

Construire un site dynamique

Partie 1



Sommaire



- Structure et organisation d'un site dynamique
 - Organisation des répertoires
 - Concept DRY
 - Les inclusions
 - Les fonctions
- Planification
 - Le parcours utilisateur
 - Paramètres d'URI
 - Planification des fichiers
- Relations entre tables.

Structure et organisation d'un site dynamique



L'organisation d'un site dynamique est très importante:

- Pour le travail en équipe.
- Pour le développement.
- Pour le référencement.
- Pour la maintenance.
- Pour les futures évolutions.

Organisation des répertoires



Une bonne organisation des répertoire est importante pour s'y retrouver.

Il faut séparer les fichiers:

- Par type (CSS, JS, images, ...).
- Par fonctionnalités (includes, templates, ...).

Une bonne structure pensée dès le départ permettra un développement facilité, mais aussi une maintenance plus rapide et des évolutions plus souples à mettre en œuvre.

Exercice pratique:
Imaginer la structure de votre future
application.



Concept DRY



DRY: Do not Repeat Yourself!

Afin d'éviter les répétitions, il faut décomposer son projet en petites "briques", plus faciles à développer et aussi à planifier. Ces petites entités pourront ensuite donner lieu à des optimisations afin d'éviter les "copies de code":

- Création de fonctions PHP.
- Création de pages types.
- Création de templates.
- ...

Inclusions

Les fonctions PHP include, require, include_once, require_once permettent de mutualiser du code (fonctions, connexion BDD, paramétrage, ...), mais aussi des parties de pages (templates).

```
<?php
$titrePage = "page dynamique";
Include "header.php"; //header contient les entêtes HTML
etc.
...
Include "footer.php"; //contient le bas de page
?>
```

Exercice pratique:
Imaginer les fichiers à inclure dans
votre projet (templates et outils).



Fonctions



Les fonctions permettent de mutualiser et réutiliser à volonté des actions de base comme:

- Les conversions de dates (FR->US ou US->FR).
- Effectuer des calculs, des moyennes, ...
- Effectuer de la mise en page (conversion tableau PHP en HTML).
- Gérer la sécurité, les droits.
- Gérer les logs.
- ...

Exercice pratique:

Imaginer les fonctions dont vous pourrez avoir besoin dans votre future application.



Une application ergonomique répond aux besoins spécifiques d'un utilisateur spécifique. Elle se définit par son utilité et par la facilité avec laquelle on s'en sert :

- Pourquoi est-elle faite ?
- A qui est-elle destinée ?
- Comment fonctionne-t-elle ?

Ce sont ces éléments qui vont permettre de concevoir une ergonomie pertinente, adaptée aux utilisateurs et à leurs objectifs. Elle doit permettre d'atteindre facilement et rapidement l'information recherchée

Parcours utilisateur



- Le parcours utilisateur est le parcours emprunté par l'utilisateur sur une interface pour atteindre son objectif. Il est constitué de différentes étapes, suivies grâce à un menu de navigation et des systèmes de navigation transverses.
- Chaque utilisateur utilise une application web avec un objectif précis comme celui de customiser sa cuisine, de trouver des informations ou de jouer à un jeu vidéo, etc. Un parcours utilisateur efficace facilite la tâche de l'internaute en le guidant de façon rapide et intuitive jusqu'à l'objectif qu'il poursuit. On limite le nombre d'étapes, on limite le nombre d'erreurs possibles, on fait en sorte de le guider vers le contenu qu'il lui faut.

Exercice pratique:

Lister les différentes étapes pour atteindre l'information sur votre future application.



Paramètres d'URI

➤ Quand on soumet un formulaire avec la méthode GET, les infos sont attachées à l'URI. On peut aussi passer manuellement des informations par l'URI afin de les utiliser dans nos pages.

➤ L'utilisation de paramètres dans l'URI permet de rendre une page dynamique et paramétrable:

Ex: <http://localhost/afficheNotes.php?eleve=5> permettra d'afficher les notes de l'élève dont l'id est 5 dans la table élèves.

