

Construire un panier

La création d'un site de vente en ligne passe par l'utilisation d'un panier virtuel dans lequel l'internaute pourra déposer des articles au gré de sa navigation dans le catalogue. Nous allons voir les bases fondamentales pour la création d'un tel panier.

Pour construire un panier, nous avons besoin essentiellement d'une variable de session. Il va donc de soi que chaque page du site doit commencer par :

```
<?php
session_start();
// ... suite du code...
?>
```

Ce que vous ne verrez pas ici

Il n'est pas question ici de la manière dont vous récupérez les valeurs dans une base de données. D'autres tutos font ça très bien.

Nous ne débordons pas non plus sur la construction du bon de commande ou le paiement en ligne.

De même nous n'aborderons pas la problématique des sessions en PHP, reportez-vous au [manuel PHP](#) et aux [tutoriaux](#) sur le sujet.

Nous ne verrons pas non plus comment coder les pages du catalogue ni les boutons d'ajout (ou de retrait) d'articles du panier. On ne voit que le panier, sa création et les manipulations qu'on peut faire avec.

Ce que vous devez connaître

Utilisation de tableaux associatifs et tableaux indexés. Une visite dans [le manuel PHP](#) peut s'avérer tout à fait utile.

Notez également un autre [tutorial basique sur la structure des tableaux indexés](#).

Création du panier

Pour notre panier, nous avons donc besoin d'une variable de session. Nous ne mettrons pas les valeurs en vrac dedans, cette variable doit être organisée. Nous allons donc la subdiviser et pour faire ça, il faut que notre variable soit un tableau. Première chose à créer, le panier global :

```
<?php
session_start();
/* Initialisation du panier */
$_SESSION['panier'] = array();
?>
```

Ça, c'est le contenant général de notre panier. On doit le subdiviser en compartiments avant d'y stocker quoique ce soit de manière organisée pour pouvoir trier son contenu et le manipuler. Là, on a trois parties à créer pour les quantités, les tailles et les prix.

```
<?php
session_start();
/* Initialisation du panier */
$_SESSION['panier'] = array();
/* Subdivision du panier */
$_SESSION['panier']['qte'] = array();
$_SESSION['panier']['taille'] = array();
$_SESSION['panier']['prix'] = array();
?>
```

Voilà, là, nous avons maintenant un panier avec trois compartiments pour y mettre nos articles. Mais on ne peut pas mettre nos articles en vrac, il faut pouvoir retrouver nos articles, les tailles correspondant exactement à chaque article ajouté et le prix aussi. Donc ce sont également des tableaux, mais si "panier" est un tableau associatif (donc avec des index nommées "qte", "taille" et "prix", chaque compartiment est un tableau indexé, c'est à dire que les index seront numérotés de 0 à n.

Prenons un cas : supposons que notre site soit un e-magasin de vêtements. J'ajoute dans mon panier 2 pantalons taille 56 d'une valeur de 84,95 euros : première chose qu'on va remarquer : il manque un identifiant permettant de dire de quel type d'article il s'agit . Rajoutons ce compartiment au tableau.

```
<?php
session_start();
/* Initialisation du panier */
$_SESSION['panier'] = array();
/* Subdivision du panier */
$_SESSION['panier']['id_article'] = array();
$_SESSION['panier']['qte'] = array();
$_SESSION['panier']['taille'] = array();
$_SESSION['panier']['prix'] = array();
?>
```

Maintenant, nous avons le point manquant. La référence de notre pantalon est "phlevis501". La première chose à faire avant d'ajouter dans le panier, c'est de vérifier que le panier existe, sinon, on le crée. Puis on ajoute dedans. Le code sera le suivant:

```
<?php
/* Démarrage ou prolongation de la session */
session_start();
/* Article exemple */
$select = array();
$select['id'] = "phlevis501";
$select['qte'] = 2;
```

```

$select['taille'] = "56";
$select['prix'] = 84.95;

/* On vérifie l'existence du panier, sinon, on le crée */
if(!isset($_SESSION['panier']))
{
    /* Initialisation du panier */
    $_SESSION['panier'] = array();
    /* Subdivision du panier */
    $_SESSION['panier']['qte'] = array();
    $_SESSION['panier']['taille'] = array();
    $_SESSION['panier']['prix'] = array();
}

/* Ici, on sait que le panier existe, donc on ajoute l'article dedans. */
array_push($_SESSION['panier']['id_article'],$select['id']);
array_push($_SESSION['panier']['qte'],$select['qte']);
array_push($_SESSION['panier']['taille'],$select['taille']);
array_push($_SESSION['panier']['prix'],$select['prix']);

/* Affichons maintenant le contenu du panier : */
?>
<pre>
<?php
var_dump($_SESSION['panier']);
?>
</pre>

```

Et le résultat affiché sera le suivant :

```

array(4) {
  ["id_article"]=>
  array(1) {
    [0]=>
    string(10) "phlevis501"
  }
  ["qte"]=>
  array(1) {
    [0]=>
    int(2)
  }
  ["taille"]=>
  array(1) {
    [0]=>
    string(2) "56"
  }
}

```

```

    ["prix"]=>
    array(1) {
        [0]=>
        float(84.95)
    }
}

```

Notez bien que le prix qui est dans la variable `$_SESSION['panier']['prix'][0]` correspond au prix unitaire de l'article. Donc pour avoir le prix total du panier, on devra commencer par multiplier le prix par la quantité puis ensuite additionner les articles.

