DOSSIER PHP SQL

Contenu

[I. Langage PHP 3](#_Toc33625113)

[A. Introduction 4](#_Toc33625114)

[B. Les variables 5](#_Toc33625115)

[C. Les variables superglobales 6](#_Toc33625116)

[D. Condition IF, case et ternaires 7](#_Toc33625117)

[1. IF,Elseif , Else 7](#_Toc33625118)

[2. SWITCH/CASE 8](#_Toc33625119)

[3. Conditions Ternaires 8](#_Toc33625120)

[E. Boucle While, until, for 9](#_Toc33625121)

[1. Boucle While 9](#_Toc33625122)

[2. Boucle do-While 9](#_Toc33625123)

[3. Boucle For 9](#_Toc33625124)

[F. Les fonctions 10](#_Toc33625125)

[1. Créer ses propres Fonctions 10](#_Toc33625126)

[2. Les fonctions existante de PHP 10](#_Toc33625127)

[G. Inclure des portions de page 11](#_Toc33625128)

[H. Transmissions des données 11](#_Toc33625129)

[1. Via URLs 11](#_Toc33625130)

[2. Via Formulaire 12](#_Toc33625131)

[I. Les sessions et cookies 16](#_Toc33625132)

[1. Les sessions 16](#_Toc33625133)

[2. Les cookies 16](#_Toc33625134)

[*J.* *Lire et Ecrire dans un Fichier* 17](#_Toc33625135)

[K. Créer une image en PHP 18](#_Toc33625136)

[1. Créer une image 18](#_Toc33625137)

[2. Afficher une image 19](#_Toc33625138)

[3. Enregistrer une image 19](#_Toc33625139)

[4. Ajouter du texte et des couleurs 20](#_Toc33625140)

[5. Ajouter des formes 21](#_Toc33625141)

[II. Langage SQL 22](#_Toc33625142)

[A. Introduction 22](#_Toc33625143)

[B. Les requêtes SQL 23](#_Toc33625144)

[C. Connexions à la Base de donnée 23](#_Toc33625145)

[D. Lire et Ecrire les données 24](#_Toc33625146)

[E. Les fonctions SQL 25](#_Toc33625147)

[1. Les fonctions scalaires 25](#_Toc33625148)

[2. Les fonctions d’agrégat 35](#_Toc33625149)

[F. Les dates en SQL 38](#_Toc33625150)

[*1.* *Les différents types de dates* 38](#_Toc33625151)

[*2.* *Les outils des comparaisons de date* 38](#_Toc33625152)

[3. Les fonctions dates 39](#_Toc33625153)

[G. Les jointures de tables. 40](#_Toc33625154)

[1. Explications 40](#_Toc33625155)

[2. Les jointures internes 41](#_Toc33625156)

[3. Les jointures externes 42](#_Toc33625157)

[H. Les expressions régulières 44](#_Toc33625158)

[1. Les fonctions 44](#_Toc33625159)

[2. Mémento des expressions régulières 45](#_Toc33625160)

# Langage PHP

## Introduction

**PHP** : c'est un langage que seuls les serveurs comprennent et qui permet de rendre votre site dynamique. C'est PHP qui « génère » la page web comme on l'a vu sur un des schémas précédents.

Parmi les concurrents de PHP, on peut citer les suivants :

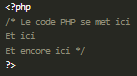
* ASP .NET : conçu par Microsoft, il exploite le framework (c'est-à-dire un ensemble de bibliothèques qui fournissent des services pour les développeurs) .NET bien connu des développeurs C#. Ce langage peut être intéressant si vous avez l'habitude de développer en C# .NET et que vous ne voulez pas être dépaysés.
* Ruby on Rails : très actif, ce framework s'utilise avec le langage Ruby et permet de réaliser des sites dynamiques rapidement en suivant certaines conventions.
* Django : il est similaire à Ruby on Rails, mais il s'utilise en langage Python.Java et les JSP (Java Server Pages) : plus couramment appelé « JEE » ou « Java EE », il est particulièrement utilisé dans le monde professionnel. Il demande une certaine rigueur. La mise en place d'un projet JEE est traditionnellement un peu plus longue et plus lourde mais le système est apprécié des professionnels et des institutions. C'est ce qui est utilisé sur le site des impôts français, par exemple

**MySQL** : c'est ce qu'on appelle un SGBD (Système de Gestion de Base de Données). Pour faire simple, son rôle est d'enregistrer des données de manière organisée afin de vous aider à les retrouver facilement plus tard. C'est grâce à MySQL que vous pourrez enregistrer la liste des membres de votre site, les messages postés sur le forum, etc. Le langage qui permet de communiquer avec la base de données s'appelle le SQL. Voici un code en langage SQL.

Parmi les concurrents de MySQL, je vous conseille de connaître (au moins de nom) les suivants :

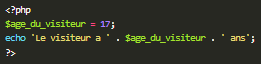
* Oracle : c'est le SGBD le plus célèbre, le plus complet et le plus puissant. Il est malheureusement payant (et cher), ce qui le réserve plutôt aux entreprises qui l'utilisent déjà massivement. Il existe cependant des versions gratuites d'Oracle, notamment pour ceux qui veulent apprendre à s'en servir.
* MariaDB : variante libre de MySQL, qui a été créée depuis que ce dernier a été racheté par... Oracle. Oui, je sais, on peut penser que ce monde est fou ! MySQL appartient à Oracle, mais reste bien une base de données à part. MariaDB est une copie (fork) de MySQL qui a la volonté de rester libre et indépendante.
* Microsoft SQL Server : édité par Microsoft, on l'utilise souvent en combinaison avec ASP .NET, bien qu'on puisse l'utiliser avec n'importe quel autre langage. Il est payant, mais il existe des versions gratuites limitées.
* PostgreSQL : il s'agit d'un SGBD libre et gratuit comme MySQL, qui propose des fonctionnalités plus avancées. Parfois comparé à Oracle, il lui reste cependant du chemin à parcourir. Il dispose d'une communauté un peu moins importante que MySQL et Oracle. OpenClassrooms utilise PostgreSQL.

Voici la forme d’une balise PHP :



## Les variables

Pour déclarer une variable il suffit de mettre le symbole « $ » suivis du nom de la variable, exemple :

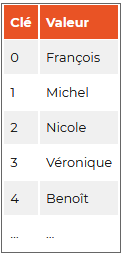


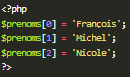
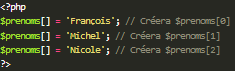
Il n y a pas besoin de préciser le type ( Int, float,string ect )

Les tableaux

Déclarer un tableaux :

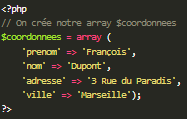
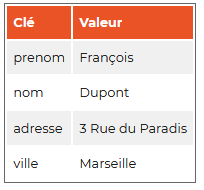
Pour créer un tableau numéroté en PHP, on utilise généralement la fonction array

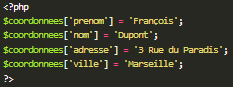


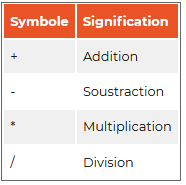


Déclarer un tableau associatif :

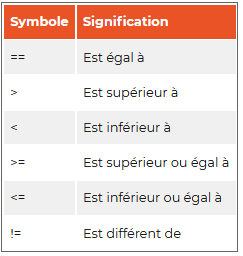
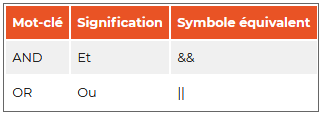




Les calculs

Les symboles de comparaison et conditions multiples

## Les variables superglobales

Il existe d'autres types de variables superglobales que nous allons découvrir dans ce chapitre. Parmi elles, certaines permettent de stocker des informations pendant la durée d'une visite, c'est le principe des sessions, mais aussi de stocker des informations sur l'ordinateur de vos visiteurs pendant plusieurs mois, c'est le principe des cookies.

Les variables superglobales sont des variables un peu particulières pour trois raisons :

* elles sont écrites en majuscules et commencent toutes, à une exception près, par un underscore (\_).$\_GETet$\_POSTen sont des exemples que vous connaissez ;
* les superglobales sont desarraycar elles contiennent généralement de nombreuses informations ;
* enfin, ces variables sont automatiquement créées par PHP à chaque fois qu'une page est chargée. Elles existent donc sur toutes les pages et sont accessibles partout : au milieu de votre code, au début, dans les fonctions, etc.

Pour afficher le contenu d'une superglobale et voir ce qu'elle contient, le plus simple est d'utiliser la fonction print\_r, puisqu'il s'agit d'un array

**$\_SERVER :** ce sont des valeurs renvoyées par le serveur. Elles sont nombreuses et quelques-unes d'entre elles peuvent nous être d'une grande utilité. Je vous propose de retenir au moins$\_SERVER['REMOTE\_ADDR']. Elle nous donne l'adresse IP du client qui a demandé à voir la page, ce qui peut être utile pour l'identifier.

**$\_ENV :** ce sont des variables d'environnement toujours données par le serveur. C'est le plus souvent sous des serveurs Linux que l'on retrouve des informations dans cette superglobale. Généralement, on ne trouvera rien de bien utile là-dedans pour notre site web.

**$\_SESSION :** on y retrouve les variables de session. Ce sont des variables qui restent stockées sur le serveur le temps de la présence d'un visiteur. Nous allons apprendre à nous en servir dans ce chapitre.

**$\_COOKIE :** contient les valeurs des cookies enregistrés sur l'ordinateur du visiteur. Cela nous permet de stocker des informations sur l'ordinateur du visiteur pendant plusieurs mois, pour se souvenir de son nom par exemple.

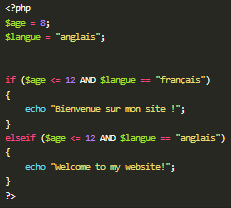
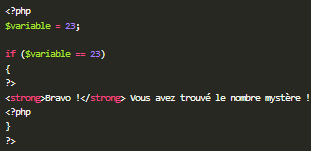
**$\_GET :** vous la connaissez, elle contient les données envoyées en paramètres dans l'URL.

**$\_POST :** de même, c'est une variable que vous connaissez et qui contient les informations qui viennent d'être envoyées par un formulaire.

