Création d'un site dynamique html/css/php/javascript

Site avec contenu statique (html + css)

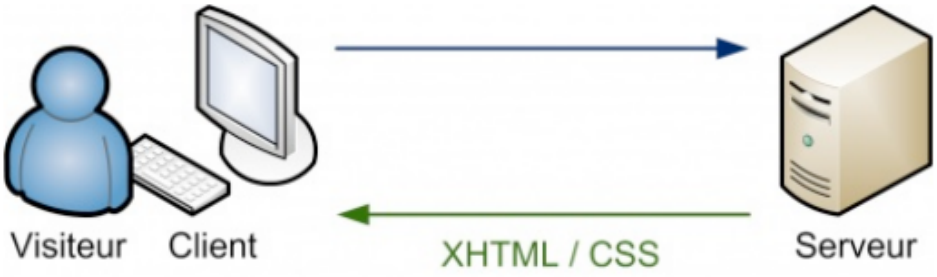
Jusqu'à présent, vous avez créé un site statique, le contenu des pages est créé par un programmeur en utilisant les langages **HTML** et **CSS**.
Exemple : site de vente de chaussures qui présente la catégorie vente enfant, un numéro de téléphone est communiqué pour passer des commandes.

Si on désire faire évoluer ce contenu, il faut contacter le programmeur afin qui apporte des modifications ce qui représente toujours un coût financier.
Exemple : évolution du site précédent pour présenter d'autres catégories, ajouter un courriel de contact.

1. Ressources nécessaires :

- un navigateur internet.
- un hébergement web pour stockage des fichiers ou hébergement local.

2. Circulation de l'information :



Le serveur envoie du XHTML et du CSS au client

Illustration tirée du site du zéro.

Site avec contenu dynamique

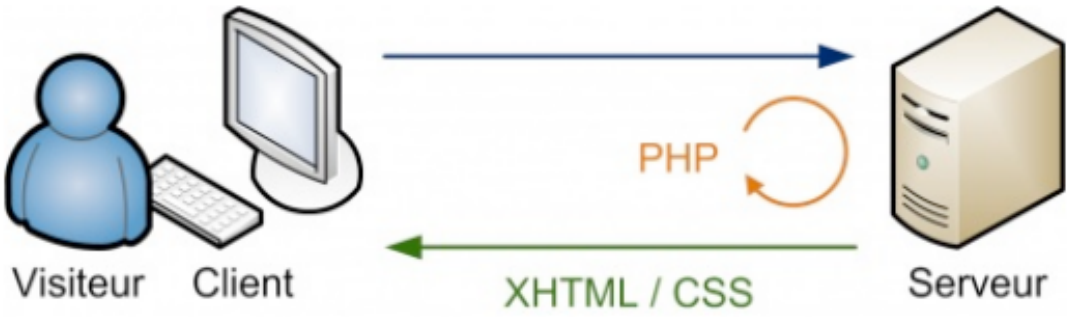
Dès que vous aurez besoin de créer des sites permettant de gérer des clients, produits, article, etc... vous aurez besoin de générer des pages web qui afficheront des informations en réponse à des "actions utilisateurs".
Exemples :

- inscription => génération d'un formulaire de saisie en html, la validation de ce formulaire viendra alimenter une table d'une base de données ou un fichier xml, etc...
- choix d'articles à commander => choix d'une catégorie puis d'un produit puis ajout de ce dernier dans un panier de commandes.
- affichage de news => les news d'un site pourront s'afficher selon la date du jour; la saison, etc...

1. Ressources nécessaires :

- un navigateur internet.
- un hébergement web pour stockage des fichiers ou hébergement local.
- un serveur web / serveur de base de données
- support d'au moins un langage de script (php, python, perl, etc...)

2. Circulation de l'information :



PHP décide ce qui va être affiché sur la page web envoyée au visiteur

Concepts à maitriser

1. Les balises php

Vous pouvez insérer des balises php dans vos pages web, n'importe où.
N'oubliez pas de ne pas faire d'erreurs de syntaxe php.

Exercice : entourez les parties écrites en php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<?php
/* un commentaire
*/
// un commentaire
?>
<head>
<title>Ceci est une page de test avec des balises
PHP</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
</head>
<body>
<h2>Page de test</h2>
<p>
Cette page contient du code (x)HTML avec des balises
PHP.<br />
<?php /* Insérer du code PHP ici */ ?>
```

```
Voici quelques petits tests :
</p>
<ul>
<li style="color: blue;">Texte en bleu</li>
<li style="color: red;">Texte en rouge</li>
<li style="color: green;">Texte en vert</li>
</ul>
<?php
/* Encore du PHP
Toujours du PHP */
?>
</body>
</html>
```

2. Notions de base PHP

- echo
- //
- /* ... */
- variables int float bool string
- if (){}
- if() {} else {}
- switch(){}
- while () {}
- for () {}
- function nom(){}

3. Les tableaux en php : array.

Vous allez voir qu'il s'agit de variables "composées", que l'on peut imaginer sous la forme de tableau.

3.1.Les deux types de tableaux

Un tableau (array) est une variable. Mais une variable un peu spéciale.
Reprenons. Jusqu'ici vous avez travaillé avec des variables toutes simples : elles ont un nom et une valeur.
Par exemple :

```
<?php
$prenom = 'Nicole';
echo 'Bonjour ' . $prenom; // Cela affichera : Bonjour Nicole
?>
```

Ce qui peut se matérialiser sous la forme :
Nom Valeur
$prenom Nicole

Nom	Valeur
$prenom	Nicole

Ici, nous allons voir qu'il est possible d'enregistrer de nombreuses informations dans une seule variable (bien plus que "Nicole") grâce aux tableaux.
On distingue deux types de tableaux:

- Les tableauxnumérotés
- Les tableauxassociatifs

3.2.Les tableaux numérotés

Ces tableaux sont très simples à imaginer. Regardez par exemple ce tableau, contenu de la variable $prenoms :

Clé	Valeur
0	François
1	Michel
2	Nicole
3	Véronique
4	Benoît
...	

$prenoms est un array : c'est ce qu'on appelle une variable "tableau". Elle n'a pas qu'une valeur mais plusieurs valeurs (vous
pouvez en mettre autant que vous voulez).

Dans un array, les valeurs sont rangées dans des "cases" différentes. Ici, nous travaillons sur un array numéroté, c'est-à-dire que chaque case est identifiée par un numéro. Ce numéro est appelé **clé**.

Un array numéroté commence toujours à la case n°0 ! Ne l'oubliez jamais, ou vous risquez de faire des erreurs par la suite...

Construire un tableau numéroté
Pour créer un tableau numéroté en PHP, on utilise généralement la fonction array.
Cet exemple vous montre comment créer l'array $prenoms :

```
<?php
// La fonction array permet de créer un array
$prenoms = array ('François', 'Michel', 'Nicole', 'Véronique',
'Benoît');
?>
```

L'ordre a beaucoup d'importance. Le premier élément ("François") aura le n°0, ensuite Michel le n°1, etc.

Vous pouvez aussi créer manuellement le tableau case par case :

```
<?php
$prenoms[0] = 'François';
$prenoms[1] = 'Michel';
$prenoms[2] = 'Nicole';
?>
```

Si vous ne voulez pas avoir à écrire vous-même le numéro de la case que vous créez, vous pouvez laisser PHP le sélectionner automatiquement en laissant les crochets vides :

