

Programmation Web Serveur

Envoi de fichiers, Architecture logicielle

D'après les cours de Philippe Renevier



Fabien Givors

Université de Nice Sophia Antipolis

Département Informatique

fabien.givors@unice.fr

Entrer des données depuis un navigateur

FORMULAIRE

PRINCIPES

- Envoi de données
 - depuis le navigateur
 - Pour traitement dans le serveur
- Deux façons de faire
 - Passage des informations dans l'url (méthode GET, déjà utilisée)
 - Passage des informations dans la requête (méthode POST)
 - Attention : ce n'est pas sécurisé

EXEMPLE: CÔTÉ NAVIGATEUR / HTML

```
<form action="page.php" method="post">  
<fieldset>  
  <legend>Placez une alerte</legend>  
  <p>Message de l'alerte : <input type="text" name="alerte" /></p>  
  <input type="submit" name="setalerte" value="declenchez l'alerte" />  
</fieldset>  
</form>
```

The diagram illustrates the mapping between the HTML code above and the rendered browser form below. Orange arrows show the following connections:

- The `<form>` tag maps to the overall form container.
- The `<legend>` tag maps to the text "Placez une alerte".
- The `<p>` tag maps to the text "Message de l'alerte :".
- The `<input type="text" name="alerte" />` tag maps to the text input field.
- The `<input type="submit" name="setalerte" value="declenchez l'alerte" />` tag maps to the "declenchez l'alerte" button.

CÔTÉ SERVEUR / PHP

```
<form action="page.php" method="post">
<fieldset>
  <legend>Placez une alerte</legend>
  <p>Message de l'alerte : <input type="text" name="alerte" /></p>
  <input type="submit" name="setalerte" value="declenchez l'alerte" />
</fieldset>
</form>
```

```
if (isset($_POST["setalerte"]))
{
  $alerte = rawurlencode(trim($_POST["alerte"]));
  file_put_contents("alerte.txt", $alerte);
}
```

BALISES DE FORMULAIRE : FORM

- Contient des éléments de contrôle de formulaire (bouton, champs, etc.)

- « block » (sauf form) ou script

- Attributs

- action (uri)

- method ("get" ou "post")

- get : envoi dans l'url des paires key/value : ? toto=val&titi=val2&...

- post : envoi

- enctype (pour une méthode "post")

- Par défaut : **application/x-www-form-urlencoded** - encodage : espace devient + et les autres non alphanumériques %HH et les retours à la ligne : "CR LF" (i.e., '%0D%0A')

- **multipart/form-data** - envoi en différentes parties (types à préciser à la source)

- accept-charset (liste - , - d'encodage possible pour les caractères acceptés par le server)

- Accept (liste - , - de types de contenu acceptés par le server)

- events : onsubmit et onreset

BALISES DE FORMULAIRE : TYPE DE INPUT

- ▣ text : champs d'entrée de texte.
- ▣ password : l'écho sont des '*'. sécurité pauvre.
- ▣ checkbox
- ▣ radio (radiobutton)
- ▣ submit : un bouton pour envoyer
- ▣ image : un bouton submit graphique. Attribut src donne l'URI de l'image. Utiliser l'attribut alt. Les coordonnées du clic sont passés au server sous la forme name.x et name.y
 - ▣ problème d'accessibilité : navigateur non graphique, clic difficile, etc.
 - ▣ à remplacer par plusieurs boutons submit ou par des scripts côté client.
- ▣ reset (bouton).
- ▣ button : bouton sans comportement prédéfini (script)
- ▣ hidden : champs caché (parfois utile pour passer une valeur masquée)
- ▣ file : sélection d'un fichier