**$\_FILES :** elle contient la liste des fichiers qui ont été envoyés via le formulaire précédent.

## Condition IF, switch case et ternaires

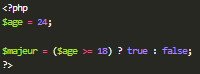
### IF,Elseif , Else

### SWITCH/CASE

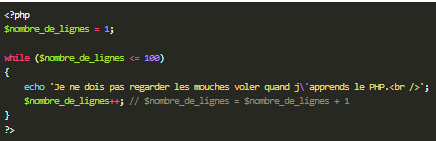


### Conditions Ternaires

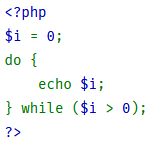
La condition testée est $age >= 18. Si c'est vrai, alors la valeur indiquée après le point d'interrogation (ici true) sera affectée à la variable $majeur. Sinon, c'est la valeur qui suit le symbole « deux-points » qui sera affectée à $majeur.

## Boucle While, until, for

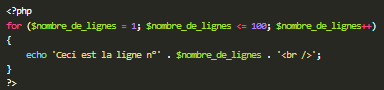
### Boucle While

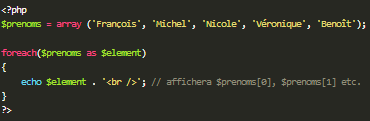


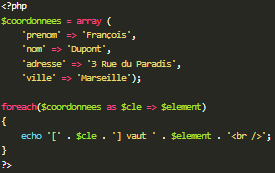
### Boucle do-While

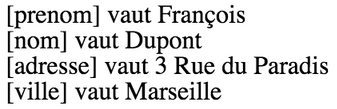
la différence avec une boucle While est que les instructions dans la boucle seront exécuter au moins une fois.

### Boucle For



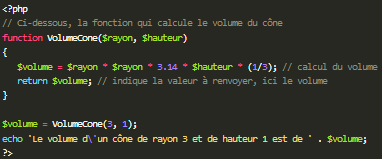
Permet de parcourir un tableaux la valeur elements prend la valeur de la variable 0 puis 1 puis 2 Ect à chaque tour de boucle





## Les fonctions

### Créer ses propres Fonctions

### Les fonctions existante de PHP

Phpinfo() ; Pour connaitre l'ensemble des informations relatives au PHP utilisé par le serveur web.

Echo « Texte » ; permet d’afficher du texte.

Array\_key\_exists(‘cle’,tableaux) : pour vérifier si une clé existe dans l'array ;

in\_array(‘valeur’,tableaux) : pour vérifier si une valeur existe dans l'array ;

array\_search(‘valeur’,tableaux) : pour récupérer la clé d'une valeur dans l'array ;

str\_replace(‘chaine rechercher’,’nouvel chaine’,’chaine de recherche’): rechercher et remplacer ;

str\_shuffle(chaine): mélanger les lettres ;

strtolower(‘chaine’): écrire en minuscules ;

strtoupper(‘chaine’): écrire en majuscule ;

date(‘time’): récupère la date (H=Heure,i=minute,d=Jour,m=Mois,Y=Année) ;

isset() : Cette fonction teste si une variable existe ;

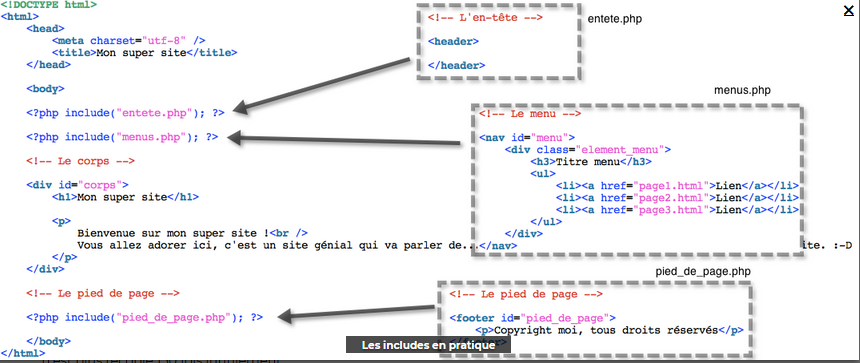
print\_r(tableaux) : affiche le contenu d’un tableaux ;

session\_start()  : démarre le système de sessions. Si le visiteur vient d'arriver sur le site, alors un numéro de session est généré pour lui. Vous devez appeler cette fonction au tout début de chacune des pages où vous avez besoin des variables de session ;

session\_destroy()  : ferme la session du visiteur. Cette fonction est automatiquement appelée lorsque le visiteur ne charge plus de page de votre site pendant plusieurs minutes (c'est le timeout), mais vous pouvez aussi créer une page « Déconnexion » si le visiteur souhaite se déconnecter manuellement

setcookie(‘nom\_du\_cookies’,’valeur\_du\_cookie’,timestamp’,Null,Null, false, true) : crée un cookie ;

## Inclure des portions de page

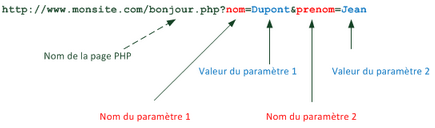


## Transmissions des données

### Via URLs

Envoie des données

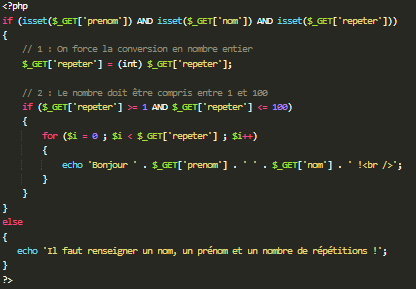
<http://www.monsite.com/bonjour.php?nom=Dupont&prenom=Jean>



Récupération des données envoyées



Tester et contrôler les paramètres reçus (TRES IMPORTANT)



En étant plus malins et surtout plus prévoyants. Voici ce qu'il faut anticiper.

Le cas où le nombre qu'on vous envoie n'est pas une valeur raisonnable.

Par exemple on peut dire que si on dépasse 100 fois, c'est trop, et il faut refuser d'exécuter la page.

De même, que se passe-t-il si je demande de répéter « $-4$ fois » ? Ça ne devrait pas être autorisé. Il faut que le nombre soit au moins égal à 1.

Le cas où la valeur n'est pas logique, où elle n'est pas du tout ce qu'on attendait. Rien n'empêche le visiteur de remplacer la valeur du paramètre repeter par du texte !

Que se passe-t-il si on essaie d'accéder à bonjour.php?nom=Dupont&prenom=Jean&repeter=grenouille ?

Cela ne devrait pas être autorisé. Le mot « grenouille » n'a pas de sens, on veut un nombre entier positif.

Que se passe-t-il si on essaie d'accéder à bonjour.php?nom=Dupont&prenom=Jean&repeter=true ?

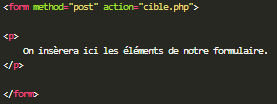
Cela ne devrait pas être autorisé non plus, on attendait un nombre entier positif, pas un booléen.

Pour résoudre ces problèmes, on doit :

* vérifier que repeter contient bien un nombre ;
* vérifier ensuite que ce nombre est compris dans un intervalle raisonnable (entre 1 et 100, par exemple).

### Via Formulaire

Envoie des données

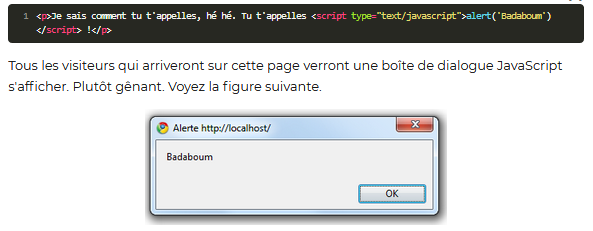
Au lieu de recevoir un array $\_GET, vous allez recevoir un array $\_POST contenant les données du formulaire !

Récupération des données envoyées

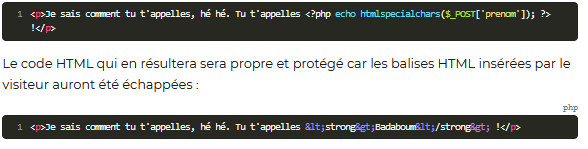


La faille XSS : attention au code HTML que vous recevez !

Outre le fait qu'il peut insérer n'importe quel code HTML (et rendre votre page invalide), ce qui n'est pas le plus grave, il peut aussi ouvrir des balises de type <script> pour faire exécuter du code JavaScript au visiteur qui visualisera la page !



Pour échapper le code HTML, il suffit d'utiliser la fonction htmlspecialchars qui va transformer les chevrons des balises HTML<>en&lt;et&gt;respectivement. Cela provoquera l'affichage de la balise plutôt que son exécution.



Si vous préférez retirer les balises HTML que le visiteur a tenté d'envoyer plutôt que de les afficher, utilisez la fonction strip\_tags.

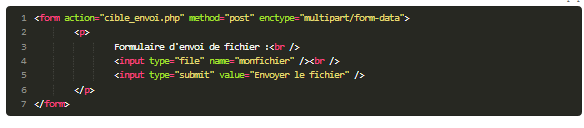
L’envoie de fichier

Dès l'instant où votre formulaire propose aux visiteurs d'envoyer un fichier, il faut ajouter l'attributenctype="multipart/form-data"à la balise<form>.



Grâce à enctype, le navigateur du visiteur sait qu'il s'apprête à envoyer des fichiers.

Maintenant que c'est fait, nous pouvons ajouter à l'intérieur du formulaire une balise permettant d'envoyer un fichier. C'est une balise très simple de type <input type="file" />. Il faut penser comme toujours à donner un nom à ce champ de formulaire (grâce à l'attribut name) pour que PHP puisse reconnaître le champ par la suite.



En fait, au moment où la page PHP s'exécute, le fichier a été envoyé sur le serveur mais il est stocké dans un **dossier temporaire**.  
C'est à vous de décider si vous acceptez définitivement le fichier ou non. Vous pouvez par exemple vérifier si le fichier a la bonne extension (si vous demandiez une image et qu'on vous envoie un « .txt », vous devrez refuser le fichier).