```
<?php
$prenoms[] = 'François'; // Créera $prenoms[0]
$prenoms[] = 'Michel'; // Créera $prenoms[1]
$prenoms[] = 'Nicole'; // Créera $prenoms[2]
?>
```

Afficher un tableau numéroté

Pour afficher un élément, il faut donner sa position entre crochets après $prenoms. Cela revient à dire à PHP :
Affiche-moi le contenu de $prenoms issu de la case numéro 1

Pour faire cela en PHP, il faut écrire le nom de la variable, suivi du numéro entre crochets. Pour afficher "Michel", on doit donc écrire :

```
<?php
echo $prenoms[1];
?>
```

3.3.Les tableaux associatifs

Les tableaux associatifs fonctionnent sur le même principe, sauf qu'au lieu de numéroter les cases, on va les étiqueter en leur donnant à chacune un nom différent.

Par exemple, supposons que je veuille, dans un seul array, enregistrer les coordonnées de quelqu'un (nom, prénom, adresse, ville, etc...). Si l'array est numéroté, comment savoir que le n°0 est le nom, le n°1 le prénom, le n°2 l'adresse ?... C'est là que deviennent utiles les tableaux associatifs.

Construire un tableau associatif

Pour les créer, on utilisera la fonction array comme tout à l'heure, mais on va mettre "l'étiquette" devant chaque information :

```
<?php
// On crée notre array $coordonnees
```

```
$coordonnees = array (
'prenom' => 'François',
'nom' => 'Dupont',
'adresse' => '3 Rue du Paradis',
'ville' => 'Marseille');
?>
```

Note importante : il n'y a ici qu'une seule instruction (un seul point-virgule). J'aurais pu tout mettre sur la même ligne, mais rien ne m'empêche de séparer ça sur plusieurs lignes pour que ça soit plus facile à lire.

Vous remarquez qu'on écrit une flèche (=>) pour dire "associé à".
Par exemple, on dit "ville associé à Marseille".

Nous avons créé un tableau qui ressemble à la structure suivante :

Clé	Valeur
prenom	François
nom	Dupont
adresse	3 rue du paradis
ville	Marseille

Il est aussi possible de créer le tableau case par case comme ceci :

```
<?php
$coordonnees['prenom'] = 'François';
$coordonnees['nom'] = 'Dupont';
$coordonnees['adresse'] = '3 Rue du Paradis';
$coordonnees['ville'] = 'Marseille';
?>
```

Afficher un tableau associatif
Pour afficher un élément, il suffit d'indiquer le nom de cet élément entre crochets, ainsi qu'entre guillemets ou apostrophes puisque l'étiquette du tableau associatif est un texte.
Par exemple, pour extraire la ville, on devra taper :

```
<?php
echo $coordonnees['ville'];
?>
```

Parcourir un tableau
Lorsqu'un tableau a été créé, on a souvent besoin de le parcourir pour savoir ce qu'il contient. Nous allons voir trois moyens
d'explorer un array :

La boucle for

Il est très simple de parcourir un tableau numéroté avec une boucle for. En effet, puisqu'il est numéroté à partir de 0, on peut
faire une boucle for qui incrémente un compteur à partir de 0 :

```
<?php
// On crée notre array $prenoms
$prenoms = array ('François', 'Michel', 'Nicole', 'Véronique',
'Benoît');
```

Puis on fait une boucle pour tout afficher :

```
for ($numero = 0; $numero < 5; $numero++)
{
echo $prenoms[$numero] . '<br />'; // affichera $prenoms[0], $prenoms[1] etc...
}
?>
```

Quand on écrit $prenoms[$numero], la variable $numero est d'abord remplacée par sa valeur.
Par exemple, si $numero vaut 2, alors cela signifie qu'on cherche à obtenir ce que contient $prenoms[2], c'est-à-dire... Nicole.

La boucle foreach

C'est une **boucle** for **spécialisée** dans les **tableaux**.
"foreach" va passer en revue chaque ligne du tableau, et lors de chaque passage, elle va mettre la valeur de cette ligne dans une variable temporaire (par exemple $element).

```
<?php
$prenoms = array ('François', 'Michel', 'Nicole', 'Véronique',
'Benoît');
foreach($prenoms as $element)
{
echo $element . '<br />'; // affichera $prenoms[0], $prenoms[1]
etc...
}
?>
```

C'est le même code que tout à l'heure mais basé ici sur une boucle foreach.
A chaque tour de boucle, la valeur de l'élément suivant est mise dans la variable $element. On peut donc utiliser $element uniquement à l'intérieur de la boucle afin d'afficher l'élément en cours.
L'avantage de foreach est qu'il permet aussi de parcourir les tableaux associatifs.

```
<?php
$coordonnees = array (
'prenom' => 'François',
'nom' => 'Dupont',
```

```
'adresse' => '3 Rue du Paradis',
'ville' => 'Marseille');
foreach($coordonnees as $element)
{
echo $element . '<br />';
}
?>
```

foreach va mettre tour à tour dans la variable $element le prénom, le nom, l'adresse et la ville contenus dans l'array $coordonnees.

On met donc entre parenthèses :
1. D'abord le nomde l'array (ici $coordonnees)
2. Ensuite le mot-clé **as** (qui signifie quelque chose comme "en tant que")
3. Enfin le nom d'une variable que vous choisissez qui va contenir tour à tour chacun des éléments de l'array (ici $element).

Entre les accolades, on n'utilise donc que la variable $element.
La boucle s'arrête lorsqu'on a parcouru tous les éléments de l'array.

Toutefois, avec cet exemple on ne récupère que la valeur. Or, on peut aussi récupérer la clé de l'élément. On doit dans ce cas écrire foreach comme ceci :

```
<?php foreach($coordonnees as $cle => $element) ?>
A chaque tour de boucle, on disposera non pas d'une mais de deuxvariables :
$cle : elle contiendra la clé de l'élément en cours d'analyse ("prenom", "nom", etc.).
$element : il contiendra la valeur de l'élément en cours ("François", "Dupont", etc.).
```

Testons le fonctionnement avec un exemple :

```
<?php
$coordonnees = array (
'prenom' => 'François',
'nom' => 'Dupont',
'adresse' => '3 Rue du Paradis',
'ville' => 'Marseille');
foreach($coordonnees as $cle => $element)
{
echo '[' . $cle . '] vaut ' . $element . '<br />';
}
?>
```

Avec cette façon de procéder, vous avez maintenant dans la boucle la clé ET la valeur.

La fonction print_r (utilisée principalement pour le débuggage)

Afficher rapidement un array avec print_r
C'est une sorte de echo spécialisé dans les array.
Cette commande a toutefois un défaut : elle ne renvoie pas de code HTML comme
 pour les retours à la ligne. Pour bien
voir les retours à la ligne, il faut donc utiliser la balise HTML <pre> qui nous permet d'avoir un affichage plus correct.