Ajouter d'autres articles

Exactement de la même manière que le premier a été ajouté, on continue avec la fonction [array_push\(\)](#) comme utilisée pour le premier ajout. Ajoutons donc 4 chemises de taille 32 référencées cm123456 au prix unitaire de 32.45 euros et 1 blouson de cuir de taille 34 référencé vm654321 au prix de 434,95 euros. Notre code pour l'ajouter sera le suivant :

```

<?php
/* Ajoutons d'autres articles */
$select['id'] = "cm123456";
$select['qte'] = 4;
$select['taille'] = "32";
$select['prix'] = 32.45;
array_push($_SESSION['panier']['id_article'],$select['id']);
array_push($_SESSION['panier']['qte'],$select['qte']);
array_push($_SESSION['panier']['taille'],$select['taille']);
array_push($_SESSION['panier']['prix'],$select['prix']);

$select['id'] = "vm654321";
$select['qte'] = 1;
$select['taille'] = "34";
$select['prix'] = 434.95;
array_push($_SESSION['panier']['id_article'],$select['id']);
array_push($_SESSION['panier']['qte'],$select['qte']);
array_push($_SESSION['panier']['taille'],$select['taille']);
array_push($_SESSION['panier']['prix'],$select['prix']);
?>

```

Pas très pratique tout ça si on doit l'écrire à chaque fois. On va donc se faciliter un peu la vie et créer une fonction à laquelle on enverra les paramètres pour chaque article.

```

<?php
/**
 * Ajout d'un article dans le panier

```

```

*
* @param array $select variable tableau associatif contenant les valeurs de l'article
*/
function ajout($select)
{
    array_push($_SESSION['panier']['id_article'],$select['id']);
    array_push($_SESSION['panier']['qte'],$select['qte']);
    array_push($_SESSION['panier']['taille'],$select['taille']);
    array_push($_SESSION['panier']['prix'],$select['prix']);
}
?>

```

Cette fonction reçoit en paramètre une variable qui est un tableau associatif avec les valeurs requises pour l'article à ajouter. Il faut donc qu'on commence par constituer cette variable puis l'envoyer dans le panier en appelant la fonction. Reprenons donc l'envoi de notre dernier article:

```

<?php
$select['id'] = "vm654321";
$select['qte'] = 1;
$select['taille'] = "34";
$select['prix'] = 434.95;
ajout($select);
?>

```

C'est déjà plus simple.

Note importante

Attention : si l'article est déjà présent dans le panier la fonction n'en tiendra pas compte et l'ajoutera comme un nouvel article. Il vous appartient donc de faire en sorte que dans votre page catalogue, le bouton [Ajouter au panier] devienne [Retirer du panier]. Nous allons voir comment le déterminer.

Vérifier qu'un article est (ou n'est pas) dans le panier

Lorsque vous affichez la page catalogue, vous avez une liste d'article et à côté de chacun quelque chose qui ressemble à un bouton [Ajouter au panier]. Si un des article est déjà dans le panier, il serait préférable que ce bouton soit absent ou encore qu'il soit transformé en [Retirer du panier]

Il faut donc créer une fonction qui va faire le tour du panier pour comparer la référence du produit à afficher et ce qui est dans le panier.

```

<?php
/**
 * Vérifie la présence d'un article dans le panier

```

```

*
* @param String $ref_article référence de l'article à vérifier
* @return Boolean Renvoie Vrai si l'article est trouvé dans le panier, Faux sinon
*/
function verif_panier($ref_article)
{
    /* On initialise la variable de retour */
    $present = false;
    /* On vérifie les numéros de références des articles et on compare avec l'article à vérifier */
    if( count($_SESSION['panier']['id_article']) > 0 && array_search($ref_article,$_SESSION['panier']
['id_article']) !== false)
    {
        $present = true;
    }
    return $present;
}
?>

```

Il suffit maintenant lors de la construction de la page catalogue pour chaque article d'appeler la fonction `verif_panier()` pour savoir si on doit afficher [Ajouter au panier] (retour = FALSE) ou [Retirer du panier] (retour = TRUE)

ATTENTION

Je n'ai pas tenu compte ici de la taille de l'article, libre à vous d'améliorer cette fonction vous permettant d'ajouter le même article mais dans une taille différente.