BALISES DE FORMULAIRE : INPUT

- ▣ Balise vide
- ▣ Attributs
 - ▣ type
 - ▣ name : nom de contrôle (très important)
 - ▣ value (valeur initiale ou libellé) : optionnel sauf pour radio et checkbox
 - ▣ size (en pixel sauf pour text et password où c'est un nombre de caractère)
 - ▣ maxlength : pour text ou password : nombre de caractères maximum
 - ▣ checked : pour radio et checkbox
 - ▣ src : pour image : la source (ne pas oublier alt)

BALISES DE FORMULAIRE : SELECT

select : menu

- (optgroup | option)+

- attributs

 - name : nom de contrôle

 - size (nombre) : nombre d'éléments visibles pour une scroll list

 - multiple (pas de valeur) : permet la sélection multiple

option

- #pcdata (texte)

- attributs

 - selected : pour présélectionner l'élément

 - value (texte) : pour donner une valeur autre que le texte (#pcdata)

 - label (texte) : pour faire apparaître un autre nom (plus court) à la charge du navigateur !! (pas sûr que cela fonctionne !!)

optgroup

- regrouper les options : (option)+

- attribut : label (texte) : libellé

BALISES DE FORMULAIRE : TEXTAREA

- Champs d'entrée sur plusieurs lignes
- textarea
 - #PCDATA : texte initiale
 - Attributs
 - name : nom de contrôle
 - cols : nombre de colonne
 - rows : nombre de ligne

BALISES DE FORMULAIRE : LABEL

- Permet d'associer un texte à un élément de formulaire sans texte
 - inline
 - *Start tag: **required**, End tag: **required***
- Attaché par l'attribut for
 - Valeur = id d'un champ de contrôle

EXEMPLE DE FORMULAIRE

```
<form action="" method="post" id="ajouterarticle">
<fieldset>
  <legend>Ajouter une news</legend>

  <select name="categorie">
    <option value='france' >france</option>
    <option value='monde' >monde</option>
    <option value='politique' >politique</option>
    <option value='sports' >sports</option>
  </select><br />

  <label for="titre">titre : </label><input type="text" name="titre" value="" /><br />
  <label for="auteur">auteur : </label><input type="text" name="auteur" value=""
/><br />
  <label for="texte">texte : </label><textarea name="texte"></textarea ><br />
  <input type="submit" /></fieldset>
</form>
```

FORMULAIRES : RÉCEPTION EN PHP

- Page qui reçoit le formulaire : attribut « action » du form
- Valeur(s) accessible(s) par :
 - tableau associatif : index est l'attribut « name » de l'input
 - `$_POST` ou `$_GET` sont des « superglobales »
- Un peu de sécurité
 - `trim(htmlspecialchars(addslashes()))`
 - `str_replace` pour remplacer des caractères « spéciaux »
 - Tests complémentaires...
- Les valeurs peuvent être des tableaux (si le name de l'input est du style `nom[]`)

EXEMPLE DE RÉCEPTION DE FORMULAIRE

```
// si toutes les valeurs du formulaire sont bien présentes
if (isset($_POST["titre"],$_POST["auteur"],$_POST["texte"],$_POST["categorie"]))
{
    // et si elles sont toutes remplies avec quelques choses
    if ($_POST["titre"] && $_POST["auteur"]
        && $_POST["texte"] && $_POST["categorie"])
    {
        // on prépare le contenu de la nouvelle / article
        $article = trim($_POST["titre"])."\n";
        $article .= trim($_POST["auteur"])."\n";
        $article .= trim($_POST["texte"]);
        // il faut déterminer un nom de fichier
        $nom Fichier = "news/" . $_POST["categorie"] . "/" . $_POST["titre"] . ".news";
        // si le fichier n'existe pas...
        if (!file_exists($nom Fichier) )
        {
            // ... il est créé
            file_put_contents($nom Fichier, $article);
        }
    }
}
```