```
<?php
$coordonnees = array (
'prenom' => 'François',
'nom' => 'Dupont',
'adresse' => '3 Rue du Paradis',
'ville' => 'Marseille');
echo '<pre>';
print_r($coordonnees);
echo '</pre>';
?>
```

Voilà, c'est facile à utiliser du temps qu'on n'oublie pas la balise <pre> pour avoir un affichage correct.
Bien entendu, vous n'afficherez jamais des choses comme ça à vos visiteurs. On peut en revanche s'en servir pour le débuggage, pendant la création du site, afin de voir rapidement ce que contient l'array.

Outils de recherche intégré au langage php

- array_key_exists : pour vérifier si une clé existe dans l'array
- in_array : pour vérifier si une valeur existe dans l'array
- array_search : pour récupérer la clé d'une valeur dans l'array

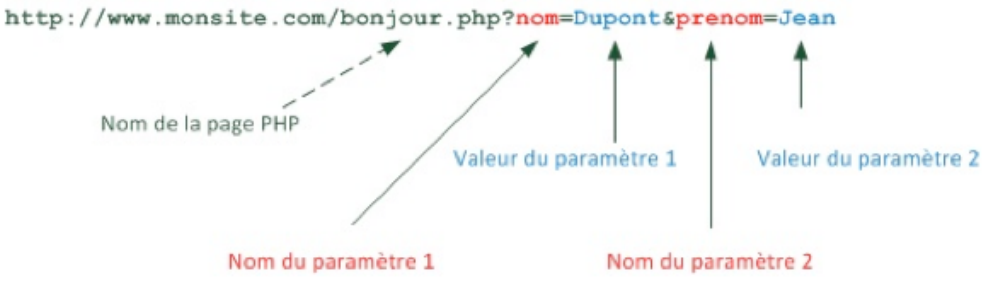
4. Transmission de données d'une page vers une autre

Pour gérer des données en utilisant vos pages web, il vous faudra transmettre des informations d'une page web vers une autre.
Par exemple, choix d'une date sur un calendrier => transmission de cette date à un script php qui va :

- ➡ créer une page de news correspondant à cette date,
- ➡ retourner cette page créée pour finalement l'afficher dans le navigateur.

4.1. Transmission par l'URL

Voici un exemple d'URL.



Cette URL permet de récupérer du code HTML+CSS+JAVASCRIPT. Ce code est créé par le code contenu dans le fichier 'bonjour.php'.
Ce fichier reçoit en paramètres d'entrée :

- nom => Dupont
- prenom => Jean

Ce fichier pourra manipuler ces données qui sont stockées dans le **tableau associatif $_GET ou $_REQUEST**. Sur l'exemple, il pourra manipuler $_GET['nom'] et $_GET['prenom'] ou $_REQUEST['nom'] et $_REQUEST['prenom'].

Attention : les majuscules sont obligatoires!!!!!!

4.2.Utilisation de cette technique dans des liens hypertextes.

Afin de pouvoir afficher telle ou telle catégorie de produit dans votre site, vous pourrez créer des liens hypertextes transmettant la catégorie au script qui affiche les catégories de produit.

"Fichier index.html"

```
.....
<a href="affichageProduit.php?categorie=enfant">Chaussures enfant</a>
<a href="affichageProduit.php?categorie=adulte">Chaussures enfant</a>
<a href="affichageProduit.php?categorie=senior">Chaussures enfant</a>
.....
```

"Fichier affichageProduit.php"

```
....
<?php
        if (isset($_GET['categorie'])){
                // récupération des données sur les produits correspondants
                // à la catégorie contenue dans $_GET['categorie']
                $categ= $_GET['categorie'];
                //exécution d'une requête SQL récupérant les produits
                //affichage des produits
                //pour test ici, on affiche uniquement la catégorie
                echo "<br /><p>Voici la catégorie choisie : $categ</p>";
        }else{
                //Quand passe-t-on dans cette partie du script???
                echo "Erreur, la catégorie est absente";

        }
 ?>
```

isset($uneVariable), signifie is set(variable) => est-ce que variable existe ?
Cette instruction retourne true ou false, true si la variable existe, false sinon.

Exercice 1 :

Testez le code des fichiers index.html et affichageProduit.php.
Cela fonctionne-t-il ?

Exercice 2 :

Modifiez ces fichiers pour afficher d'autres catégories de produits.

Exercice 3 :

Refaire ce travail mais en utilisant un seul fichier nommé "index.php".
Ce fichier contient les liens hypertextes mais aussi l'affichage des produits de la catégorie si l'on a cliqué sur un des liens.

5. Les formulaires

Un formulaire HTML permet de saisir des informations et de les transmettre à un script php.

5.1.La balise form et les champs de saisie.

Un formulaire contient :
une balise form avec les attributs suivants :

action => "lien vers le script php"
method => "get" ou "post"
name => nom du formulaire (interne au document actif)

des champs de saisie :

balise input

type
"text", zone de texte,
Attributs : value (valeur par défaut), name (nom), id,
"hidden", zone de texte cachée,
Attributs : value (valeur par défaut), name (nom), id,
"password", zone de texte avec données cachées (*****)
Attributs : value (valeur par défaut), name (nom), id,
"button", bouton personnalisable
Attributs : value (valeur par défaut), name (nom), id,
"submit", bouton de soumission de formulaire
Attributs : value (valeur par défaut), name (nom), id,
"reset", bouton de réinitialisation des champs du formulaire
Attributs : value (valeur par défaut), name (nom), id,
"select", zone de liste déroulante
Attributs : value (valeur par défaut), name (nom), id,
contient des lignes définies par la balise option
<option value="valeur">valeur affichée</option>
<option value="valeur2">valeur affichée2</option>
"checkbox", cases à cocher
Attributs : value (valeur par défaut), name (nom), id, checked
"radio", boutons radio
Attributs : value (valeur par défaut), name (nom), id, checked

balise textarea

Attributs :

Attributs : value (valeur par défaut), name (nom), id, checked, rows, cols

5.1.1. Exercice 1 - formulaire=>traitement :

créer le **formulaire** (id="monForm" action="traitement" method="post") qui permet ,

- la saisie de deux informations :
  - identifiant, (id et name vaudront "identifiant")
  - mot de passe, (id et name vaudront "mot_de_passe")
- la **validation** du formulaire en sachant que :
  - le bouton de soumission du formulaire affichera "**s'identifier**" et **redirigera** la page vers "traitement.php",
- la remise à zéro des champs du formulaire, ce bouton affichera "**raz**",

créer le fichier "**traitement.php**" qui affichera un récapitulatif des informations saisies dans le précédent formulaire, quelle(s) critique(s) pouvez-vous émettre ?

Fichier "formulaire.html"

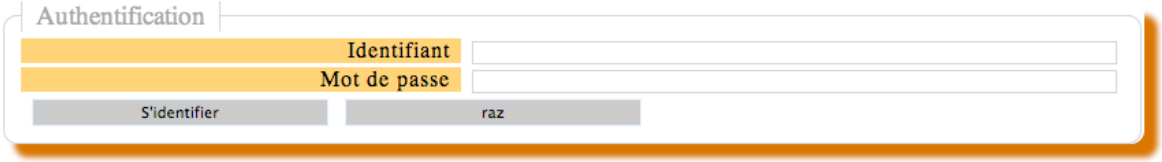
```
<!DOCTYPE html>
<html>

	<head>
		<meta charset="utf-8">
		<title>Formulaire de saisie</title>


	</head>
	<body>
		<form action="./traitement.php" method="post" id="monForm">
			<label for="identifiant">Identifiant</label>
				<input type="text" name="identifiant" id="identifiant" value="" />
				<span id="erreur_identifiant"></span>

				<label for="mot_de_passe">Mot de passe</label>
				<span id="erreur_mot_de_passe"></span>
				<input type="password" name="mot_de_passe" id="mot_de_passe" value=""/>