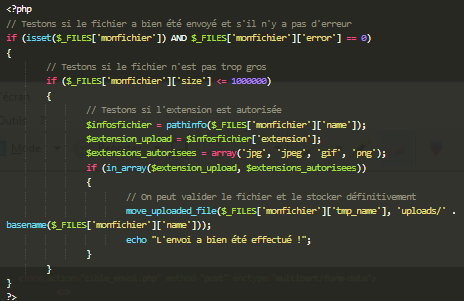
Si le fichier est bon, vous l'accepterez grâce à la fonction move\_uploaded\_file, et ce, d'une manière définitive.

| **Variable** | **Signification** |
| --- | --- |
| $\_FILES['monfichier']['name'] | Contient le nom du fichier envoyé par le visiteur. |
| $\_FILES['monfichier']['type'] | Indique le type du fichier envoyé. Si c'est une image gif par exemple, le type seraimage/gif. |
| $\_FILES['monfichier']['size'] | Indique la taille du fichier envoyé. **Attention**: cette taille est en octets. Il faut environ 1 000 octets pour faire 1 Ko, et 1 000 000 d'octets pour faire 1 Mo. **Attention**: la taille de l'envoi est limitée par PHP. Par défaut, impossible d'uploader des fichiers de plus de 8 Mo. |
| $\_FILES['monfichier']['tmp\_name'] | Juste après l'envoi, le fichier est placé dans un répertoire temporaire sur le serveur en attendant que votre script PHP décide si oui ou non il accepte de le stocker pour de bon. Cette variable contient l'emplacement temporaire du fichier (c'est PHP qui gère ça). |
| $\_FILES['monfichier']['error'] | Contient un code d'erreur permettant de savoir si l'envoi s'est bien effectué ou s'il y a eu un problème et si oui, lequel. La variable vaut 0 s'il n'y a pas eu d'erreur. |

Je vous propose de faire les vérifications suivantes pour décider si l'on accepte le fichier ou non.

1. Vérifier tout d'abord si le visiteur a bien envoyé un fichier (en testant la variable$\_FILES['monfichier']avec isset()) et s'il n'y a pas eu d'erreur d'envoi (grâce à $\_FILES['monfichier']['error']).
2. Vérifier si la taille du fichier ne dépasse pas 1 Mo par exemple (environ 1 000 000 d'octets) grâce à$\_FILES['monfichier']['size'].
3. Vérifier si l'extension du fichier est autorisée (il faut interdire à tout prix que les gens puissent envoyer des fichiers PHP, sinon ils pourraient exécuter des scripts sur votre serveur). Dans notre cas, nous autoriserons seulement les images (fichiers .png, .jpg, .jpeg et .gif).  
   Nous analyserons pour cela la variable$\_FILES['monfichier']['name'].

Nous allons donc faire une série de tests dans notre pagecible\_envoi.php.

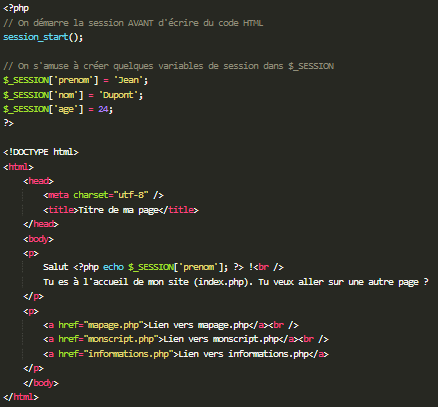


Soyez toujours très vigilants sur la sécurité, vous devez éviter que quelqu'un puisse envoyer des fichiers PHP sur votre serveur.

## Les sessions et cookies

### Les sessions

il faut appeler session\_start() sur chacune de vos pages AVANT d'écrire le moindre code HTML (avant même la balise <!DOCTYPE> ;). Si vous oubliez de lancer session\_start() , vous ne pourrez pas accéder aux variables superglobales $\_SESSION .



### Les cookies

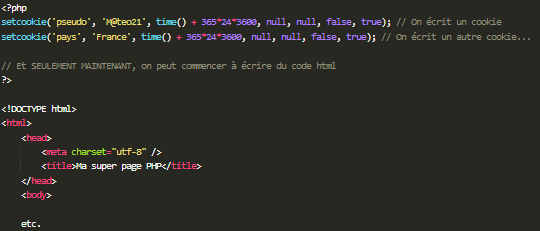
Pour écrire un cookie, on utilise la fonction PHP setcookie (qui signifie « Placer un cookie » en anglais).

On lui donne en général trois paramètres, dans l'ordre suivant :

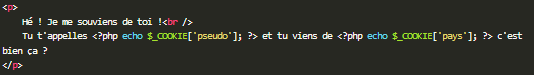
* le nom du cookie (ex. : pseudo ) ;
* la valeur du cookie (ex. : M@teo21 ) ;
* la date d'expiration du cookie, sous forme de timestamp (ex. : 1090521508 ).

Le paramètre correspondant à la date d'expiration du cookie mérite quelques explications. Il s'agit d'un timestamp, c'est-à-dire du nombre de secondes écoulées depuis le 1er janvier 1970. Le timestamp est une valeur qui augmente de 1 toutes les secondes. Pour obtenir le timestamp actuel, on fait appel à la fonction time() . Pour définir une date d'expiration du cookie, il faut ajouter au « moment actuel » le nombre de secondes au bout duquel il doit expirer.

Ecrire ou modifier un cookie



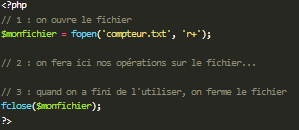
Afficher un cookie



## Lire et Ecrire dans un Fichier

Tout d’abord pour pouvoir écrire dans un fichier il faudra changer le CHMOD c’est-à-dire les permissions publiques d’criture sur le fichier. Il faut rentrer 777.

Ouvrir et fermer un fichier

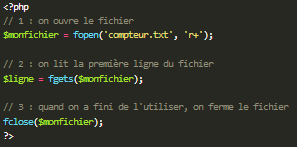


| **Mode** | **Explication** |
| --- | --- |
| r | Ouvre le fichier en lecture seule. Cela signifie que vous pourrez seulement lire le fichier. |
| r+ | Ouvre le fichier en lecture et écriture. Vous pourrez non seulement lire le fichier, mais aussi y écrire (on l'utilisera assez souvent en pratique). |
| a | Ouvre le fichier en écriture seule. Mais il y a un avantage : si le fichier n'existe pas, il est automatiquement créé. |
| a+ | Ouvre le fichier en lecture et écriture. Si le fichier n'existe pas, il est créé automatiquement. Attention : le répertoire doit avoir un CHMOD à 777 dans ce cas ! À noter que si le fichier existe déjà, le texte sera ajouté à la fin. |

Lire dans un fichier

Il existe deux possibilités :

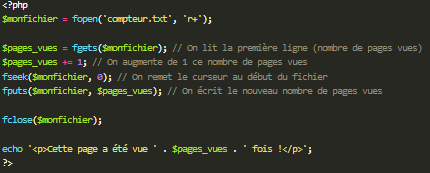
* Lire caractère par caractère avec la fonction fgetc
* Lire ligne par ligne avec fgets



Ecrire dans un fichier

Une seul possibilité utilisé la fonction fputs

Pour réinitialiser la position du curseur après par exemple un fgets, il faut utiliser la fonction fseek($nomdufichier,0) ;

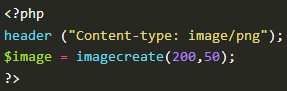


## Créer une image en PHP

### Créer une image

Pour cela il faut activer la bibliothèque GD en PHP

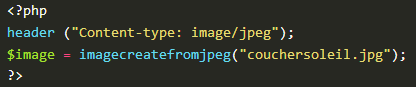
A partir d’une image vide.



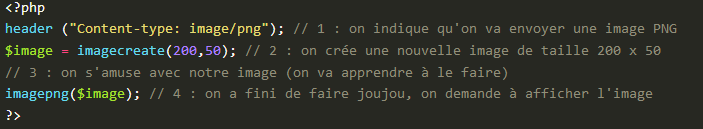
A partir d’une image existante

Grâce à la fonction :

* JPEG : il faut utiliser la fonction imagecreatefromjpeg
* PNG : il faut utiliser la fonction imagecreatefrompng



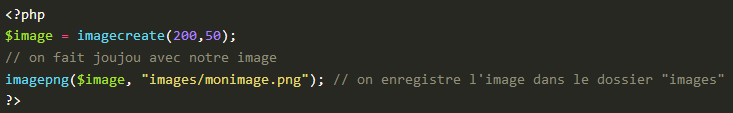
### Afficher une image



Pour utiliser l’image comme je le souhaite n’ importe où dans mon site il faudra faire comme une image classique.



### Enregistrer une image



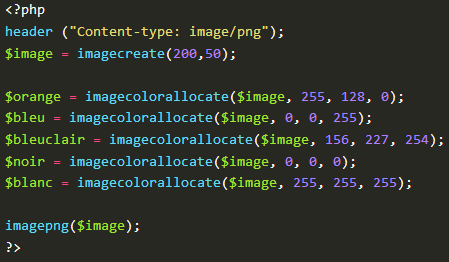
Ici on utilise plus header car nous n’enverrons plus d’image étant donnée qu’elle sera enregistré sur le disque dure. On pourra faire appel à elle.



### Ajouter du texte et des couleurs

Ajouter des couleurs

Pour cela nous utilisons la fonction imagecolorallocate(image,rouge,vert,bleu)

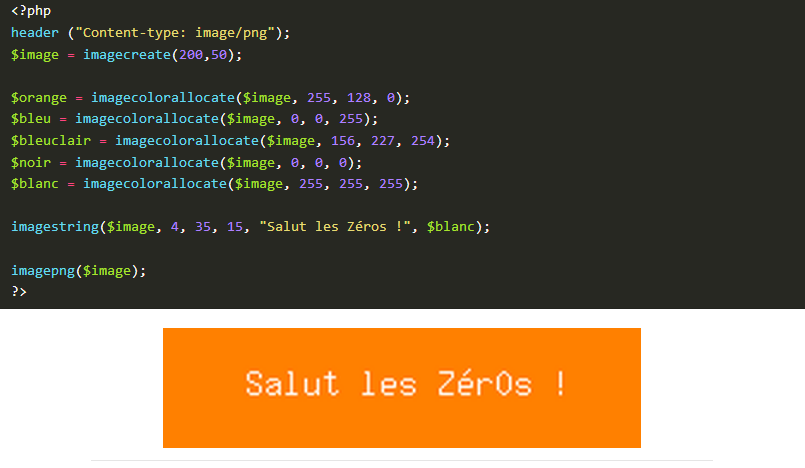


La première couleur sela la couleur de fond de l’image.

Ecrire du texte

Pour cela nous utilisons la fonction imagestring(image,police,x,y,texte,couleur de texte )





### Ajouter des formes

ImageSetPixel (image,x,y,couleur):Ajoute un pixel aux coordonnée x et y.

ImageLine (image, x1, y1, x2, y2, couleur):Ajoute une droite de coordonnées (x1,y1) (x2,y2).