TÉLÉCHARGEMENT D'UN FICHER

- Méthode post
 - Le champs caché MAX_FILE_SIZE (mesuré en octets) doit précéder le champ input de type file et sa valeur représente la taille maximale acceptée du fichier.
- Variable globale \$_FILES (ci-dessous, 'userfile' est l'attribut «name» donné en html à la balise input de type file)
 - \$_FILES['userfile']['name'] : Le nom original du fichier, tel que sur la machine du client web.
 - \$_FILES['userfile']['type'] : Le type MIME du fichier, si le navigateur a fourni cette information. Par exemple, cela pourra être "image/gif".
 - \$_FILES['userfile']['size'] : La taille, en octets, du fichier téléchargé.
 - \$_FILES['userfile']['tmp_name'] : Le nom temporaire du fichier qui sera chargé sur la machine serveur.
 - \$_FILES['userfile']['error'] : Le code d'erreur error code associé au téléchargement de fichier. Cet élément a été introduit en PHP 4.2.0

LES CODES ERREURS

- ▣ `UPLOAD_ERR_OK`
 - ▣ Valeur : 0. Aucune erreur, le téléchargement est correct.
- ▣ `UPLOAD_ERR_INI_SIZE`
 - ▣ Valeur : 1. Le fichier téléchargé excède la taille de `upload_max_filesize`, configuré dans le `php.ini`.
- ▣ `UPLOAD_ERR_FORM_SIZE`
 - ▣ Valeur : 2. Le fichier téléchargé excède la taille de `MAX_FILE_SIZE`, qui a été spécifiée dans le formulaire HTML.
- ▣ `UPLOAD_ERR_PARTIAL`
 - ▣ Valeur : 3. Le fichier n'a été que partiellement téléchargé.
- ▣ `UPLOAD_ERR_NO_FILE`
 - ▣ Valeur : 4. Aucun fichier n'a été téléchargé.
- ▣ `UPLOAD_ERR_NO_TMP_DIR`
 - ▣ Valeur : 6. Un dossier temporaire est manquant. Introduit en PHP 4.3.10 et PHP 5.0.3.
- ▣ `UPLOAD_ERR_CANT_WRITE`
 - ▣ Valeur : 7. Échec de l'écriture du fichier sur le disque. Introduit en PHP 5.1.0.
- ▣ Note : Ces constantes sont apparues en PHP 4.3.0.

TÉLÉCHARGEMENT D'UN FICHER

- `is_uploaded_file ($filename)`
 - retourne TRUE si le fichier filename a été téléchargé par HTTP POST. Cela est très utile pour vous assurer qu'un utilisateur n'essaie pas d'accéder intentionnellement à un fichier auquel il n'a pas droit (comme /etc/passwd).
 - Pour un fonctionnement correct, la fonction `is_uploaded_file()` nécessite un argument comme `$_FILES['userfile']['tmp_name']`
 - le nom du fichier téléchargé sur la machine cliente `$_FILES['userfile']['name']` ne fonctionne pas.
- `move_uploaded_file ($filename, $destination)`
 - vérifie que le fichier filename est un fichier téléchargé par HTTP POST. Si le fichier est valide, il est déplacé jusqu'à destination et retourne TRUE
 - Si filename n'est pas valide, rien ne se passe, et `move_uploaded_file()` retournera FALSE.
 - Si filename est un fichier téléchargé, mais que pour une raison quelconque, il ne peut être déplacé, rien ne se passe, et `move_uploaded_file()` retourne FALSE. De plus, une alerte sera affichée.

EXEMPLE CÔTÉ HTML – UN FICHER

```
<form action="acceptefichiers.php" method="post"
  enctype="multipart/form-data">
  <div>
    <input type="submit" value="Envoi!" /> <br /> <div
  id="fichiers">
<!-- MAX_FILE_SIZE doit precéder le champs input de type
file -->
  input type="hidden" name="MAX_FILE_SIZE"
  value="30000" />

  fichier &agrave; t&eacute;l&eacute;charger : <input
type="file" name="fichier" />
  </div>
</form>
```