				<input type="submit" value="S'identifier" />
				<input type="reset" value="raz" />
			</fieldset>
		</form>
	</body>
</html>
```

Fichier "traitement.php"

```
<!DOCTYPE html>
<html>
	<head>
		<meta charset="utf-8">
		<title>Formulaire de saisie</title>
	</head>
	<body>
	<?php
		//recupération des données provenant du formulaire de saisie
		$identifiant=$_POST['identifiant'];
		$mot_de_passe=$_POST['mot_de_passe'];
		echo "	<div id=\"resultats\"><p>Voici les données transmises par le formulaire</p>
			<p>	Identifiant : <span id=\"resultat_identifiant\">".$identifiant."</span></p>
			<p>	Mot de passe : <span id=\"resultat_mot_de_passe\">".$mot_de_passe."</span></p>
		</div>";
	?>
	</body>
</html>
```

5.1.2.Exercice 2 - formulaire avec fieldset et css => traitement :

Voici ce qu'il faudra obtenir :



Complétez le formulaire précédent en :

* ***ajoutant dans la balise formulaire un fieldset.***

Un fieldset est un ensemble de champs possédant une légende, cela permet de regrouper les champs dans un cadre.

Utilisez pour cela la balise fieldset et la balise legend.

Ecrire ici votre code html "formulaire.html" :

```
<!DOCTYPE html>
<html>

	<head>
		<meta charset="utf-8">
		<title>Formulaire de saisie</title>

		<!--  style de notre document html -->
		<link rel="stylesheet" type="text/css" href="./css/exercice1.css" media="all"/>
```

```
	</head>
	<body>
		<form action="./traitement.php" method="post" id="monForm">
			<fieldset>
				<legend>
					Authentification
				</legend>
			  <label for="identifiant">Identifiant</label>
				<input type="text" name="identifiant" id="identifiant" value="" />
				<span id="erreur_identifiant"></span>

				<label for="mot_de_passe">Mot de passe</label>
				<span id="erreur_mot_de_passe"></span>
				<input type="password" name="mot_de_passe" id="mot_de_passe" value=""/>

				<input type="submit" value="S'identifier" />
				<input type="reset" value="raz" />
			</fieldset>
		</form>
	</body>
</html>
```

**** ajoutant le CSS suivant :***

```
#monForm
{
	width: 60%;
	-moz-box-shadow: 10px 10px 5px 0px #ce6301;
	-webkit-box-shadow: 10px 10px 5px 0px #ce6301;
	-o-box-shadow: 10px 10px 5px 0px #ce6301;
	box-shadow: 10px 10px 5px 0px #ce6301;
	-moz-border-radius: 10px;
	-webkit-border-radius: 10px;
	border-radius: 10px;
}
#monForm p {	margin: 2px 0;}

#monForm fieldset
{
	margin-bottom: 10px;
	border: #CCC 1px solid;

	-moz-border-radius: 10px;
	-webkit-border-radius: 10px;
	border-radius: 10px;
```

```
}
#monForm fieldset:hover{  background-color: #FFF;}

#monForm fieldset legend{
	padding: 0 10px;
	border-left: #CCC 1px solid;
	border-right: #CCC 1px solid;
	font-size: 1.2em;
	color: #999;
}
/* Label */
#monForm label
{
	background-color: #FFCC66;
	display: block;
	width: 39%;
	float: left;
	padding-right: 1%;
	text-align: right;
	letter-spacing: 1px;
}
#monForm label:hover{	font-weight: bold;}

#monForm .form_label_nostyle{	background: none;}
/* Input */
#monForm input, #monForm select
{
	margin-left: 1%;
	width: 58%;
	border: #CCC 1px solid;
}
#monForm input:hover, #monForm select:hover, #monForm input:focus, #monForm select:focus
{
	border: #999 1px solid;
	background-color: #DDEEFF;
}
/* button submit */
#monForm input[type="submit"]
{
	border: #DDEEFF 1px solid;
	width: 27%;
}

#monForm input[type="submit"]:hover
{
	background-color: #66CC33;
	cursor: pointer;
}

#monForm input[type="reset"]
{
	border: #DDEEFF 1px solid;
	width: 27%;
}

#monForm input[type="reset"]:hover
{
	background-color: #E6484D;
	cursor: pointer;
}
```

5.2.Protéger les données transmises pas un formulaire (cf http://fr.php.net/manual/fr/function.strip-tags.php, http://php.net/manual/fr/function.htmlspecialchars.php).

La faille XSS, d'où son véritable nom CSS (Cross Site Scripting) modifié pour ne pas confondre avec le CSS des feuilles de style (Cascading Style Sheet), est en fait une injection de code javascript dans des données lorsque le HTML n'est pas désactivé.

Afin d'éviter les risques d'injections de code lors de la transmission de données entre un formulaire et le script php, vous allez :
- supprimer toutes les éventuelles balises,
- "échapper" dans vos scripts php toutes les données provenant du formulaire de saisie.

5.2.1.Supprimer les éventuelles balises HTML.

C'est la fonction strip_tags qui permet de supprimer les balises html d'une chaine de caractères.

SYNTAXE:

chaineSansBalise strip_tags(**chaine**)

chaine : chaine de caractères originale.
chaineSansBalise : chaine dans laquelle toutes les balises ont été supprimées.

5.2.2.Echapper une donnée.

Echapper une donnée c'est ajouter un anti-slash devant tous les caractères spéciaux.

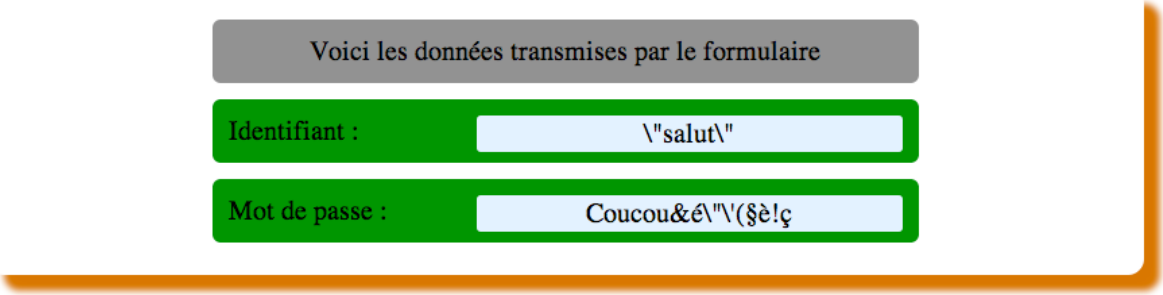
SYNTAXE :

chaineEchappée htmlspecialchars(**chainedecaractères**)

chaineEchappée : chainedecaractères dont les caractères spéciaux ont été echappés.
chainedecaractères : chaine de caractères originale.

5.2.3.Exercice 3.

Complétez le fichier traitement php afin de :
- supprimer les tags éventuels,
- échapper les " et les ', ***l'affichage obtenu*** devra ressembler à :



Code php "traitement.php" :

```php
<!DOCTYPE html>
<html>

	<head>
		<meta charset="utf-8">
		<title>Formulaire de saisie</title>

		<!--  style de notre document html -->
		<link rel="stylesheet" type="text/css" href="./css/exercice1.css" media="all"/>