ImageEllipse (image, x, y, largeur, hauteur, couleur):Ajoute une Ellipse.

ImageRectangle (image, x1, y1, x2, y2, couleur): Ajoute une Rectangle.

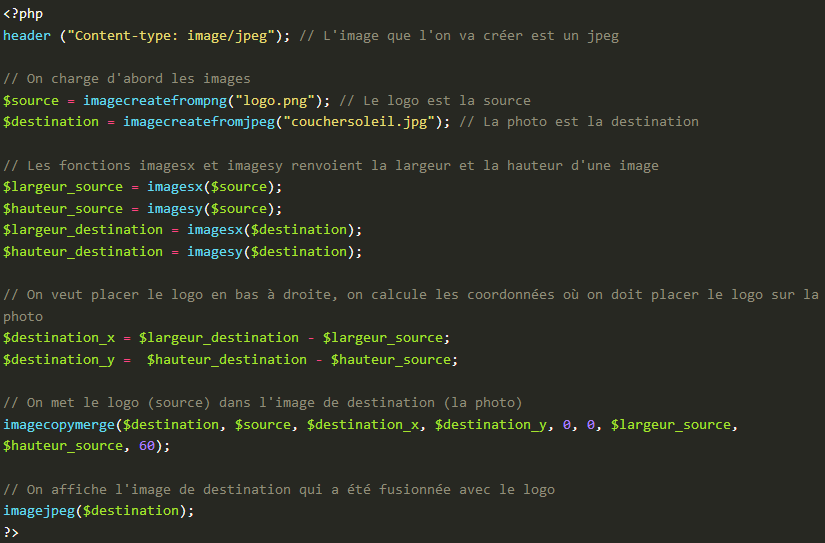
ImagePolygon (image, array\_points, nombre\_de\_points, couleur): Ajoute un polygon.

Rendre une image transparente

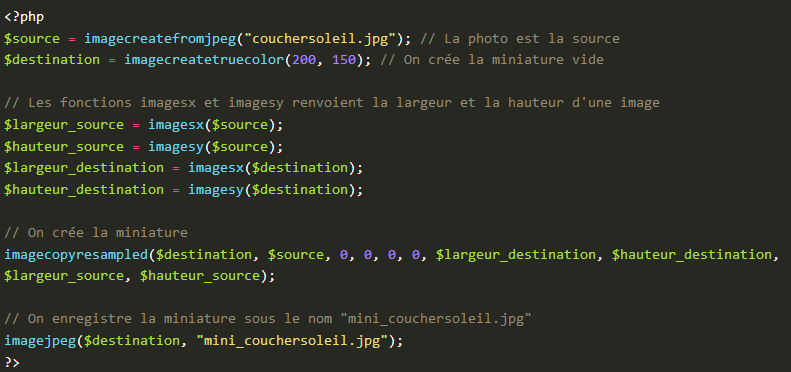
Utiliser la fonction : Imagecolortransparent(image, couleur de fond): on rendra le fond transparent.

Mélanger deux images.

Il faut utiliser la fonction : imagecopymerge



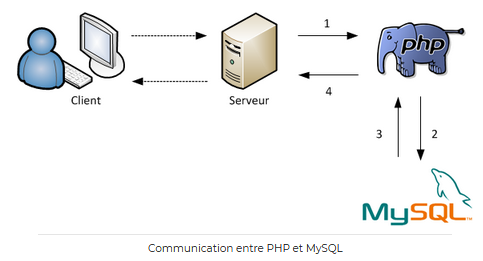
Redimensionner une image

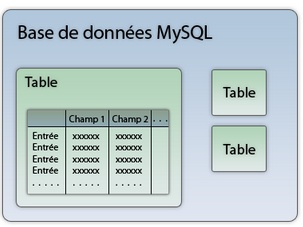


# Langage SQL

## Introduction

La base de données (BDD) est un système qui enregistre des informations. Un peu comme un fichier texte ? Non, pas vraiment. Ce qui est très important ici, c'est que ces informations sont toujours classées. Et c'est ça qui fait que la BDD est si pratique : c'est un moyen simple de ranger des informations.





* Une base de données est un outil qui stocke vos données de manière organisée et vous permet de les retrouver facilement par la suite.
* On communique avec MySQL grâce au langage SQL. Ce langage est commun à tous les systèmes de gestion de base de données (avec quelques petites différences néanmoins pour certaines fonctionnalités plus avancées).
* PHP fait l'intermédiaire entre vous et MySQL.
* Une base de données contient plusieurs tables. Chaque table est un tableau où les colonnes sont appelées « champs » et les lignes « entrées ».

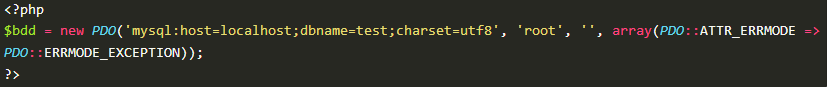
## Connexions à la Base de donnée

Nous allons avoir besoin de quatre renseignements :

* **le nom de l'hôte :** c'est l'adresse de l'ordinateur où MySQL est installé (comme une adresse IP). Le plus souvent, MySQL est installé sur le même ordinateur que PHP : dans ce cas, mettez la valeur localhost (cela signifie « sur le même ordinateur »). Néanmoins, il est possible que votre hébergeur web vous indique une autre valeur à renseigner (qui ressemblerait à ceci :sql.hebergeur.com). Dans ce cas, il faudra modifier cette valeur lorsque vous enverrez votre site sur le Web ;
* **la base :** c'est le nom de la base de données à laquelle vous voulez vous connecter.
* **Le port :** indique le port du serveur utilisé.
* **le login :** il permet de vous identifier. Renseignez-vous auprès de votre hébergeur pour le connaître. Le plus souvent (chez un hébergeur gratuit), c'est le même login que vous utilisez pour le FTP ;
* **le mot de passe :** il y a des chances pour que le mot de passe soit le même que celui que vous utilisez pour accéder au FTP. Renseignez-vous auprès de votre hébergeur.



Pour activer et traquer les erreurs :





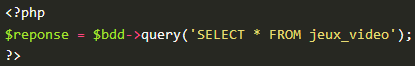
## Lire et Ecrire les données

Pour lire et écrire les données d’une base il faut :

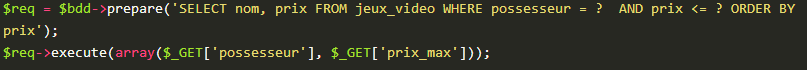
1. Se connecter à la base de données



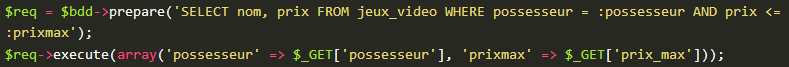
1. Utiliser la méthode appropriée ( $bdd->query , $bdd->prepare , $bdd->exec ) et enregistré la valeur dans une variable.



Ou



Ou



Ou

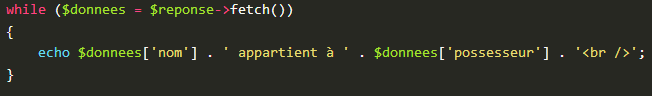


La méthode renvoie le nombre de ligne modifiées.

1. Lire une ligne de la bdd en utilisant fetch() ;



1. Ou lire plusieurs ligne de la bdd en utilisant une boucle while



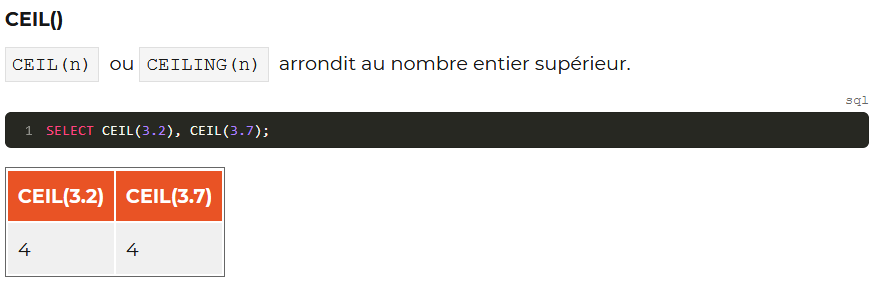
1. Utiliser la méthode closeCursor() Elle provoque la « fermeture du curseur d'analyse des résultats ». Cela signifie, en d'autres termes plus humains, que vous devez effectuer cet appel à closeCursor()chaque fois que vous avez fini de traiter le retour d'une requête, afin d'éviter d'avoir des problèmes à la requête suivante. Cela veut dire qu'on a terminé le travail sur la requête.