EXEMPLE CÔTÉ PHP – PLUSIEURS FICHIERS

```
$uploaddir = "dossierUpload/";
$uploadfile = $uploaddir . basename($_FILES['fichier']
    ['name']);
// !! droit en écrire nécessaire sur le dossier $uploaddir !!
if (move_uploaded_file($_FILES['fichier']['tmp_name'],
    $uploadfile))
    {
    // telechargement ok, placement du fichier à l'endroit voulu
    }
else
    {
    // échec dans le téléchargement
    }
```

EXEMPLE CÔTÉ HTML - PLUSIEURS FICHIERS

```
<form action="acceptefichiers.php" method="post"
  enctype="multipart/form-data">
  <div>
    <!-- MAX_FILE_SIZE doit precéder le champs input de type
    file -->
    <input type="hidden" name="MAX_FILE_SIZE" value="30000" />
    fichiers &agrave; t&eacute;l&eacute;charger : <br />
    <input type="file" name="fichiers[]" /><br />
    <input type="file" name="fichiers[]" /><br />
    <input type="file" name="fichiers[]" /><br />
    <input type="file" name="fichiers[]" /><br />
    <input type="file" name="fichiers[]" /><br />
    <input type="submit" value="Envoi!" />
  </div>
</form>
```

EXEMPLE CÔTÉ PHP – PLUSIEURS FICHIERS

```
foreach ($_FILES['fichiers']['tmp_name'] as $key =>
    $tmp_file) {
    $uploadfile = $uploaddir . basename($_FILES['fichiers']
        ['name'][$key]);
    if (move_uploaded_file($tmp_file, $uploadfile))
        {
        // telechargement ok, placement du fichier à l'endroit
        voulu
        }
    else
        {
        // échec dans le téléchargement
        }
    }
}
```

Variable \$_SERVER

VERS DU CODE PHP PLUS GÉNÉRIQUE

`$_SERVER`

- <http://fr2.php.net/reserved.variables.server.php>
- `$_SERVER` est un tableau contenant des informations comme les en-têtes, dossiers et chemins du script.
- Les entrées de ce tableau créées par le serveur web.
- Aucune garantie que tous les serveurs les rempliront tous ; certains en oublieront quelques-unes et en rajouteront de nouvelles non mentionnées ici.
Cependant, un grand nombre de ces variables fait partie des [» spécifications CGI/1.1](#), et vous pouvez donc vous attendre à les retrouver.

\$_SERVER : EXEMPLE SUR UN SERVER LOCAL (1/2)

Array (

[MIBDIRS] => D:/xampp-1.7/php/extras/mibs

[MYSQL_HOME] => \xampp\mysql\bin

[OPENSSL_CONF] => D:/xampp-1.7/apache/bin/openssl.cnf

[PHP_PEAR_SYSCONF_DIR] => \xampp\php

[PHPRC] => \xampp\php

[TMP] => \xampp\tmp

[HTTP_HOST] => localhost

[HTTP_CONNECTION] => keep-alive

[HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

[HTTP_USER_AGENT] => Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.31 (KHTML, like Gecko)

Chrome/26.0.1410.64 Safari/537.31

[HTTP_REFERER] => http://localhost/option1/workspace/TP08-05/categorie.php

[HTTP_ACCEPT_ENCODING] => gzip,deflate,sdch

[HTTP_ACCEPT_LANGUAGE] => en-US,en;q=0.8,fr;q=0.6

[HTTP_ACCEPT_CHARSET] => ISO-8859-1,utf-8;q=0.7,*;q=0.3

[HTTP_COOKIE] => PHPSESSID=kbpa5h176b347s9qjihssk8055

[PATH] => C:\Program Files\Common Files\Microsoft Shared\Windows Live;C:\Program Files (x86)\Common Files\Microsoft Shared\Windows Live;C:\Program Files\Dell\DW WLAN Card;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\Windows Live\Shared;D:\Program Files\TortoiseGit\bin;C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL Server\100\DTS\Binn\;D:\Program Files (x86)\MiKTeX 2.9\miktex\bin\;D:\Program Files\TortoiseSVN\bin

[SystemRoot] => C:\Windows

[COMSPEC] => C:\Windows\system32\cmd.exe

[PATHEXT] => .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC

[WINDIR] => C:\Windows

[...]