	</head>
	<body>
	<?php
		//recupération des données provenant du formulaire de saisie
		$identifiant=strip_tags($_POST['identifiant']);
		$identifiant=htmlspecialchars($identifiant) ;

		$mot_de_passe=strip_tags($_POST['mot_de_passe']);
		$mot_de_passe=htmlspecialchars($mot_de_passe );

		echo "	<div id=\"resultats\"><p>Voici les données transmises par le formulaire</p>
			<p>	Identifiant : <span id=\"resultat_identifiant\">".$identifiant."</span></p>
			<p>	Mot de passe : <span id=\"resultat_mot_de_passe\">".$mot_de_passe."</span></p>
		</div>";

	?>

	</body>
</html>
```

CSS associé :

```css
#resultats{
	width: 60%;
	-moz-box-shadow: 10px 10px 5px 0px #ce6301;
	-webkit-box-shadow: 10px 10px 5px 0px #ce6301;
	-o-box-shadow: 10px 10px 5px 0px #ce6301;
	box-shadow: 10px 10px 5px 0px #ce6301;

	-moz-border-radius: 10px;
	-webkit-border-radius: 10px;
	border-radius: 10px;
	padding: 10px;
}
#resultats p{
	background-color: #008800;
	width:400px;
	margin:10px auto 10px auto;
	padding:10px;

	-moz-border-radius: 5px;
	-webkit-border-radius: 5px;
	border-radius: 5px;
}
#resultats>p:FIRST-CHILD{
	text-align: center;
```

```
        background-color: grey;
}
#resultat_identifiant, #resultat_mot_de_passe{
        background-color: #DDEEFF;
        float:right;
        padding:2px;;
        margin:0px;
        width:250px;

        -moz-border-radius: 3px;
        -webkit-border-radius: 3px;
        border-radius: 3px;
        text-align: center;
        min-height:18px;
}
```

6. Variables superglobales

Ce sont des variables que php génère automatiquement, elles sont écrites en majuscules et de type array.

Variable	Description
$_SERVER	Ce sont des valeurs renvoyées par le serveur. Elles sont nombreuses et quelques-unes d'entre elles peuvent nous être d'une grande utilité. Je vous propose de retenir au moins `$_SERVER['REMOTE_ADDR'].` Elle nous donne l'adresse IP du client qui a demandé à voir la page. Cela peut être utile pour l'identifier.
$_ENV	Ce sont des variables d'environnement, toujours données par le serveur.
$_SESSION	On y retrouve les variables de session. Ce sont des variables qui restent stockées sur le serveur le temps de la visite d'un visiteur. Nous allons apprendre à nous en servir dans la partie suivant.
$_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 par exemple pour se souvenir de son nom.
$_GET	vous la connaissez, elle contient les données envoyées en paramètre dans l'URL

$_POST	De même, c'est une variable que vous connaissez qui contient les informations qui viennent d'être envoyées par un formulaire.
$_REQUEST	De même, c'est une variable qui permet de récupérer les informations envoyées par un formulaire, peu importe cette fois la méthode employée.
$_FILES	Elle contient la liste des fichiers qui ont été envoyés via le formulaire précédent.

7. Les sessions

7.1.C'est quoi?

Vous avez vu que vous pouviez passer des paramètres d'une page web à une autre en utilisant par des champs de formulaires et les superglobales $_GET ou $_POST ou $_REQUEST.

Lorsque vous désirez transmettre une information d'une page contenant du code php à une autre page, une solution est de passer par des champs cachés, mais le code n'est pas "élégant", la solution est d'utiliser les sessions.

Une session permet de stocker lorsque vous le désirez des informations dans la superglobale $_SESSION.

7.2.Ce qu'il faut respecter à tout prix !

Pour pouvoir utiliser les sessions dans votre site web, vous devrez sur **chaque page** écrire l'instruction php ***session_start(); avant tout envoi de code HTML au navigateur***.

Exemples :

==> Correct

```
<?php
	session_start();

	/**
	 * affichage du formulaire d'identification
	 */

	?>

<!DOCTYPE html>
<html>

	<head>
```

==> Pas correct:

```
<!DOCTYPE html>
<html>
<?php
        session_start();

        /**
         * affichage du formulaire d'identification
         */

        ?>
        <head>
```

7.3."id" de session.

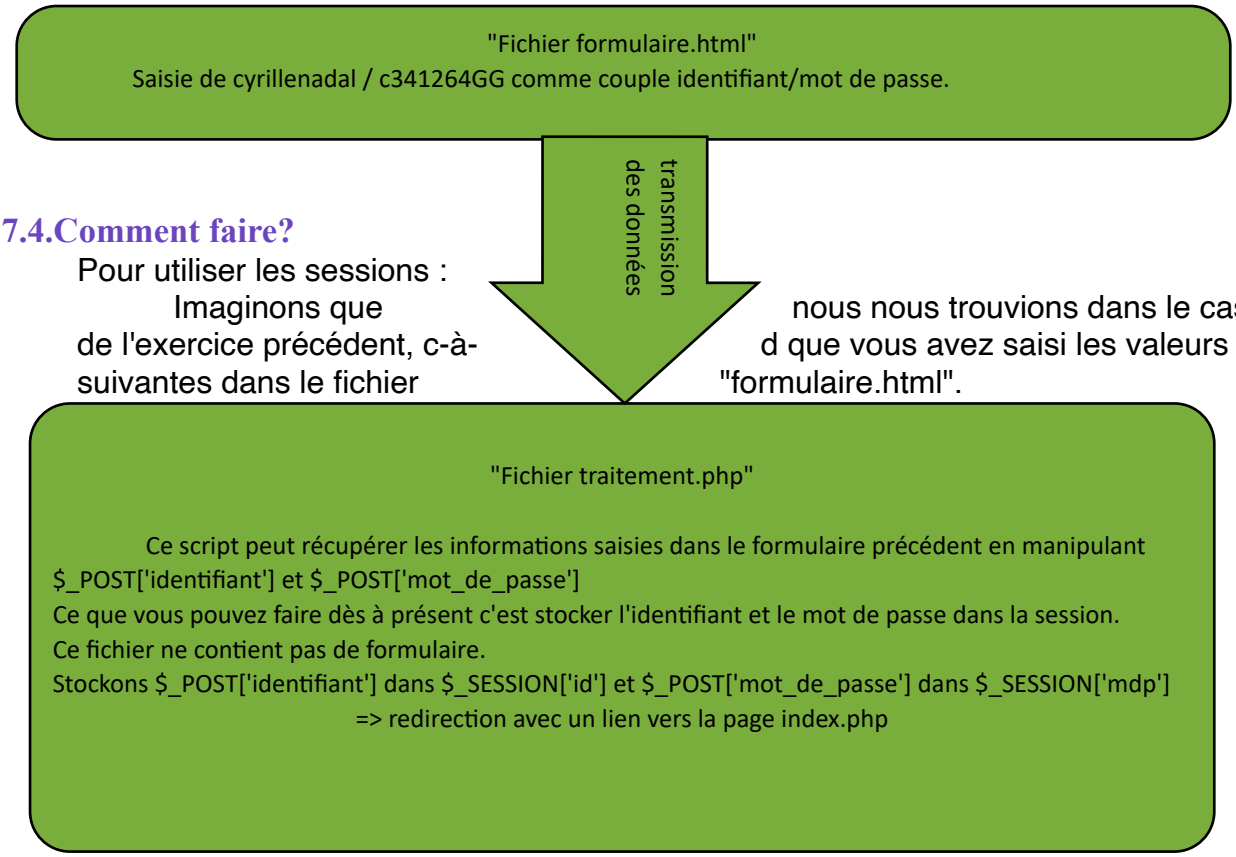
Lorsque vous créez une session, le serveur vous attribue, pour une durée donnée, un numéro (id) de session.
Si vous désirez afficher l'id de session, utilisez :

```
echo "Voici mon id de session : ".session_id();
```

7.4.Comment faire?

Pour utiliser les sessions :

Imaginons que nous nous trouvions dans le cas de l'exercice précédent, c-à-d que vous avez saisi les valeurs suivantes dans le fichier "formulaire.html".



"Fichier index.php"

Ce script va récupérer les informations dans la session et les afficher.

Voilà, la gestion des sessions c'est simple !

Voici le code associé :

le fichier "formulaire.html" contient le code de l'exercice précédent.
le fichier "traitement.php" contient le code suivant :

```
<?php
session_start();
?>
.... code identique à l'exercice précédent....