Modifier la quantité d'un article

Dans notre panier, nous avons pour le moment 2 pantalons. Réflexion faite, nous allons en mettre 3. Mais comment aller précisément pointer sur la bonne quantité du bon article. Nous allons créer encore une fonction qui va faire le travail :

```

<?php
/**
 * Modifie la quantité d'un article dans le panier
 *
 * @param String $ref_article Identifiant de l'article à modifier
 * @param Int $qte Nouvelle quantité à enregistrer
 * @return Boolean Retourne VRAI si la modification a bien eu lieu, FAUX sinon.
 */
function modif_qte($ref_article, $qte)
{
    /* On compte le nombre d'articles différents dans le panier */
    $nb_articles = count($_SESSION['panier']['id_article']);
    /* On initialise la variable de retour */
    $ajoute = false;

```

```

/* On parcourt le tableau de session pour modifier l'article précis. */
for($i = 0; $i < $nb_articles; $i++)
{
    if($ref_article == $_SESSION['panier']['id_article'][$i])
    {
        $_SESSION['panier']['qte'][$i] = $qte;
        $ajoute = true;
    }
}
return $ajoute;
}
?>

```

Procédons à l'ajout :

```

<?php
$id_art = "phlevis501";
$nouvelle_qte = 3;
$modifier = modif_qte($id_art,$nouvelle_qte);
?>

```

Il ne reste qu'à tester la valeur de **\$modifier** pour s'assurer que tout s'est bien déroulé. Si vous obtenez FALSE, c'est que l'article est absent du panier.

Supprimer un article

En fin de compte, je décide d'annuler l'achat du blouson. Il y a deux méthodes. Une astuce consisterait à modifier la quantité actuelle en la remplaçant par 0. Le problème, c'est que le blouson, même avec une quantité de 0 est toujours dans le panier. Le client va peut-être culpabiliser et décider de l'acheter quand même finalement, mais ne rêvez pas trop. Il faut donc le supprimer complètement. Une autre fonction va nous permettre de faire ceci très simplement, mais la manoeuvre est plus délicate. Nous allons construire un tableau temporaire en ajoutant chaque article sauf celui à supprimer, puis nous reconstruisons le panier:

```

<?php
/**
 * Supprimer un article du panier
 *
 * @param String $ref_article numéro de référence de l'article à supprimer
 * @return Boolean Retourne TRUE si la suppression a bien été effectuée, FALSE sinon
 */
function supprim_article($ref_article)
{
    $suppression = false;
    /* création d'un tableau temporaire de stockage des articles */

```

```

$panier_tmp = array("id_article"=>array(),"qte"=>array(),"taille"=>array(),"prix"=>array());
/* Comptage des articles du panier */
$nb_articles = count($_SESSION['panier']['id_article']);
/* Transfert du panier dans le panier temporaire */
for($i = 0; $i < $nb_articles; $i++)
{
    /* On transfère tout sauf l'article à supprimer */
    if($_SESSION['panier']['id_article'][$i] != $ref_article)
    {
        array_push($panier_tmp['id_article'], $_SESSION['panier']['id_article'][$i]);
        array_push($panier_tmp['qte'], $_SESSION['panier']['qte'][$i]);
        array_push($panier_tmp['taille'], $_SESSION['panier']['taille'][$i]);
        array_push($panier_tmp['prix'], $_SESSION['panier']['prix'][$i]);
    }
}
/* Le transfert est terminé, on ré-initialise le panier */
$_SESSION['panier'] = $panier_tmp;
/* Option : on peut maintenant supprimer notre panier temporaire : */
unset($panier_tmp);
$suppression = true;
return $suppression;
}
?>

```

On peut maintenant supprimer notre blouson:

```

<?php
$reference = "vm654321";
$retrait = supprim_article($reference);
?>

```

On peut maintenant vérifier la valeur de **\$retrait**. On peut aussi afficher le panier de la même manière qu'on l'a fait au tout début.

J'ai annoncé deux méthodes : par pur esprit de contrariété, en voici une troisième. Cette méthode proposée par **Ripat** exploite les fonctions sur les manipulations de tableaux en PHP. Moins facile à appréhender, ça n'en est pas moins efficace, bien au contraire. Le procédé consiste à chercher la clé correspondant à la référence de l'article à supprimer. On supprime ensuite l'article s'il est trouvé. Par la suite on peut en option ré-indexer ou non le tableau. J'ai ajouté un paramètre pour une ré-indexation automatique qu'on peut désactiver en envoyant **FALSE**

Voici cette fonction :

```

<?php
/**
 * Supprimer un article du panier : autre méthode.
 *

```



```

* @param String      $ref_article numéro de référence de l'article à supprimer
* @param Boolean     $reindex : facultatif, par défaut, vaut true pour ré-indexer le tableau après
*                    suppression. On peut envoyer false si cette ré-indexation n'est pas nécessaire.
* @return Mixed      Retourne TRUE si la suppression a bien été effectuée,
*                    FALSE sinon, "absent" si l'article était déjà retiré du panier
*/
function supprim_article2($ref_article, $reindex = true)
{
    $suppression = false;
    $aCleSuppr = array_keys($_SESSION['panier']['id_article'], $ref_article);

    /* sortie la clé a été trouvée */
    if (!empty ($aCleSuppr))
    {
        /* on traverse le panier pour supprimer ce qui doit l'être */
        foreach ($_SESSION['panier'] as $k=>$v)
        {
            foreach($aCleSuppr as $v1)
            {
                unset($_SESSION['panier'][$k][$v1]);    // remplace la ligne foireuse
            }
            /* si la réindexation est indispensable pour la suite de l'appli, faire ici: */
            if($reindex == true)
            {
                $_SESSION['panier'][$k] = array_values($_SESSION['panier'][$k]);
            }
            $suppression = true;
        }
    }
    else
    {
        $suppression = "absent";
    }
    return $suppression;
}
?>

```

On peut procéder aux mêmes vérification qu'avec la précédente méthode.