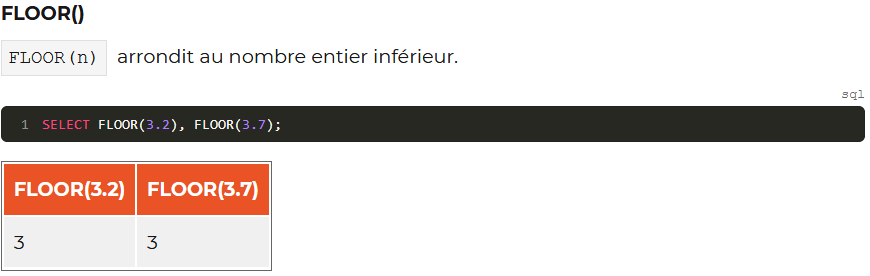


## Les fonctions SQL

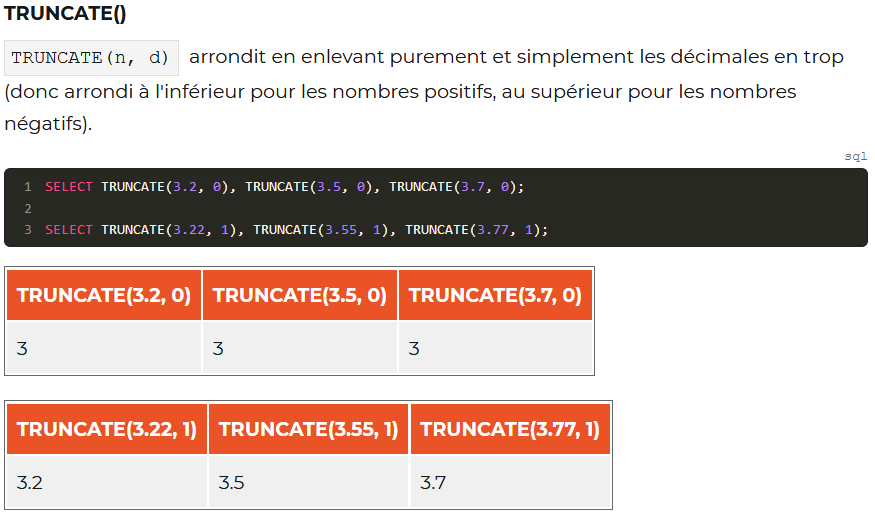
### Les fonctions scalaires

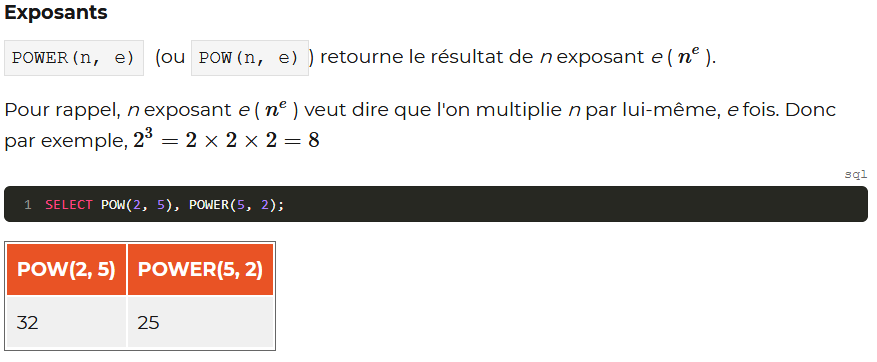
Elles agissent sur chaque entrée. Par exemple, vous pouvez transformer en majuscules la valeur de chacune des entrées d'un champ ;