<?php
//stockage des informations dans la session
        $_SESSION['id']=$identifiant;
        $_SESSION['mdp']=$mot_de_passe;
?>

<a href="./index.php">retour à l'index</a>
... code identique pour la fin ....
```

le fichier index.php contient le code suivant :

```
<?php
session_start();

.... balises du squelette d'une page web html5 ....

echo "<p>Voici les informations reçues par ce script : ".$_SESSION['id']."</p>";
echo "<p>Voici le mot de passe crypté en MD5 : ".md5($_SESSION['mdp'])."</p>";

?>

.... fermeture correcte des balises ....
```

7.5.Exercice 4.

Codez ces trois fichiers.
Indiquez les difficultés rencontrés.

7.6.Traitements sur les sessions

7.6.1.Existence d'une session

Pour tester l'existence d'une session, il faut utiliser empty($_SESSION). empty est une fonction qui retourne true lorsque la variable passée en paramètre est vide, false sinon.

```
<?php
        if (empty($_SESSION)){
                echo "<p>Erreur, session non ouverte</p>";
        else{
                echo "<p>Voici l'id : ".$_SESSION['id']."</p>";
                echo "<p>Voici le mot de passe : ".md5($_SESSION['mdp'])."</p>";
        }

?>
```

7.6.2.Destruction d'une session

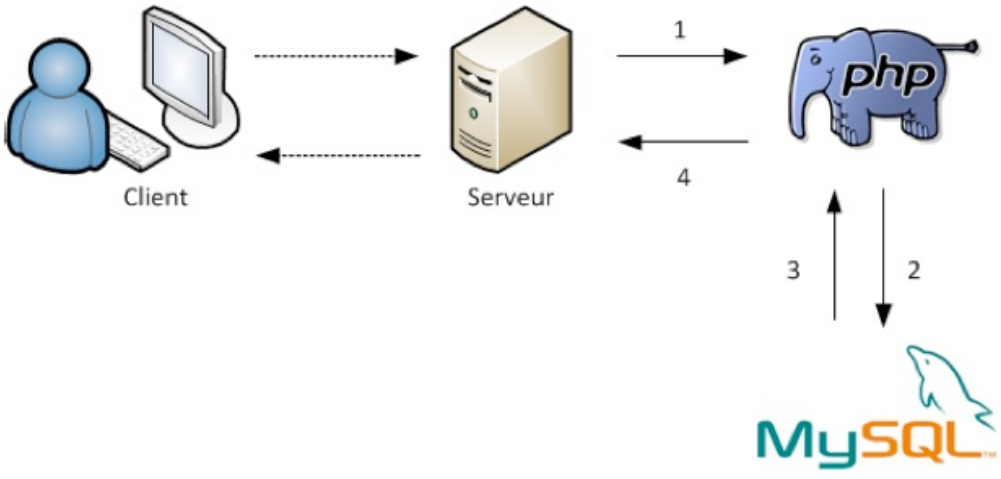
Pour terminer une session, il faut utiliser la fonction "session_destroy()".
Pour faire cela, vous pouvez créer un lien pour se déconnecter, lorsque vous cliquez sur ce lien :
- destruction de la session avec session_destroy();
- redirection de la page vers la pages précédente avec header("Location:index.php").

7.6.3.Exercice 5

Complétez les fichiers créés dans l'exercice 4 en ajoutant dans index.php :
- le test d'existence de la session
- un lien vers un fichier php permettant de se déconnecter (destruction de la session).

8. Manipuler des informations stockées dans une base de données.

Voici un schéma tiré du site du zéro qui représente la circulation de l'information entre le navigateur client et le serveur web.



Expliquez ce schéma en donnant une phrase par point suivant :

1 :

```
2 :
3 :
4 :
```

8.1.Une base de données.

Une base de données est composée de tables, pour pouvoir vous connecter, vous devez posséder un couple login/mot de passe.

Vous allez travailler sur le SGBDR postgres, le serveur est :

postgres.sjperse.fr

Vous avez déjà un compte sur ce serveur, de plus, vous y avez créé une table nommée utilisateur.

8.2.Langage : sql

Le langage utilisé pour utiliser la base de données est sql.
Vous savez manipuler sql pour interroger, insérer, modifier des données issues d'une table.

8.3.Accès à la base de données : requête de sélection

Vous allez utiliser une classe d'accès aux données pour exécuter une requête de sélection.

Cette classe se nomme **gestionUtilisateur**, elle se trouve en **annexe** de ce polycopié.
Grâce à cette classe, vous pourrez vous connecter au serveur, exécuter une requête SQL, récupérer les lignes provenant de la requête et les traiter.

Voici un **exemple de code php qui se connecte**

à un sgbdr postgres avec un login/mot de passe à la base de données "test"

puis exécute une requête sur la table utilisateur.

Ensuite, il **parcourt** chaque ligne pour **afficher** le **champ** "**prenom"**.

```
<?php
include_once './gestionUtilisateur.php';

try {
	//creation de l'acces à la bd nommée test => votre bd aura un nom différent !!!!
	//                                          login      motdepasse  basededonnees
	$connexion=new gestionUtilisateur("xxxxxxx",   "xxxxxxx!",         "test");

	//exécution de la requête et affectations de toutes les lignes dans $lesLignes
	$lesLignes=$connexion->requeteSelection("select * from public.utilisateur");

	//tant que l'on peut extraire une ligne depuis lesLignes
	//uneLigne reçoit à chaque fois une ligne différente, (la première puis la seconde, etc...)
	while($uneLigne=$lesLignes->fetch(PDO::FETCH_OBJ)){
		echo "<p>".$uneLigne->prenom.'</p>';
	}
```

```
} catch (PDOException $e) {
        echo $e->getMessage();
}
```

8.4.Exercice 6

Testez le code ci-dessus en l'appliquant à votre table utilisateur.
ATTTTTention ! : je vous rappelle que votre table utilisateur se trouve dans le schéma portant votre nom, donc la requête sera par exemple :
"select * from gomez.utilisateur".

Structure de la table utilisateur.

	identifiant [PK] characte	mot_de_pass character var	nom character var	prenom character var	adresse character var	numero_de_t character var	courriel character var	datedenaissa character var
1	12	siocoladr	colin	adrien				
2	13	siosaiale	saintorens	alexandre				
3	14	siolapaud	laporte	audrey				

8.5.Accès à la base de données : requête d'action

Une requête d'action permet d'utiliser les instructions sql suivantes :
insert / delete / update / grant / revoke / create table / drop table, etc....

Voici un exemple de code qui insère une ligne dans la table utilisateur.