)

\$_SERVER : EXEMPLE SUR UN SERVER LOCAL (2/2)

```
Array (
  [...]
  [SERVER_SIGNATURE] =>
Apache/2.2.21 (Win32) mod_ssl/2.2.21 OpenSSL/1.0.0e PHP/5.3.8 mod_perl/2.0.4 Perl/v5.10.1 Server at localhost
Port 80

  [SERVER_SOFTWARE] => Apache/2.2.21 (Win32) mod_ssl/2.2.21 OpenSSL/1.0.0e PHP/5.3.8 mod_perl/2.0.4
Perl/v5.10.1
  [SERVER_NAME] => localhost
  [SERVER_ADDR] => 127.0.0.1
  [SERVER_PORT] => 80
  [REMOTE_ADDR] => 127.0.0.1
  [DOCUMENT_ROOT] => E:/local
  [SERVER_ADMIN] => postmaster@localhost
  [SCRIPT_FILENAME] => E:/local/option1/workspace/TP08-05/apropos.php
  [REMOTE_PORT] => 54998
  [GATEWAY_INTERFACE] => CGI/1.1
  [SERVER_PROTOCOL] => HTTP/1.1
  [REQUEST_METHOD] => GET
  [QUERY_STRING] =>
  [REQUEST_URI] => /option1/workspace/TP08-05/apropos.php
  [SCRIPT_NAME] => /option1/workspace/TP08-05/apropos.php
  [PHP_SELF] => /option1/workspace/TP08-05/apropos.php
  [REQUEST_TIME] => 1366889160
)
```

\$_SERVER : EXEMPLE SUR UN SERVER

Array

(

[HTTP_HOST] => deptinfo.unice.fr

[HTTP_USER_AGENT] => Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31

[HTTP_REFERER] => <http://adresse.origine/de/la/page/>

[REMOTE_ADDR] => 134.59.9.210

[QUERY_STRING] =>

[REQUEST_URI] => /progwebserveur/tps/apropos.php

[SCRIPT_NAME] => /progwebserveur/tps/apropos.php

[PHP_SELF] => /progwebserveur/tps/apropos.php

[REQUEST_TIME] => 1366889569

)

`$_SERVER` ET L'URL / LE SCRIPT (1/2)

- Exemple avec `tp02.php?dossier=politique`
 - `["QUERY_STRING"] => dossier=politique`
 - `["REQUEST_URI"] => /option1/workspace/tp02.php?dossier=politique`
 - `["SCRIPT_NAME"] => /option1/workspace/tp02.php`
 - `["PHP_SELF"] => /option1/workspace/tp02.php`
- Exemple avec `tp02.php?indice=2.news&dossier=politique`
 - `["QUERY_STRING"] => indice=2.news&dossier=politique`
 - `["REQUEST_URI"] => /option1/workspace/tp02.php?indice=2.news&dossier=politique`
 - `["SCRIPT_NAME"] => /option1/workspace/tp02.php`
 - `["PHP_SELF"] => /option1/workspace/tp02.php`

\$_SERVER ET L'URL / LE SCRIPT (2/2)

- Exemple avec /index.php

```
["QUERY_STRING"] =>
```

```
["REQUEST_URI"] => /option1/workspace/index.php
```

```
["SCRIPT_NAME"] => /option1/workspace/index.php
```

```
["PHP_SELF"] => /option1/workspace/index.php
```

- Exemple avec /

```
["QUERY_STRING"] =>
```

```
["REQUEST_URI"] => /option1/workspace/
```

```
["SCRIPT_NAME"] => /option1/workspace/index.php
```

```
["PHP_SELF"] => /option1/workspace/index.php
```

- Différences !!