Note importante

En terme de rapidité, la suppression directe des éléments à supprimer (la méthode de **Ripat**) est en moyenne 5 fois plus rapide, mesure faite à la suite d'un petit bench (100 boucles sur chaque appel aux fonctions) sur des paniers de 10, 100 et 1000 éléments.

Cette différence vient du fait que dans la première méthode, on traverse tout le tableau \$_SESSION['panier']['indice'] soit, disons 1000 éléments, pour ne retenir que ceux qui ne sont pas à supprimer.

Donc en conclusion, on peut dire que les deux méthodes arrivent exactement au même résultat, mais si la première est plus facile à comprendre, elle est plus lente, d'autant plus lente que la taille du panier est importante, la seconde en revanche est beaucoup plus performante mais demande un niveau d'abstraction plus important pour en comprendre le mécanisme. Je vous encourage vivement à l'étudier attentivement et à faire un tour dans la [documentation du manuel sur les fonctions de manipulation de tableaux](#).

Compter les articles ou certains éléments (le prix total?)

Nous savons maintenant ajouter des articles, en supprimer, modifier les quantités mais il serait bon de savoir où nous en sommes. Nous allons calculer le prix total de notre panier. Pour ce faire, comme je l'ai mentionné au début, les prix enregistrés sont les prix unitaires. Il nous faut donc deux éléments : d'une part le prix unitaire mais aussi les quantités. Une nouvelle fonction va nous retourner le montant total du panier.

```
<?php
/**
 * Calcule Le montant total du panier
 *
 * @return Double
 */
function montant_panier()
{
    /* On initialise Le montant */
    $montant = 0;
    /* Comptage des articles du panier */
    $nb_articles = count($_SESSION['panier']['id_article']);
    /* On va calculer Le total par article */
    for($i = 0; $i < $nb_articles; $i++)
    {
        $montant += $_SESSION['panier']['qte'][$i] * $_SESSION['panier']['prix'][$i];
    }
    /* On retourne Le résultat */
    return $montant;
}
?>
```

Un simple appel de la fonction vous retourne le total. Notez que vous pouvez utiliser la même méthode pour compter ce que vous voulez dans le panier. Adaptez selon vos propres besoins.

Vider le panier

L'opération déplaisante, notre client s'est soudain rendu compte qu'il n'a pas un sou vaillant et qu'il ne peut pas acheter quoique ce soit. Il veut vider son panier. L'opération est redoutablement simple:

```
<?php
function vider_panier()
{
    $vide = false;
    unset($_SESSION['panier']);
    if(!isset($_SESSION['panier']))
    {
        $vide = true;
    }
    return $vide;
}
?>
```

On peut vérifier maintenant :

```
<?php
$vider = vider_panier();
?>
<p>Panier <?php echo($vider?"vidé":"encore plein"); ?>.</p>
```

Navigation avec <[Précédent] ou [Suivant]> et les surprises

Un problème survient : notre internaute en navigant ajoute un article et revient à la page précédente en utilisant le bouton [Précédent] de son navigateur. Jusque là, ça peut encore aller, mais il lui prend ensuite l'idée de reprendre son chemin et il clique sur le bouton [Suivant] de son navigateur... et là, il ajoute un nouvel article, exactement le même qu'il venait de mettre dans son panier, mais comme si c'était un nouvel article différent.

Comment sortir de ce problème ? Nous allons donc introduire un nouvel élément dans la fonction qui ajoute des articles au panier. Au début de la fonction, on va faire un appel à la fonction de vérification qui va nous dire si l'article est déjà présent ou non dans le panier.

Si l'article est absent, on va le rajouter. Mais s'il est déjà là, que faire : deux possibilités : ingérer l'ajout purement et simplement, ou bien modifier la quantité. Pour ce faire, nous ferons appel à la fonction dédiée à cette opération. Notre fonction devient donc:

```
<?php
/**
 * Ajout d'un article dans le panier. Vérifie d'abord que l'article n'est pas déjà dans le panier.
 * Si l'article est absent, on l'ajoute.
 * S'il est présent, on met à jour en modifiant la quantité (y compris si c'est la même).
 *
 * @param array $select variable tableau associatif contenant les valeurs de l'article
 */
function ajout($select)
{
```

```

if(!verif_panier($select['id']))
{
    array_push($_SESSION['panier']['id_article'],$select['id']);
    array_push($_SESSION['panier']['qte'],$select['qte']);
    array_push($_SESSION['panier']['taille'],$select['taille']);
    array_push($_SESSION['panier']['prix'],$select['prix']);
}
else
{
    modif_qte($select['id'],$select['qte']);
}
}
?>

```

Mais s'il l'a fait après un ajout, rien interdit qu'il le fasse pour une suppression ou une modification, voire en vidant son panier. S'il le fait en modifiant une quantité, ça n'aura pas de conséquence particulière sinon de faire exécuter inutilement le code de mise à jour. S'il supprime un article, les navigations aller/retour avec les flèches du navigateur pourraient par contre générer une erreur, quoique bien improbable puisque ne trouvant pas la référence en parcourant le tableau, la fonction ne supprimera rien du tout, mais une fois de plus, le code sera à nouveau exécuté pour rien. Quant à la suppression par contre, elle va générer une erreur en ne trouvant pas le panier du tout puisqu'il aura été supprimé.