```
try {


        //creation de l'acces à la bd nommée test ====> votre bd aura un nom différent !!!!
        $connexion2=new gestionUtilisateur("xxxxxxx", "xxxxxxxx", "test");

        //exécution de la requête et affectations de toutes les lignes dans $lesLignes
        $nbLignes=$connexion2->requeteAction("insert into public.utilisateur
values ('','monMotPasse','nadal','cyr','12 rue' , '1212121212' , 'nc@free.fr' , '12/12/1912')");

        echo "<p>La requête à insérée $nbLignes ligne(s)</p>";

} catch (PDOException $e) {
        echo $e->getMessage();
}
```

8.6.Exercice 7

Testez le code ci-dessus, après avoir inséré la ligne, exécutez le code de l'exercice 6 pour apprécier les changements (ajout de la ligne).

9. TP à faire en binôme.

Votre binôme devra être composé d'un étudiant slam et d'un étudiant sisr. Le travail sera à rendre pour le .../..../..... au plus tard.

Ce qu'il faut faire :

Créer un site web permettant de gérer des utilisateurs, sur le thème de ce que l'on à vu au cours du premier semestre en SI4.

Voici les fonctionnalités attendues :

- inscription d'un utilisateur,
- authentication d'un utilisateur => création d'une session => affichage des informations de l'utilisateur connecté.
- SI l'utilisateur est authentifié :
  - menu permettant de
    - **afficher** ses informations personnelles,
    - **modifier** ses informations personnelles,
    - **supprimer** l'utilisateur et de **fermer** la session
    - **fermer** la session

Annexes

fichier "gestionUtilisateur.php"

```
<?php

/**
 * @author ncyr
 *
 *
 */
class gestionUtilisateur {
	//attributs / proprits
	public $connexion;
	public $donnees;

	//mthodes
	//constructeur.
	function __construct($nomUtilisateur,$motDePasse,$baseDeDonnees) {
		//connexion  la base de donnes

	$serveur = "pgsql:dbname=$baseDeDonnees;host=postgres.sjperse.fr;port=5432";

		//connexion en utilisant la classe d'accs aux donnes : PDO

		try {

		   $this->connexion = new PDO($serveur, $nomUtilisateur, $motDePasse);
		   $this->connexion->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);

		}
		catch (PDOException $e) {
		   echo 'Echec lors de la connexion : ' . $e->getMessage();
			$this->connexion=null;
		}
	}
	public function connexionActive(){
```

```
            return $this->connexion!=null;
            //equivalent  :
            // if($this->connexion!=null) return true;
            // else return false;
        }
        function requeteSelection($requete){
            try {
                if($this->connexionActive()==false){
                    return null;
                }else{
                    $this->donnees=$this->connexion->query($requete);
                    return $this->donnees;
                }

            } catch (PDOException $e) {
                echo "Echec lors de l'execution de la requete : " . $e->getMessage();
                return null;
            }
        }

        function requeteAction($requete){
            try {
                if($this->connexionActive()==false){
                    return null;
                }else{
                    $this->donnees=$this->connexion->exec($requete);
                    return $this->donnees;
                }

            } catch (PDOException $e) {
                echo "Echec lors de l'execution de la requete : " . $e->getMessage();
                return null;
            }
        }
}

?>
```

Gérer les évènements "clients" : le langage javascript.

Le langage javascript est un langage qui s'exécute sur l'ordinateur de l'utilisateur qui souhaite visiter un site web, c-à-d dans un navigateur internet.

1. La balise script

Le code javascript étant affiché par un navigateur doit être inclus dans des pages web. Cela se fait en ajoutant une balise script dans une page html.

Voici la syntaxe à utiliser lorsque votre code est écrit dans votre page html.

```
<script type="text/javascript">
        // code javascript
</script>
```

Voici la syntaxe à utiliser lorsque votre code javascript est stocké dans le fichier "./scripts/verifFormulaire.js".

```
<script type="text/javascript" src="./scripts/verifFormulaire.js">

</script>
```

2. Que faire avec javascript ?

Javascript va vous permettre de gérer les interactions entre l'utilisateur et la page web que vous avez créé.

Par exemple,
- lorsqu'un champ de votre formulaire recevra le focus, le fond du champ sera colorié en "jaune"
- lorsque le formulaire sera validé vous pourrez vérifier que les champs contiennent une valeur correcte, afficher un ou des messages d'erreurs et bien sur annuler la validation du formulaire.

3. Quelle syntaxe pour les instructions de base ?

La syntaxe est presque identique au php/c/c++/java et donc vous connaissez les bases de ce langage.

Attention le javascript est sensible à la casse !!!!!

3.1.Les variables

Les variables sont créées à la volée (dynamiquement) comme en php à ceci près qu'une variable ne doit pas commencer par un chiffre ou correspondre à un mot clé de javascript.

Syntaxe :

```
var i;
var chien, _chat;
i=3.0;
chien = i * 9;
```

Les variables ne sont pas typées.

3.2.Structures conditionnelles (if, switch).

Identique à la syntaxe php.

3.3.Structures itératives (for, while, do).

Identique au php.

3.4.Test du type d'une variable.

Vous pouvez utiliser typeof(variable) pour déterminer si une variable est de type string, number, boolean ou autre....

Exemple :

```
<script type="text/javascript">
var mot;
mot=prompt("Veuillez saisir un mot");
alert(typeof(mot));
if (typeof(mot) != "string"){
        alert("Mauvaise saisie!!!");
}
mot=3.0;
alert(typeof(mot));
if (typeof(mot) != "string"){
        alert("Mauvaise saisie!!!");
}
</script>
```

3.5.Chaine de caractères

Le type est string, l'opérateur de concaténation est "+".

3.6.Saisie de données depuis une boite de message.

Utiliser prompt(message).
prompt retourne toujours une chaine de caractères.

Exemple :

```
var mot;
mot=prompt("Veuillez saisir un mot");
```

3.7.Affichage de données dans une boite de message.

Exemple :

```
var mot;
mot=prompt("Veuillez saisir un mot");
alert(mot);
```

3.8.Changer le type d'une variable.

Utiliser parseType(variable) pour modifier la variable en changeant son type.

soit :

parseInt(variable)
parseFloat(variable)
String(variable)
number(variable)

Exemple :

```
var mot;
mot=5;
mot=String(mot);
alert(typeof(mot));
```

3.9.Les tableaux.

3.9.1. Les tableaux ordonnées.

Un tableau commence à l'indice 0.

Voici la syntaxe à utiliser :

Pour déclarer un tableau

```
var montab=new Array("voiture",3,"Bonjour");
var montab2= ["toto","lolo",3,5];
```

Pour ajouter un element

```
montab[3]="33";
montab2.push("lili");
montab.push("lolo");
```

Pour afficher un élément

```
alert(montab[1]);
alert(montab2[3]);
```

Pour supprimer le premier élément du tableau : shift()

```
montab.shift();
alert(montab);
```

Pour supprimer le dernier élément du tableau : pop()

```
montab2.pop();
alert(montab2);
```

3.9.2. Les tableaux associatifs.

La syntaxe est un peu différente ici par rapport à php.

Pour créer un tableau associatif