➤ On peut aussi passer plusieurs paramètres par l'URI:

<http://localhost/afficheNotes.php?matiere=1&eleve=5> permettra d'afficher les notes de l'élève d'id 5 pour la matière d'id 1.

Exercice pratique:

Etablir la liste des pages que vous pouvez réutiliser dans votre application et la liste des paramètres à passer par l'URI.



Planification



- Une fois que l'on a décomposé son projet en petites "briques", que l'on a estimé:
 - les inclusions
 - les fonctions
 - l'architecture sur le disque
 - Les parcours utilisateurs
 - Les paramètres URI.
 - ...

on peut passer à la phase importante de la planification.

Planification



Cette phase va permettre de définir:

- Le planning du projet.
- Le planning de chaque développeur.
- La date de livraison.
- Les coûts de développement.
- Les ressources à trouver.
- ...

Exercice pratique:

Etablir un premier jet de planning pour votre projet.



Relations entre tables



On peut établir des relations de cardinalité entre les tables d'une base de données. La cardinalité est le nombre de participation d'une entité à une relation.

Ces cardinalités peuvent être:

➤ 1 : 1

➤ 1 : n

➤ n : n

Relations entre tables

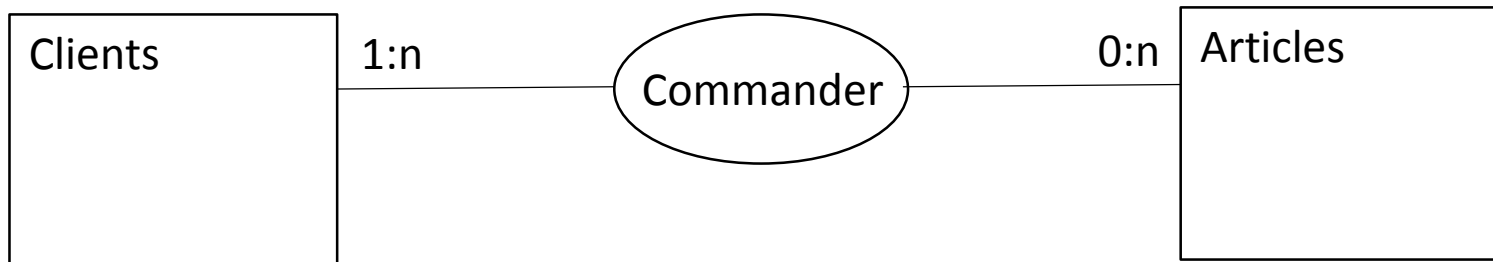
Pour définir la cardinalité il faut se poser les questions:

- Combien de fois au minimum un client peut-il commander un article?
- Combien de fois au maximum un client peut-il commander un article?

Pour être client, un client doit au moins commander un article. Il peut commander plusieurs articles.

Idem avec les articles:

- Un article peut ne pas être commandé.
- Un article peut être commandé plusieurs fois.



Relations entre tables cardinalité 1 à 1



Imaginons qu'à la webforce3 un ordinateur soit affecté de façon officielle à chaque élève :

Un ordinateur est affecté à un seul élève et une élève n'a qu'un seul ordinateur affecté.

Dans ce cas là, la table élève et la table ordinateur ont une cardinalité dite 1:1

Relations entre tables cardinalité 1 à plusieurs



A la webforce3 les élèves sont répartis par école :

Une école peut recevoir plusieurs élèves mais chaque élève n'est inscrit que dans une seule classe.

Dans ce cas là, la table élève et la table école ont une cardinalité dite 1:n (1 pour la table élève et n pour la table école).

Relations entre tables cardinalité plusieurs à plusieurs



A la webforce3 les élèves travaillent seuls ou en équipe :

Un seul code PHP peut être écrit par plusieurs élèves (travail collaboratif) et un élève peut écrire plusieurs codes PHP.

Dans ce cas là, la table élève et la table codePHP ont une cardinalité dite n:n (n pour la table élève et n pour la table école).

Exercice pratique:

Dessiner les tables de votre projet et imaginer les cardinalités entre tables.



Exercices pratiques