Corrigeons d'abord le code pour la modification de quantité et la suppression d'article. Commençons par la modification de quantité. Nous ajouterons une vérification de la quantité de l'article. Il nous faut donc une nouvelle fonction.

```

<?php
/**
 * Vérifie la quantité enregistrée d'un article dans le panier
 *
 * @param String $ref_article référence de l'article à vérifier
 * @return Mixed Renvoie le nombre d'article s'il y en a, ou Faux si cet article est absent du panier
 */
function nombre_article($ref_article)
{
    /* On initialise la variable de retour */
    $nombre = false;
    /* Comptage du panier */
    $nb_art = count($_SESSION['panier']['id_article']);
    /* On parcourt le panier à la recherche de l'article pour vérifier le cas échéant combien sont
    enregistrés */
    for($i = 0; $i < $nb_art; $i++)
    {
        if($_SESSION['panier']['id_article'][$i] == $ref_article)
            $nombre = $_SESSION['panier']['qte'][$i];
    }
    return $nombre;
}

```

```
}  
?>
```

Nous modifions ensuite la fonction `modif_panier` comme suit :

```
<?php  
/**  
 * Modifie La quantité d'un article dans Le panier  
 *  
 * @param String $ref_article  Identifiant de L'article à modifier  
 * @param Int $qte            Nouvelle quantité à enregistrer  
 * @return Mixed              Retourne VRAI si La modification a bien eu lieu,  
 *                             FAUX sinon,  
 *                             "absent" si L'article est absent du panier,  
 *                             "qte_ok" si La quantité n'est pas modifiée car déjà correctement  
enregistrée.  
 */  
function modif_qte($ref_article, $qte)  
{  
    /* On initialise La variable de retour */  
    $ajoute = false;  
    if(nombre_article($ref_article) != false && $qte != nombre_article($ref_article))  
    {  
        /* On compte Le nombre d'articles différents dans Le panier */  
        $nb_articles = count($_SESSION['panier']['id_article']);  
        /* On parcourt Le tableau de session pour modifier L'article précis. */  
        for($i = 0; $i < $nb_articles; $i++)  
        {  
            if($ref_article == $_SESSION['panier']['id_article'])  
            {  
                $_SESSION['panier']['qte'] = $qte;  
                $ajoute = true;  
            }  
        }  
    }  
    else  
    {  
        /* L'article est absent du panier, donc on ne peut pas modifier La quantité ou bien  
        * Le nombre est exactement Le même et il est inutile de Le modifier  
        * Si L'article est absent, comme on a ni La taille ni Le prix, on ne peut pas L'ajouter  
        * Si Le nombre est Le même, on ne fait pas de changement. On peut donc retourner un autre  
        * type de valeur pour indiquer une erreur qu'il faudra traiter à part Lors du retour d'appel  
        */  
        if(nombre_article($ref_article) != false)  
        {  
            $ajoute = "absent";  
        }  
    }  
}
```

```

    }
    if($qte != nombre_article($ref_article))
    {
        $ajoute = "qte_ok";
    }
}
return $ajoute;
}
?>

```

À présent, traitons la suppression d'article : de la façon dont fonctionne la méthode de suppression, ajouter quoique ce soit ne changera strictement rien. On peut certes éviter l'exécution du code de cette fonction, mais ce serait pour en exécuter un autre qui consisterait à appeler la fonction de vérification de la présence d'un article. Comme ce sera tout de même moins lourd que la création d'un panier temporaire, nous allons modifier la fonction comme ceci :

```

<?php
/**
 * Supprimer un article du panier
 *
 * @param String    $ref_article numéro de référence de l'article à supprimer
 * @return Mixed    Retourne TRUE si la suppression a bien été effectuée,
 *                  FALSE sinon, "absent" si l'article était déjà retiré du panier
 */
function supprim_article($ref_article)
{
    $suppression = false;
    /* On vérifie que l'article à supprimer est bien présent dans le panier */
    if(nombre_article($ref_article) != false)
    {
        /* création d'un tableau temporaire de stockage des articles */
        $panier_tmp = array("id_article"=>array(), "qte"=>array(), "taille"=>array(), "prix"=>array());
        /* Comptage des articles du panier */
        $nb_articles = count($_SESSION['panier']['id_article']);
        /* Transfert du panier dans le panier temporaire */
        for($i = 0; $i < $nb_articles; $i++)
        {
            /* On transfère tout sauf l'article à supprimer */
            if($_SESSION['panier']['id_article'][$i] != $ref_article)
            {
                array_push($panier_tmp['id_article'], $_SESSION['panier']['id_article'][$i]);
                array_push($panier_tmp['qte'], $_SESSION['panier']['qte'][$i]);
                array_push($panier_tmp['taille'], $_SESSION['panier']['taille'][$i]);
                array_push($panier_tmp['prix'], $_SESSION['panier']['prix'][$i]);
            }
        }
        /* Le transfert est terminé, on ré-initialise le panier */
    }
}