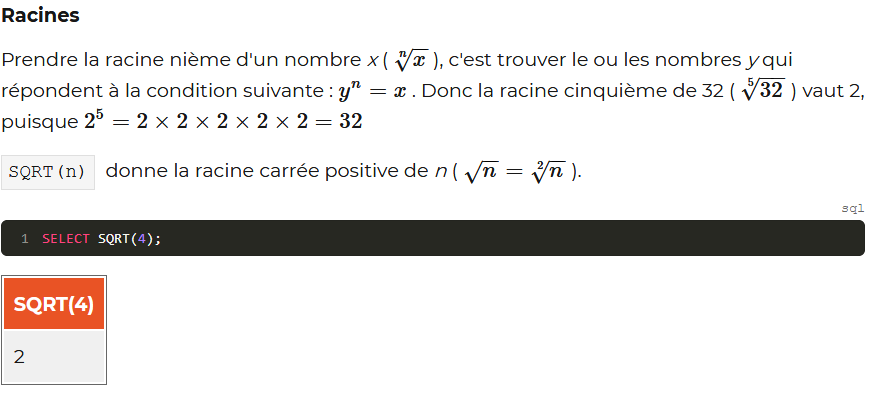


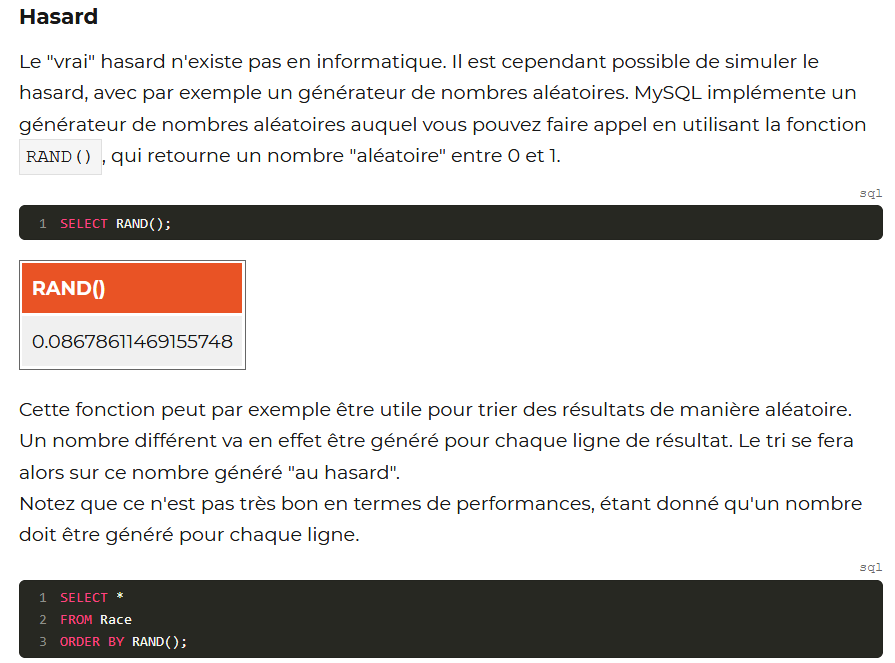


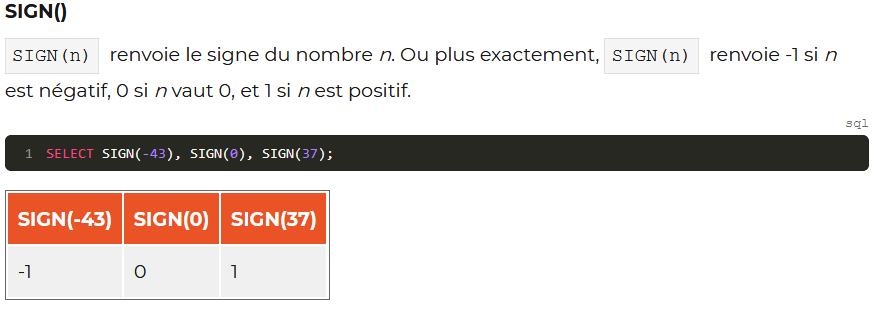


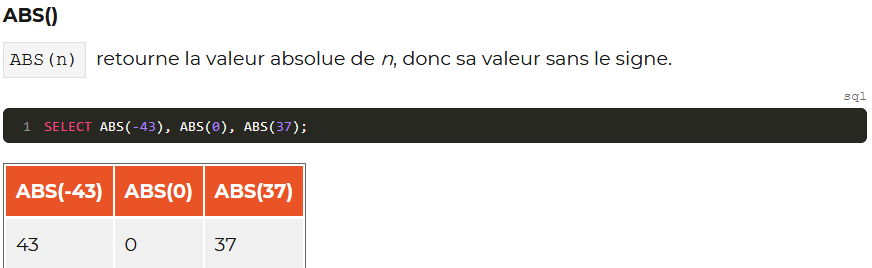


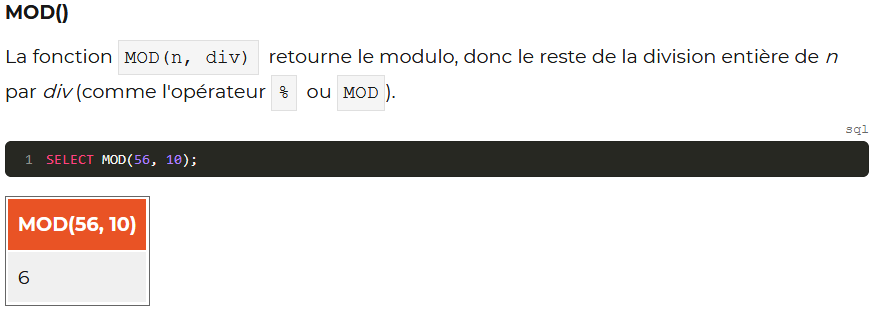




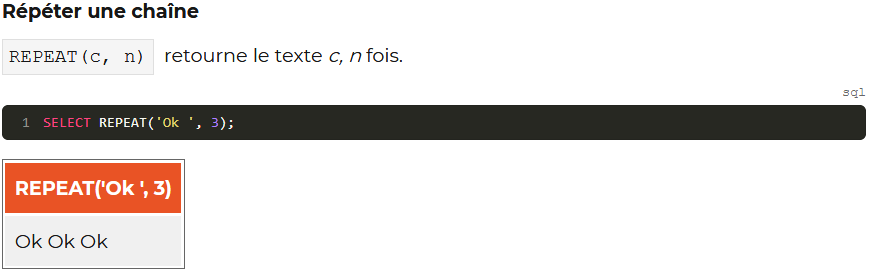






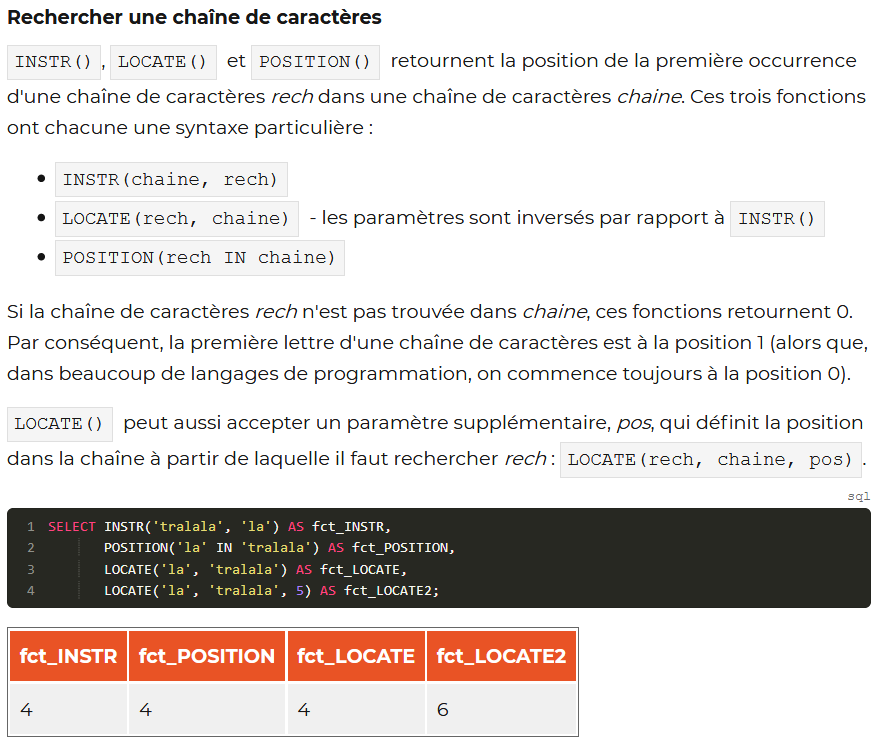




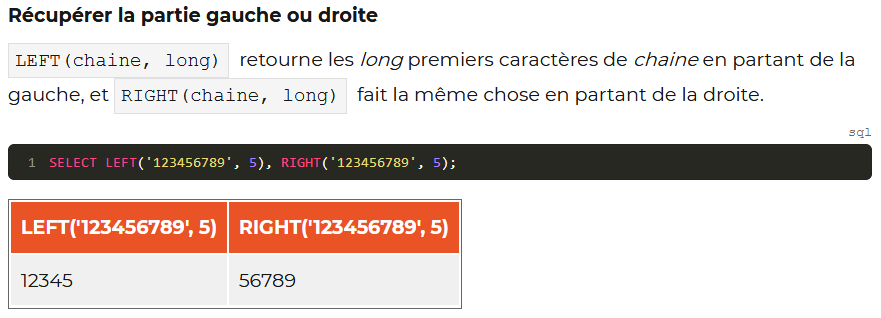


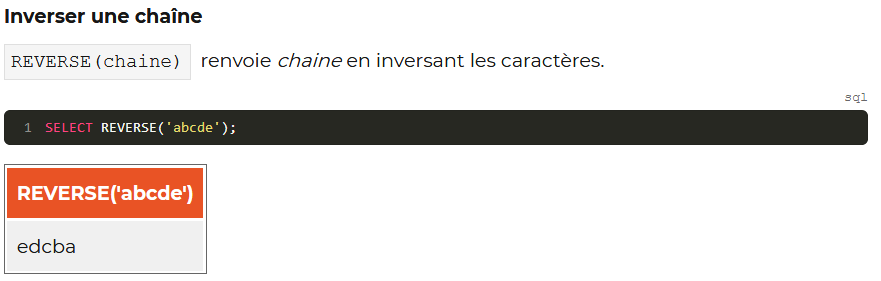




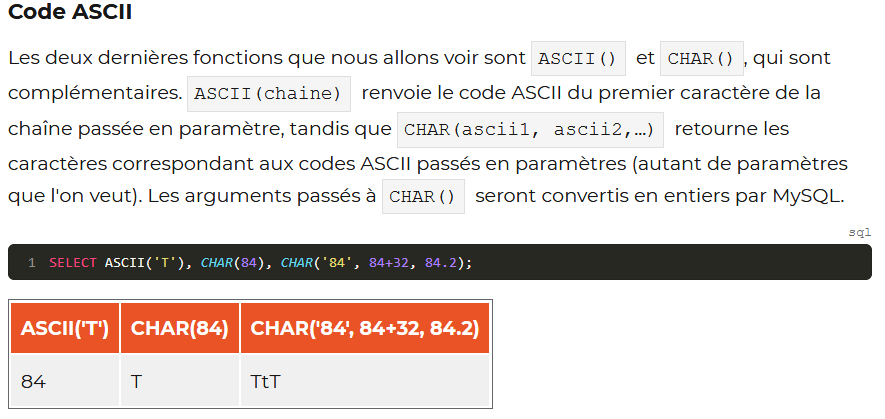


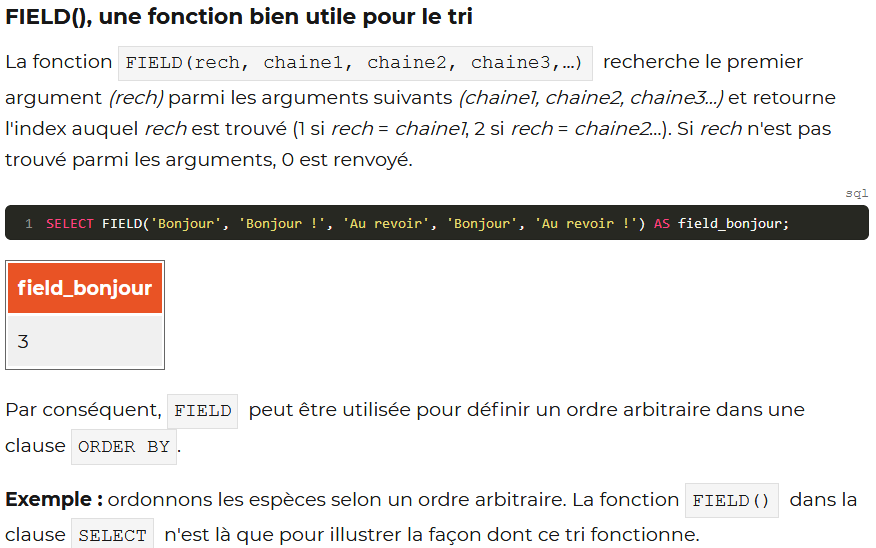


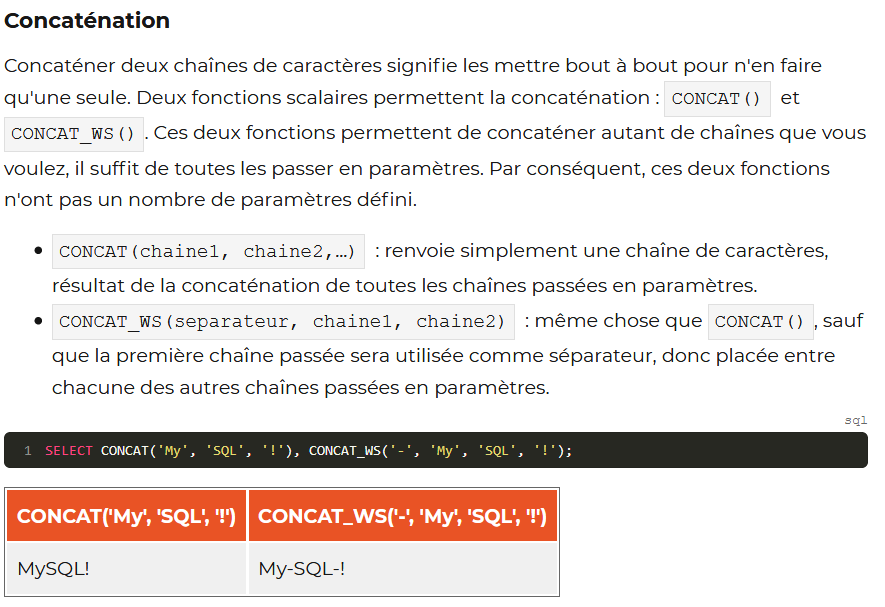






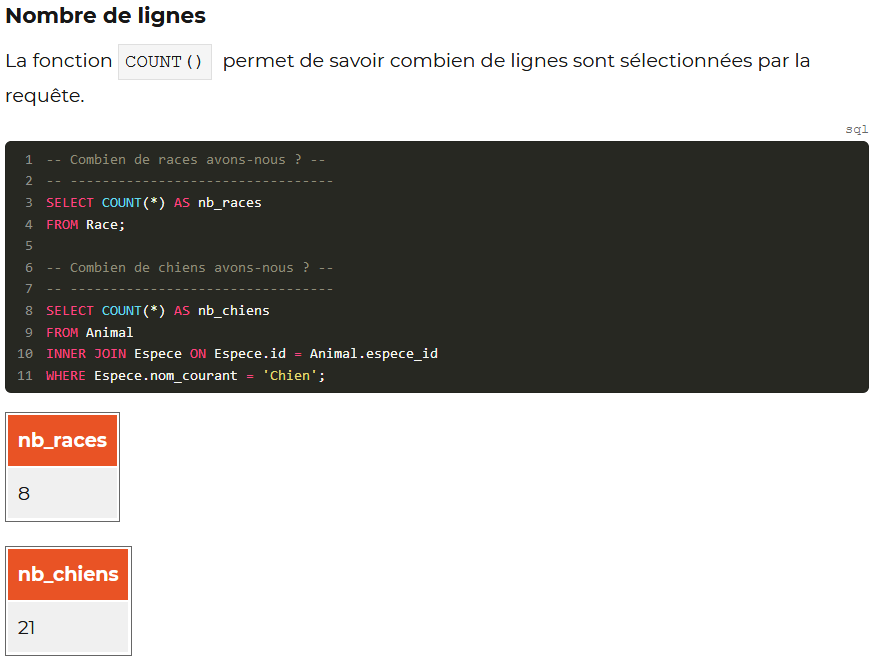


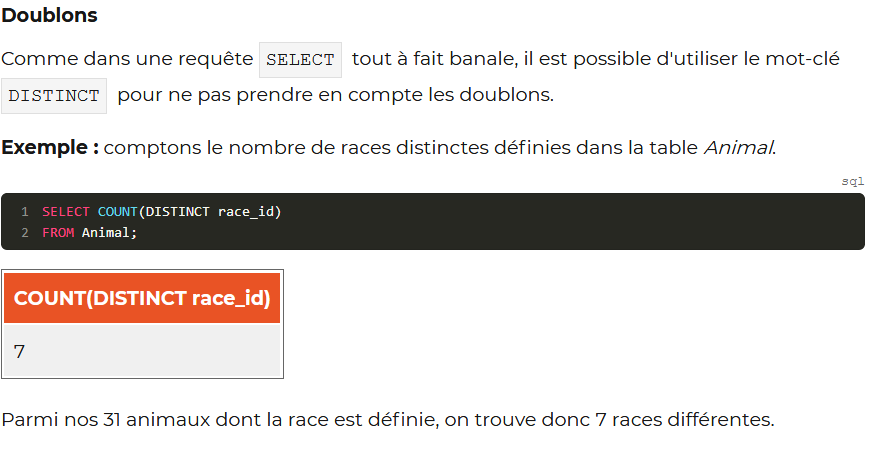




### Les fonctions d’agrégat

Lorsque vous utilisez ce type de fonctions, des calculs sont faits sur l'ensemble de la table pour retourner une valeur. Par exemple, calculer la moyenne des prix retourne une valeur : le prix moyen.