- Adresse / fichier de la page (pas des fichiers inclus)

UTILISATION

- Permet de savoir la page courante...
 - Mettre en valeur dans le menu la page en cours
 - Attention aux paramètres
 - Ordre
 - Plus ou moins de paramètres
- Gestion progressive
 - D'abord sans gérer les paramètres (égalité simple entre une adresse et `$_SERVER["PHP_SELF"]`)
 - Puis traitement des paramètres...

VERS L'ARCHITECTURE LOGICIELLE

QUESTION COMMENT « AUTOMATISER » LA MISE EN VALEUR DE LA PAGE COURANTE

The image shows two screenshots of a website interface. The top screenshot shows a breadcrumb trail: [Accueil](#) > [Catégories](#) > [monde](#). A red circle highlights the 'Catégories' dropdown menu, which lists: [France](#), [monde](#), [politique](#), and [sports](#). The bottom screenshot shows the same breadcrumb trail: [Accueil](#) > [Catégories](#) > [monde](#). A red circle highlights the 'monde' link in the breadcrumb trail, and a red arrow points from the 'monde' link in the top screenshot to this 'monde' link. The article title is 'Le pape démissionnera le 28/02/2013'.

[Accueil](#) > [Catégories](#) > [monde](#)

[Accueil](#) > [Catégories](#) > [monde](#)

[France](#)
[monde](#)
[politique](#)
[sports](#)

[Ecologie](#)
[France](#)
[media](#)
[monde](#)
[politique](#)
[sports](#)

Le pape démissionnera le 28/02/2013

Benoît XVI a surpris le monde entier en annonçant lundi son choix de démissionner de son pontificat, pour des raisons respectées par les catholiques.

La démission du pape Benoît XVI a pris le monde entier au dépourvu, ouvrant la voie à toutes les spéculations sur son départ fin février, une première dans l'histoire de l'Eglise moderne, au motif qu'il n'a "plus les forces" pour la diriger. Le pape a surpris les fidèles à travers le monde. Mais de nombreuses voix ont surtout exprimé du respect pour une décision d'un souverain pontife sera désigné "d'ici Pâques", a indiqué le Vatican.

Le pape démissionnera le 28/02/2013

Benoît XVI a surpris le monde entier en annonçant lundi son choix de démissionner de son pontificat, pour des raisons respectées par les catholiques.

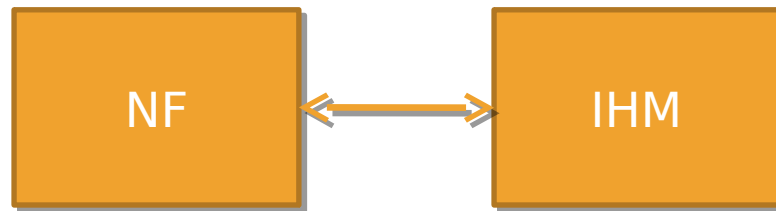
La démission du pape Benoît XVI a pris le monde entier au dépourvu, ouvrant la voie à toutes les spéculations sur son départ fin février, une première dans l'histoire de l'Eglise moderne, au motif qu'il n'a "plus les forces" pour la diriger. Le pape a surpris les fidèles à travers le monde. Mais de nombreuses voix ont surtout exprimé du respect pour une décision d'un souverain pontife sera désigné "d'ici Pâques", a indiqué le Vatican.