```

```

        $_SESSION['panier'] = $panier_tmp;
        /* Option : on peut maintenant supprimer notre panier temporaire: */
        unset($panier_tmp);
        $suppression = true;
    }
    else
    {
        /* L'article n'est pas dans le panier, on pourrait renvoyer "true" puisque de toute façon,
        * Le but était de le supprimer. Mais pour la propreté du travail, on va simplement renvoyer
        * une autre valeur indiquant que l'article est absent et ne peut donc pas être supprimé.
        * il conviendra lors de la récupération de traiter les informations à afficher selon ce
        résultat */
        $suppression = "absent";
    }
    return $suppression;
}
?>

```

Notez que la seconde version proposée de suppression d'article contient déjà une vérification pour la raison simple qu'on a besoin de trouver la clé correspondant à l'identifiant de l'article à supprimer, donc elle ne nécessitera pas de modification sur ce point.

Enfin, la suppression du panier. La modification ne sera pas majeure, il suffit simplement de vérifier l'existence du panier. On va donc modifier la fonction comme suit :

```

<?php
/**
 * Vider le panier
 *
 * @return Mixed    Retourne VRAI si l'exécution s'est correctement déroulée, Faux sinon et
 "inexistant" si
 *                  Le panier avait déjà été détruit ou n'avait jamais été créé.
 */
function vider_panier()
{
    $vide = false;
    if(isset($_SESSION['panier']))
    {
        unset($_SESSION['panier']);
        if(!isset($_SESSION['panier']))
        {
            $vide = true;
        }
    }
    else
    {
        /* Le panier était déjà détruit, on renvoie une autre valeur exploitable au retour */
    }
}

```

```

        $vide = "inexistant";
    }
    return $vide;
}
?>

```

Voilà, toutes nos fonctions sont maintenant opérationnelles sur un plan basique. Attachons-nous maintenant à un autre aspect de la question.

Sécuriser notre panier

Un problème de sécurité pourrait se présenter lors du passage à la caisse. Ce problème souligné par Cerber que je remercie en passant a été fort en vogue dans les début du commerce électronique. La méthode consistait à ouvrir deux pages d'une même navigateur sur le catalogue et à modifier le contenu du panier sur une page pendant qu'on était la la phase de paiement sur l'autre. La méthode consiste donc à verrouiller le panier lors de la confirmation du paiement.

Pour ce faire, nous allons rajouter un index à notre variable de session dans la petite fonction suivante :

```

<?php
/**
 * Fonction de verrouillage du panier pendant la phase de paiement.
 *
 */
function preparerPaiement()
{
    $_SESSION['panier']['verrouille'] = true;
    header("Location: URL_DU_SITE_DE_BANQUE");
}
?>

```

Lorsque le paiement aura été accepté par la banque, il vous restera une dernière mesure de précaution à prendre avec cette autre fonction :

```

<?php
/**
 * Fonction qui va enregistrer les informations de la commande dans
 * la base de données et détruire le panier.
 *
 */
function paiementAccepte()
{
    /* ~~~~~ */
    /* Stockage du panier dans la BDD */
}

```



```

/* ajoutez ici votre code d'insertion */
/* ~~~~~ */
unset($_SESSION['panier']);
}
?>

```

Il nous reste à sécuriser les autres fonctions interdisant les modifications en phase de paiement. Voici comment seront modifiées nos fonctions, exemple avec l'ajout d'article :

```

<?php
function ajout($select)
{
    $ajout = false;
    if(!isset($_SESSION['panier']['verrouille']) || $_SESSION['panier']['verrouille'] == false)
    {
        /* Reste du code */
    }
    return $ajout;
}
?>

```

Optimiser tout ça

Un option qui reste pratique serait de créer un fichier unique, **panier.php** par exemple, regroupant toutes vos fonctions dans un même fichier externe de façon à pouvoir l'utiliser dans les différentes parties de votre catalogue sans devoir chaque fois tout ré-écrire.

Voici ce que ça donne avec une remise en ordre, le fichier étant en quelque sorte coupé en deux, d'abord les fonctions de base puis les fonctions annexes :

```

<?php
/* ~~~~~ */
/* Fonctions de base de gestion du panier */
/* ~~~~~ */

/**
 * Ajoute un article dans le panier après vérification que nous ne sommes pas en phase de paiement
 *
 * @param array $select variable tableau associatif contenant les valeurs de l'article
 * @return Mixed Retourne VRAI si l'ajout est effectué, FAUX sinon voire une autre valeur si l'ajout
 * est renvoyé vers la modification de quantité.
 * @see {@link modif_qte()}
 */
function ajout($select)