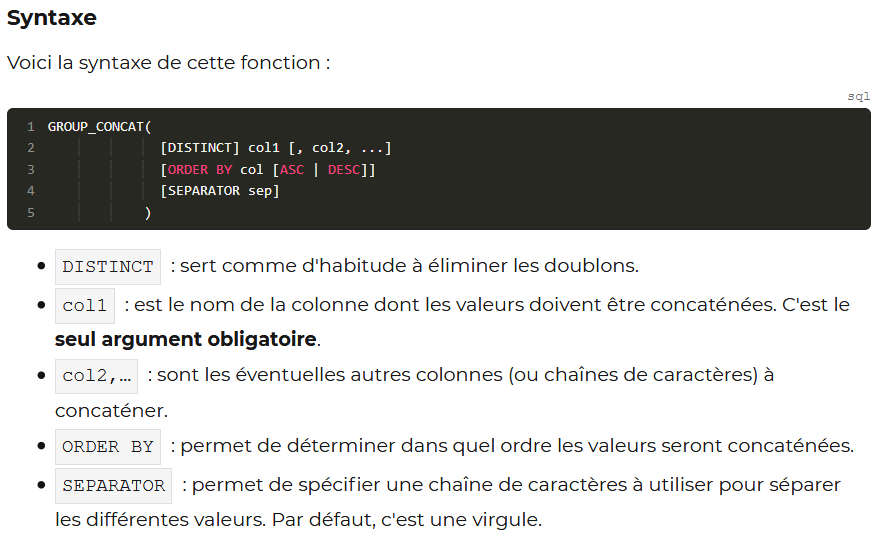












## Les dates en SQL

### Les différents types de dates

Voici les différents types de dates que peut stocker MySQL :

* DATE : stocke une date au format AAAA-MM-JJ (Année-Mois-Jour) ;



* TIME : stocke un moment au format HH:MM:SS (Heures:Minutes:Secondes) ;
* DATETIME : stocke la combinaison d'une date et d'un moment de la journée au format AAAA-MM-JJ HH:MM:SS. Ce type de champ est donc plus précis ;



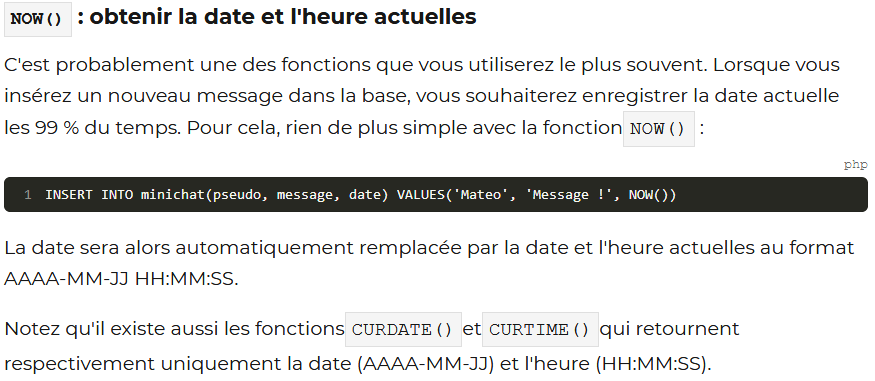
* TIMESTAMP : stocke le nombre de secondes passées depuis le 1er janvier 1970 à 00 h 00 min 00 s ;
* YEAR : stocke une année, soit au format AA, soit au format AAAA.

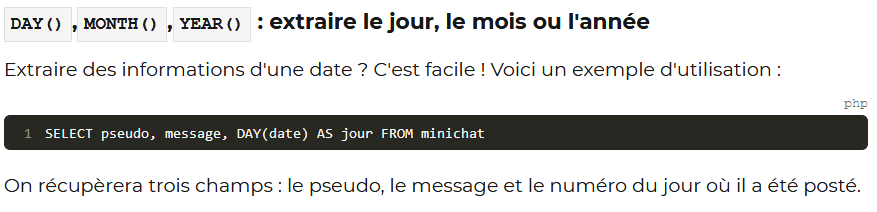
### Les outils des comparaisons de date

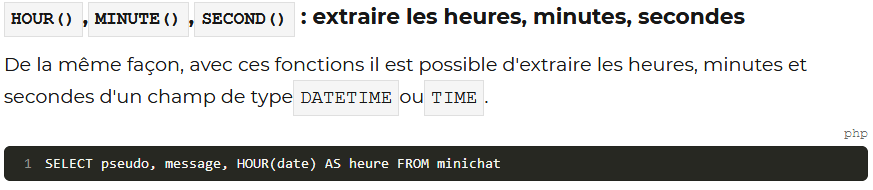


C’est pareil que :

### Les fonctions dates







DATE\_FORMAT

La fonction DATE\_FORMAT permet d’adapter directement la date au format que l’on préfère. Voici comment on l’utilise :



DATE\_ADD

La fonction DATE\_ ADD permet d’ajouter du temps à une date. Voici comment on l’utilise :



DATE\_SUB

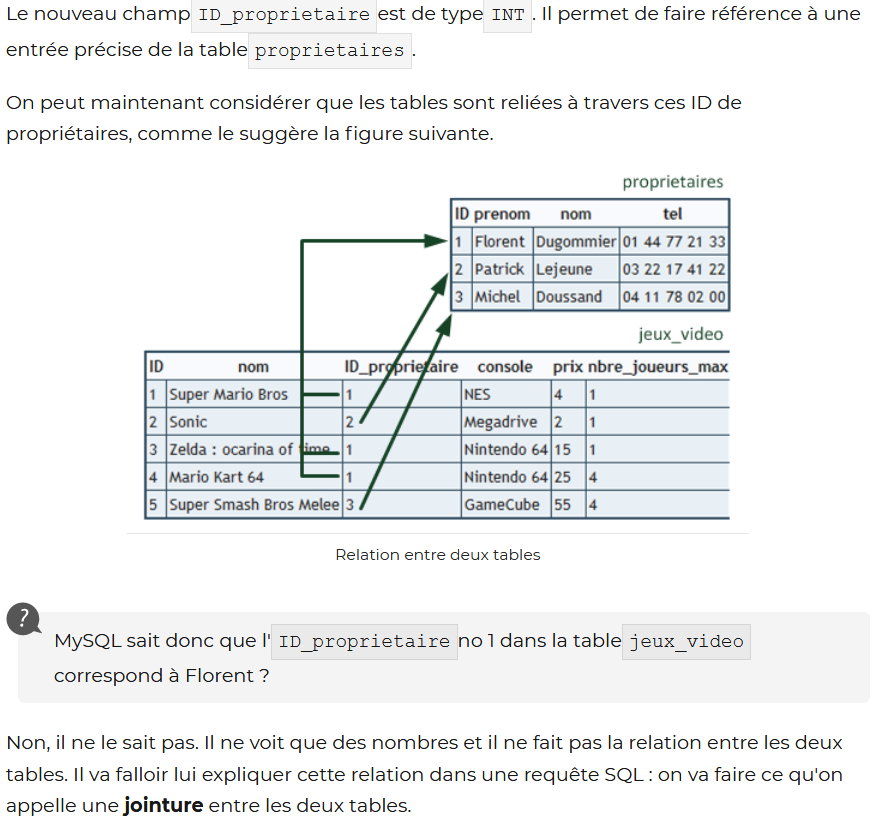
La fonction DATE\_ SUB permet de soustraire du temps à une date. Voici comment on l’utilise :



## Les jointures de tables.

### Explications

Le but des jointures est de lié les tables entre elles. Exemple ci-dessous :



Il existe plusieurs types de jointures, qui nous permettent de choisir exactement les données que l'on veut récupérer. Je vous propose d'en découvrir deux, les plus importantes :

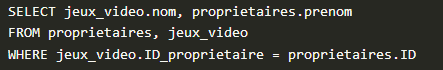
* **les jointures internes :** elles ne sélectionnent que les données qui ont une correspondance entre les deux tables ;
* **les jointures externes :** elles sélectionnent toutes les données, même si certaines n'ont pas de correspondance dans l'autre table.

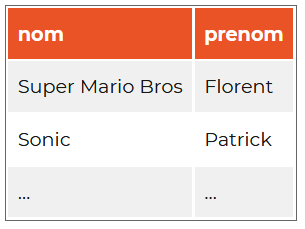
### Les jointures internes

Une jointure interne peut être effectuée de deux façons différentes :

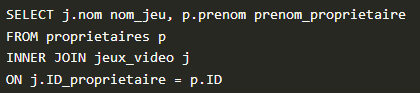
* à l'aide du mot-clé **WHERE** : c'est l'ancienne syntaxe, toujours utilisée aujourd'hui, qu'il faut donc connaître mais que vous devriez éviter d'utiliser si vous avez le choix ;
* à l'aide du mot-clé **JOIN** : c'est la nouvelle syntaxe qu'il est recommandé d'utiliser. Elle est plus efficace et plus lisible.