IHM ET ARCHITECTURE

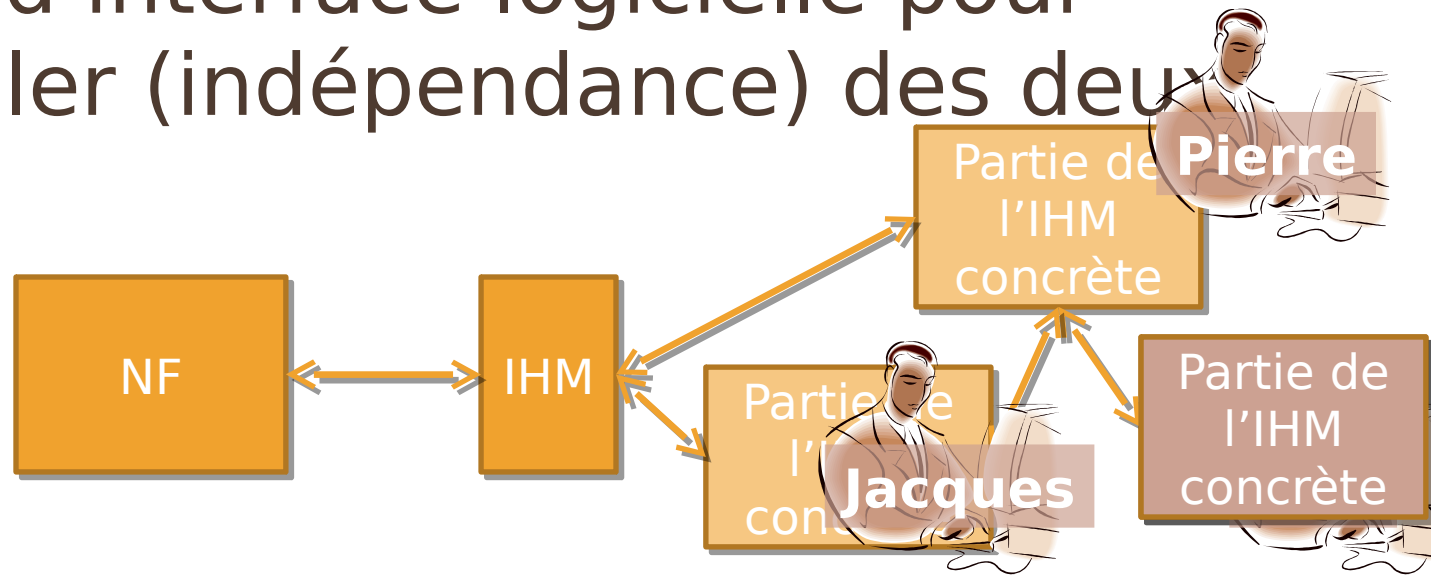
- Séparation possible
 - Code pour IHM
 - Code « Métier »
- Exemple
 - IHM différente pour une Gestion d'un stock de chaussures ou de bibelots ?
 - GoogleMaps sur android ou sur le web, le système change-t-il ?
- Objectif : éviter de tout modifier si on change la partie fonctionnelle ou la partie IHM

DÉCOUPAGE

- Exemple de Séparation IHM / NF
 - Noyau fonctionnel (ou partie Métier)



- Besoin d'interface logicielle pour découpler (indépendance) des deux parties

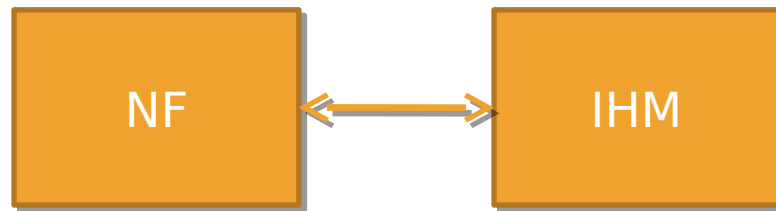


QU'EST CE QU'UNE INTERFACE LOGICIELLE?

- C'est juste la définition de comment utiliser un code
- Cela peut être
 - des fonctions,
 - des paramètres à ces fonctions,
 - des variables,
 - des éléments de configurations, etc.

DÉCOUPAGE

- Exemple de Séparation IHM / NF
 - Noyau fonctionnel (ou partie Métier)



- Besoin d'interface logicielle pour découpler (indépendance) des deux parties