```

```

{
    $ajout = false;
    if(!isset($_SESSION['panier']['verrouille']) || $_SESSION['panier']['verrouille'] == false)
    {
        if(!verif_panier($select['id']))
        {
            array_push($_SESSION['panier']['id_article'],$select['id']);
            array_push($_SESSION['panier']['qte'],$select['qte']);
            array_push($_SESSION['panier']['taille'],$select['taille']);
            array_push($_SESSION['panier']['prix'],$select['prix']);
            $ajout = true;
        }
        else
        {
            $ajout = modif_qte($select['id'],$select['qte']);
        }
    }
    return $ajout;
}

/**
 * Modifie la quantité d'un article dans le panier après vérification que nous ne sommes pas en phase de paiement
 *
 * @param String $ref_article    Identifiant de l'article à modifier
 * @param Int $qte               Nouvelle quantité à enregistrer
 * @return Mixed                Retourne VRAI si la modification a bien eu lieu,
 *                              FAUX sinon,
 *                              "absent" si l'article est absent du panier,
 *                              "qte_ok" si la quantité n'est pas modifiée car déjà correctement enregistrée.
 */
function modif_qte($ref_article, $qte)
{
    /* On initialise la variable de retour */
    $modifie = false;
    if(!isset($_SESSION['panier']['verrouille']) || $_SESSION['panier']['verrouille'] == false)
    {
        if(nombre_article($ref_article) != false && $qte != nombre_article($ref_article))
        {
            /* On compte le nombre d'articles différents dans le panier */
            $nb_articles = count($_SESSION['panier']['id_article']);
            /* On parcourt le tableau de session pour modifier l'article précis. */
            for($i = 0; $i < $nb_articles; $i++)
            {
                if($ref_article == $_SESSION['panier']['id_article'][$i])
                {
                    $_SESSION['panier']['qte'][$i] = $qte;
                    $modifie = true;
                }
            }
        }
    }
}

```

```

    }
}
else
{
    /* L'article est absent du panier, donc on ne peut pas modifier la quantité ou bien
    * le nombre est exactement le même et il est inutile de le modifier
    * Si l'article est absent, comme on a ni la taille ni le prix, on ne peut pas l'ajouter
    * Si le nombre est le même, on ne fait pas de changement. On peut donc retourner un autre
    * type de valeur pour indiquer une erreur qu'il faudra traiter à part lors du retour d'appel
    */
    if(nombre_article($ref_article) != false)
    {
        $modifie = "absent";
    }
    if($qte != nombre_article($ref_article))
    {
        $modifie = "qte_ok";
    }
}
}
return $modifie;
}

/**
 * Supprimer un article du panier après vérification que nous ne sommes pas en phase de paiement
 *
 * @param String    $ref_article numéro de référence de l'article à supprimer
 * @return Mixed    Retourne TRUE si la suppression a bien été effectuée,
 *                  FALSE sinon, "absent" si l'article était déjà retiré du panier
 */
function supprim_article($ref_article)
{
    $suppression = false;
    if(!isset($_SESSION['panier']['verrouille']) || $_SESSION['panier']['verrouille'] == false)
    {
        /* On vérifie que l'article à supprimer est bien présent dans le panier */
        if(nombre_article($ref_article) != false)
        {
            /* création d'un tableau temporaire de stockage des articles */
            $panier_tmp = array("id_article"=>array(),"qte"=>array(),"taille"=>array(),"prix"=>array());
            /* Comptage des articles du panier */
            $nb_articles = count($_SESSION['panier']['id_article']);
            /* Transfert du panier dans le panier temporaire */
            for($i = 0; $i < $nb_articles; $i++)
            {
                /* On transfère tout sauf l'article à supprimer */

```

```

        if($_SESSION['panier']['id_article'][$i] != $ref_article)
        {
            array_push($panier_tmp['id_article'],$SESSION['panier']['id_article'][$i]);
            array_push($panier_tmp['qte'],$SESSION['panier']['qte'][$i]);
            array_push($panier_tmp['taille'],$SESSION['panier']['taille'][$i]);
            array_push($panier_tmp['prix'],$SESSION['panier']['prix'][$i]);
        }
    }
    /* Le transfert est terminé, on ré-initialise le panier */
    $_SESSION['panier'] = $panier_tmp;
    /* Option : on peut maintenant supprimer notre panier temporaire: */
    unset($panier_tmp);
    $suppression = true;
}
else
{
    $suppression == "absent";
}
}
return $suppression;
}

/**
 * Supprimer un article du panier : autre méthode.
 *
 * @param String    $ref_article numéro de référence de l'article à supprimer
 * @param Boolean   $reindex : facultatif, par défaut, vaut true pour ré-indexer le tableau après
 *                          suppression. On peut envoyer false si cette ré-indexation n'est pas nécessaire.
 * @return Mixed    Retourne TRUE si la suppression a bien été effectuée,
 *                  FALSE sinon, "absent" si l'article était déjà retiré du panier
 */
function supprim_article2($ref_article, $reindex = true)
{
    $suppression = false;
    if(!isset($_SESSION['panier']['verrouille']) || $_SESSION['panier']['verrouille'] == false)
    {
        $aCleSuppr = array_keys($_SESSION['panier']['id_article'], $ref_article);

        /* sortie la clé a été trouvée */
        if (!empty ($aCleSuppr))
        {
            /* on traverse le panier pour supprimer ce qui doit l'être */
            foreach ($_SESSION['panier'] as $k=>$v)
            {
                foreach($aCleSuppr as $v1)
                {
                    unset($_SESSION['panier'][$k][$v1]);    // remplace la ligne foireuse
                }
            }
        }
    }
}