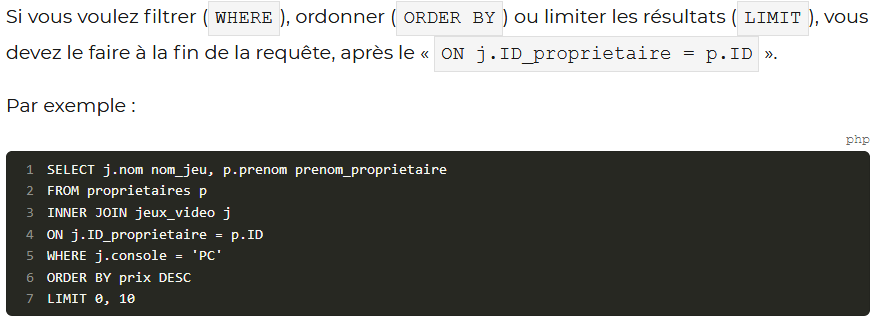
Jointure avec Where





Jointure avec JOIN



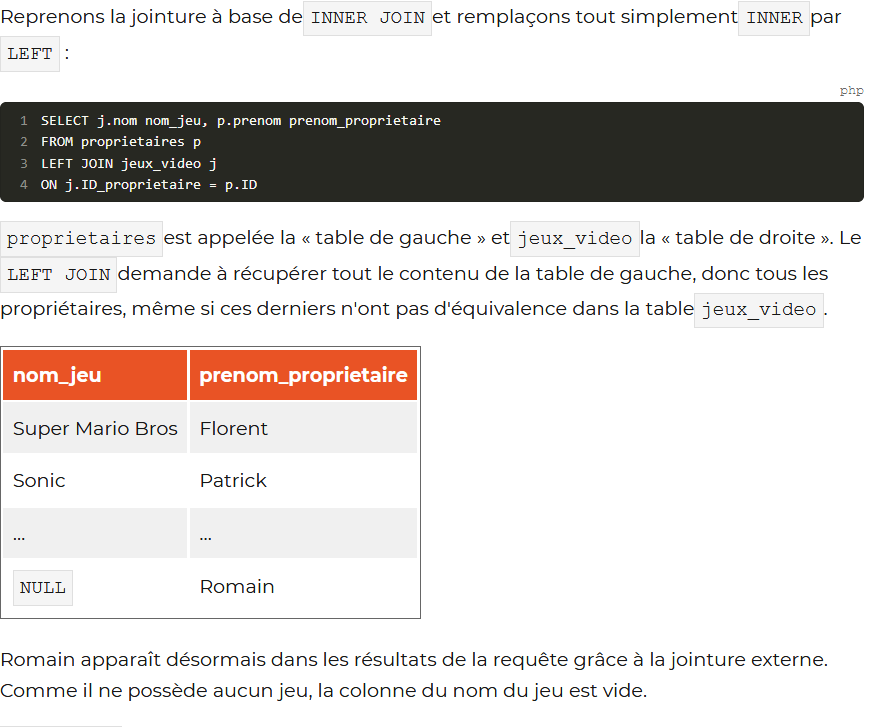


### Les jointures externes

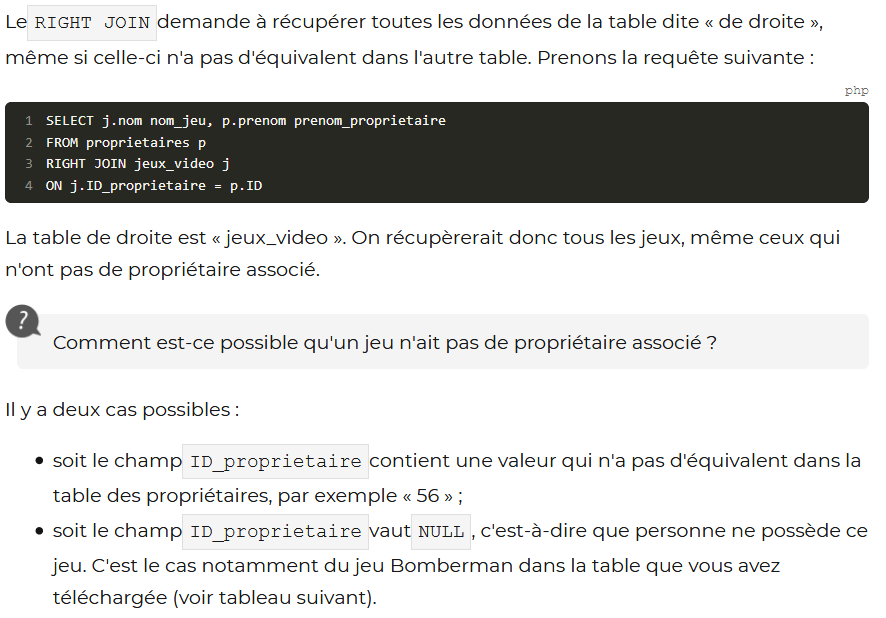
Les jointures externes permettent de récupérer toutes les données, même celles qui n'ont pas de correspondance.

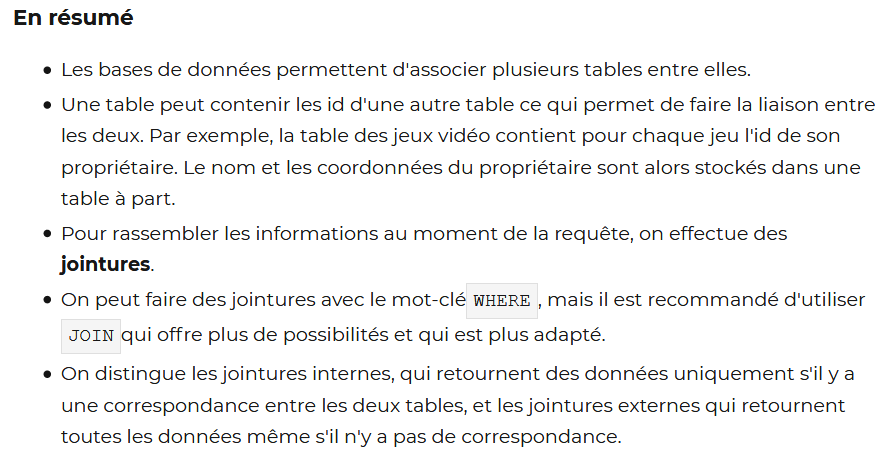
Cette fois, la seule syntaxe disponible est à base de **JOIN**. Il y a deux écritures à connaître : **LEFT JOIN** et **RIGHT JOIN**. Cela revient pratiquement au même, avec une subtile différence que nous allons voir.

Jointure avec LEFT JOIN (récupère tout la table de gauche)



Jointure avec RIGHT JOIN (récupère toute la table de droite)

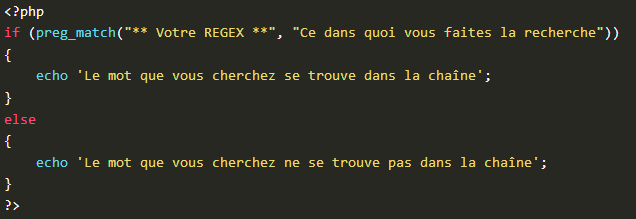




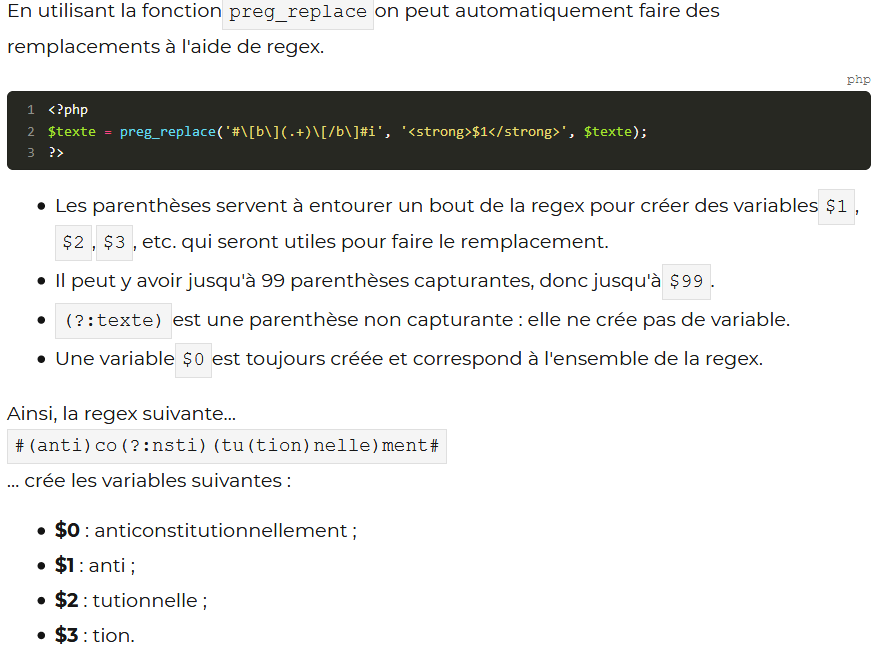
## Les expressions régulières

### Les fonctions

* preg\_grep ;
* preg\_split ;
* preg\_quote ;
* preg\_match ; fait une recherche et retourne FALSE ou TRUE.



* preg\_match\_all ;
* preg\_replace\_callback.
* preg\_replace ;



### Mémento des expressions régulières

| **egex** | **Explication** |
| --- | --- |
| #guitare# | Cherche le mot « guitare » dans la chaîne. |
| #guitare|piano# | Cherche le mot « guitare » OU « piano ». |
| #^guitare# | La chaîne doit commencer par « guitare ». |
| #guitare$# | La chaîne doit se terminer par « guitare ». |
| #^guitare$# | La chaîne doit contenir uniquement « guitare ». |

Les classes de caractères

| **regex** | **Explication** |
| --- | --- |
| #gr[ioa]s# | Chaîne qui contient « gris », ou « gros », ou « gras ». |
| [a-z] | Caractères minuscules de a à z. |
| [0-9] | Chiffres de 0 à 9. |
| [a-e0-9] | Lettres de « a » à « e » ou chiffres de 0 à 9. |
| [0-57A-Za-z.-] | Chiffres de 0 à 5, ou 7, ou lettres majuscules, ou lettres minuscules, ou un point, ou un tiret. |
| #[^0-9]# | Chaîne ne contenant PAS de chiffres. |
| #^[^0-9]# | Chaîne ne commençant PAS par un chiffre. |

Les quantificateurs

| **regex** | **Explication** |
| --- | --- |
| #a?# | « a » peut apparaître 0 ou 1 fois. |
| #a+# | « a » doit apparaître au moins 1 fois. |
| #a\*# | « a » peut apparaître 0, 1 ou plusieurs fois. |
| #bor?is# | « bois » ou « boris ». |
| #Ay(ay|oy)\*# | Fonctionne pour Ay, Ayay, Ayoy, Ayayayoyayayoyayoyoyoy, etc. |
| #a{3}# | « a » doit apparaître 3 fois exactement (« aaa »). |
| #a{3,5}# | « a » doit apparaître de 3 à 5 fois (« aaa », « aaaa », « aaaaa »). |
| #a{3,}# | « a » doit apparaître au moins 3 fois (« aaa », « aaaa », « aaaaa », « aaaaaa », etc.). |

Les Métacaractères

Les métacaractères sont :

# ! ^ $ ( ) [ ] { } | ? + \* .

Pour utiliser un métacaractère dans une recherche, il faut l'échapper avec un antislash : \

| **regex** | **Explication** |
| --- | --- |
| #Hein?# | Cherche « Hei » ou « Hein ». |
| #Hein\?# | Cherche « Hein? ». |

Les abréviations

| **Classe abrégée** | **Correspondance** |
| --- | --- |
| **\d** | [0-9] |
| **\D** | [^0-9] |
| **\w** | [a-zA-Z0-9\_] |
| **\W** | [^a-zA-Z0-9\_] |
| **\t** | Tabulation |
| **\n** | Saut de ligne |
| **\r** | Retour chariot |
| **\s** | Espace blanc (correspond à\t\n\r) |
| **\S** | N'est PAS un espace blanc (\t\n\r) |
| **.** | Classe universelle |

Les Options

Il existe de nombreuses options que l'on peut utiliser avec les regex PCRE.

Parmi les trois que nous sommes le plus souvent amenés à utiliser, il y a :

* i : la regex ne fera plus la différence entre majuscules / minuscules ;
* s : le point (classe universelle) fonctionnera aussi pour les retours à la ligne (\n) ;
* U : mode « Ungreedy » (pas gourmand). Utilisé pour que la regex s'arrête le plus tôt possible. Pratique par exemple pour le bbCode[b][/b] : la regex s'arrêtera à la première occurrence de[/b].