoluer en fonction :

- Du temps
- Des données disponibles sur le serveur
- De ce que souhaite consulter le visiteur

Programmation :

→ Le serveur ne contient pas la page HTML mais le programme qui la fabrique. (Elle est engendrée à chaque fois qu'un client la demande)



Identificateurs :

Les noms de variables :

- Commencent par \$
- **Distinction entre majuscule et minuscule**

Les noms de fonctions, constantes, méthodes :

- Pas de \$ initial
- PAS de distinction entre majuscule et minuscule

⇒ maFonction() eq à mafonction()

Variables :

- Il n'y a **pas de déclaration** : les variables sont créées à leur première apparition.
- Il n'y a **pas de typage explicite** : le type d'une variable est déterminé par la valeur qu'elle contient.
- Une même variable peut tantôt contenir un entier, tantôt une chaîne, etc...
- Mais les **valeurs** ont un type : booléen, entier, flottant, chaîne, objet ...

Les types scalaires :

-**Booléens** : Boolean TRUE et FALSE

-**entier** : int ou integer

- decimal : commence par un chiffre
- octal : commence par 0 puis chiffre
- hexa : commence par 0x

-**Flottants** : double ou float

-**Chaînes** : string

7/2 vaut 3.5 : flottant

6/3 : entier

6.0/3 : flottant

RQ :

Les entiers trop grands ou trop petits sont automatiquement convertis en flottants.

Int : codé sur 32 bits $[-2^{31}, 2^{31}-1]$

(int) (-)2.7 = 2 (int)2.1 = 2

Les tables :

Ce sont plutôt des listes ou des tables de hachage.

- Longueur dynamique
- On peut indexer par des entiers non consécutifs
- On peut indexer par des chaînes.
- L'ordre de rangement est **l'ordre dans lequel la table a été définie.**

Les fonctions :

- Les params commencent par \$
- Possibilité de donner des valeurs par défaut aux paramètres
- **Pas d'accès automatiques aux variables globales**

Les paramètres par défaut :

- Précédé par '='
- Peuvent alors être omis à l'appel
- Quand on définit une valeur par défaut pour un paramètre il faut le faire pour **tous les suivants.**

Notes du Cours 1

PHP

Les objets et les classes :

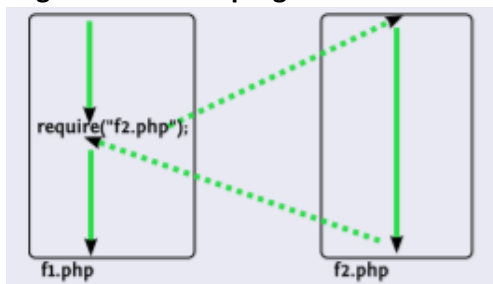
- Le constructeur unique s'appelle **__construct()**
- Dans la classe **\$this** désigne l'objet lui-même
- Usage de **\$this** obligatoire pour accéder aux attributs depuis les méthodes

Syntaxe :

- La désignation d'un composant (méthode ou attribut) utilise " -> " pas le point
- Les attributs commencent par \$ dans la déclaration. Mais le \$ disparaît quand l'attribut est précédé de " -> "

```
class Fraction {
    private $numérateur = 0;
    private $dénominateur = 1;
    public function numérateur() { //accesseur
        return $this->numérateur;
    }
    public function dénominateur() { /* à compléter...*/ }
    public function enNumerique() {
        return $this->numérateur / $this->dénominateur;
    }
    function __construct($num = 0, $den = 1) {
        if ($den == 0)
            throw new Exception('Dénominateur nul');
        $p = pgcd($num, $den);
        $this->numérateur = $num / $p;
        $this->dénominateur = $den / $p;
    }
} // fin de la classe Fraction
```

Organisation d'un programme :



L'instruction return :

- Dans une fonction return provoque l'arrêt de la fonction et fournit le résultat.
- Hors d'une fonction : return provoque l'arrêt du script du fichier courant. Si celui-ci a été appelé par un require (ou include) l'interprétation du script appelant continue.

L'instruction exit :

L'instruction exit() met fin à l'exécution du script et des scripts appelants.