```

```

    }
    /* Réindexation des clés du panier si l'option $reindex a été laissée à true */
    if($reindex == true)
    {
        $_SESSION['panier'][$k] = array_values($_SESSION['panier'][$k]);
    }
    $suppression = true;
}
}
else
{
    $suppression = "absent";
}
}
return $suppression;
}

/**
 * Fonction qui supprime tout le contenu du panier en détruisant la variable après
 * vérification qu'on ne soit pas en phase de paiement.
 *
 * @return Mixed    Retourne VRAI si l'exécution s'est correctement déroulée, Faux sinon et "inexistant" si
 *                  le panier avait déjà été détruit ou n'avait jamais été créé.
 */
function vider_panier()
{
    $vide = false;
    if(!isset($_SESSION['panier']['verrouille']) || $_SESSION['panier']['verrouille'] == false)
    {
        if(isset($_SESSION['panier']))
        {
            unset($_SESSION['panier']);
            if(!isset($_SESSION['panier']))
            {
                $vide = true;
            }
        }
        else
        {
            /* Le panier était déjà détruit, on renvoie une autre valeur exploitable au retour */
            $vide = "inexistant";
        }
    }
    return $vide;
}

/* ~~~~~ */

```

```

/*          Fonctions annexes de gestion du panier          */
/* ~~~~~ */

/**
 * Vérifie la quantité enregistrée d'un article dans le panier
 *
 * @param String $ref_article référence de l'article à vérifier
 * @return Mixed Renvoie le nombre d'article s'il y en a, ou Faux si cet article est absent du panier
 */
function nombre_article($ref_article)
{
    /* On initialise la variable de retour */
    $nombre = false;
    /* Comptage du panier */
    $nb_art = count($_SESSION['panier']['id_article']);
    /* On parcourt le panier à la recherche de l'article pour vérifier le cas échéant combien sont enregistrés */
    for($i = 0; $i < $nb_art; $i++)
    {
        if($_SESSION['panier']['id_article'][$i] == $ref_article)
            $nombre = $_SESSION['panier']['qte'][$i];
    }
    return $nombre;
}

/**
 * Calcule le montant total du panier
 *
 * @return Double
 */
function montant_panier()
{
    /* On initialise le montant */
    $montant = 0;
    /* Comptage des articles du panier */
    $nb_articles = count($_SESSION['panier']['id_article']);
    /* On va calculer le total par article */
    for($i = 0; $i < $nb_articles; $i++)
    {
        $montant += $_SESSION['panier']['qte'][$i] * $_SESSION['panier']['prix'][$i];
    }
    /* On retourne le résultat */
    return $montant;
}

/**
 * Vérifie la présence d'un article dans le panier
 *

```

```

* @param String $ref_article référence de l'article à vérifier
* @return Boolean Renvoie Vrai si l'article est trouvé dans le panier, Faux sinon
*/
function verif_panier($ref_article)
{
    /* On initialise la variable de retour */
    $present = false;
    /* On vérifie les numéros de références des articles et on compare avec l'article à vérifier */
    if( count($_SESSION['panier']['id_article']) > 0 && array_search($ref_article,$_SESSION['panier']
['id_article']) !== false)
    {
        $present = true;
    }
    return $present;
}

/**
* Fonction de verrouillage du panier pendant la phase de paiement.
*
*/
function preparerPaiement()
{
    $_SESSION['panier']['verrouille'] = true;
    header("Location: URL_DU_SITE_DE_BANQUE");
}

/**
* Fonction qui va enregistrer les informations de la commande dans
* la base de données et détruire le panier.
*
*/
function paiementAccepte()
{
    /* ~~~~~ */
    /* Stockage du panier dans la BDD */
    /* ajoutez ici votre code d'insertion */
    /* ~~~~~ */
    unset($_SESSION['panier']);
}
?>

```

ATTENTION

Comme il s'agit d'un fichier externe, on ne met plus `session_start()`; au début. N'oubliez pas de le mettre au début de vos pages.

Voilà, ceci n'est bien entendu qu'un guide. Tous les détails ne sont pas là, on pourrait écrire presque un livre complet sur la construction de paniers virtuels et ce n'est pas l'objet de ce tutorial.

En outre, pour aller plus loin, il faudrait transformer ça en programmation objet et créer une classe encore plus simple à manipuler : ça déborde du but poursuivi dans cette page.

Bon code.

Remerciements

Un merci tout spécial à **Cerber** pour son intervention des plus utiles sur la sécurité.

Un merci également à **Ripat** pour ses interventions documentées sur la partie suppression d'articles et sa proposition tout à fait pertinente.

Merci également à tous les ViPhP de PHPFrance qui ont participé en m'apportant leur collaboration et leurs remarques pour corriger et optimiser autant que faire se pouvait ce tutorial.