```
var monchien3={
        nom : "minet",
        prenom : "jaune"
        }
```

Pour accéder à un élément

```
alert(monchien3.nom);
```

Pour ajouter un élément

```
monchien3["age"]=49;
```

3.9.3. Parcours.

Pour un tableau ordonné (on connait le nombre d'éléments) :

```
for (var i = 0;i<montab2.length;i++){
        alert("Valeur de la case "+ i + " : " +montab2[i]);
}
```

Pour un tableau associatif (on ne connait pas le nombre d'éléments) :

```
for (var cle in monchien3){
        alert(cle);  //equivalent à monchien3[i]
}
```

4. Les sous-programmes.

Un sous-programme va contenir des instructions qui se répètent dans votre code.

Les sous-programmes sont des fonctions qui peuvent retourner une valeur (simple : integer, string, ... ou composée : tableau, objet, ...).

Syntaxe :

```
function nomFonction([ argument(s) ]){

        [return valeur;]
}
```

Exemple : Cet exemple intercepte l'événement focus sur les champs du formulaire.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Saisie</title>
</head>
<body>

        <form action="./traitement.php">
                <label for="identifiant">Identifiant : </label><input type="text" value="Saisir un identifiant"
id="identifiant" name="identifiant" onclick="champFocus(this);"/>
                <br />
                <label for="mdp">Mot de passe : </label><input type="text" value="" id="mdp" name="mdp"
onclick="champFocus(this);"/>
                <br />
                <input type="submit" onclick="" />
        </form>

<script type="text/javascript">

function champFocus(leChamp){
        // leChamp represente un champ de saisie d'un formulaire
        if (leChamp.value != "")
                leChamp.value="";

        leChamp.style.backgroundColor="yellow";
}

</script>
</body>
</html>
```

La fonction champFocus prend en paramètre un "lien" vers un champ de saisie, elle efface sa valeur (attribut value) à "" et colorie ensuite son fond en jaune.

Exercice :

Testez l'exemple précédent, analysez-le et complétez-le en :

- ajoutant deux zones de saisie pour l'adresse et le courriel,
- faisant les mêmes traitements sur le focus pour ces deux nouveaux champs,
- ajoutant la fonction champBlur qui prend en paramètre "leChamp" et qui colorie le fond en vert si la valeur de leChamp est <> "" en rouge sinon.
- Sachant que Blur signifie "perdre le focus", rajoutez onblur="champBlur(this);" sur tous les input du formulaire (sauf le bouton).

Corrigé :

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Saisie</title>
</head>
<body>
	<form action="./traitement.php">
		<label for="identifiant">Identifiant : </label><input type="text" value="Saisir un identifiant"
id="identifiant" name="identifiant" onclick="champFocus(this);" onblur="champBlur(this);"/>
		<br />
		<label for="mdp">Mot de passe : </label><input type="text" value="" id="mdp" name="mdp"
onclick="champFocus(this);" onblur="champBlur(this);"/>
		<br />
		<label for="adresse">Adresse : </label><input type="text" value="" id="adresse" name="adresse"
onclick="champFocus(this);" onblur="champBlur(this);" />
		<br />
		<label for="courriel">Courriel : </label><input type="text" value="" id="courriel" name="courriel"
onclick="champFocus(this);" onblur="champBlur(this);"/>
		<br />
		<input type="submit" onclick="" />
	</form>

<script type="text/javascript">

function champFocus(leChamp){
	// leChamp represente un champ de saisie d'un formulaire
	if (leChamp.value != "")
		leChamp.value="";

	leChamp.style.backgroundColor="yellow";
}

function champBlur(leChamp){
	if (leChamp.value != "")
		leChamp.style.backgroundColor="green";
	else
		leChamp.style.backgroundColor="red";
}
</script>
</body>
</html>
```

5. Manipuler le document html : DOM (document object model).

DOM contient une API (interface de programmation : ensemble d'outils) permettant de manipuler les documents écrits en XML et HTML.

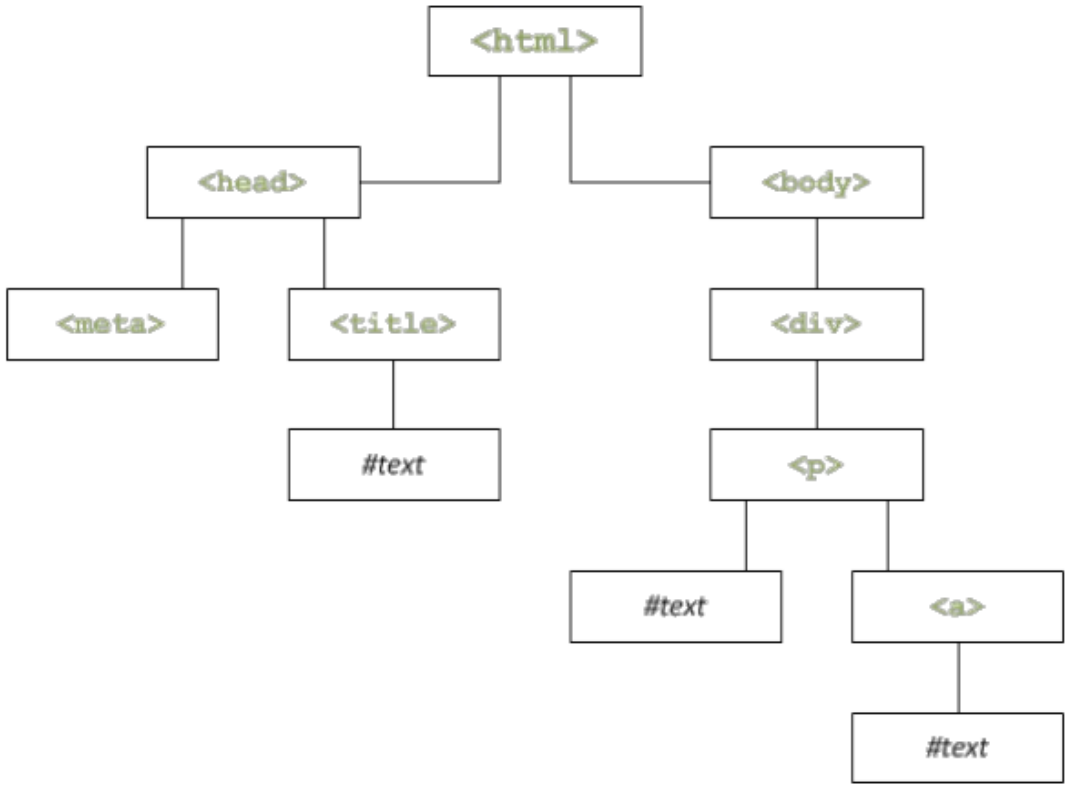
Un document XML ou HTML est en fait un arbre, il possède une racine et un ensemble de noeuds pour se terminer par des feuilles.

DOM vous permettra :

- ➡d'ajouter des éléments dynamiquement (images, zones de textes, ...),
- ➡de modifier des éléments dynamiquement (changement d'image, de texte, etc...),
- ➡de supprimer des éléments dynamiquement (suppression d'un formulaire et ajout d'un texte de récapitulatif, etc...).

5.1.Le document

On peut représenter le document html de la façon suivante :



Le code correspondant est :

```
<!doctype html>
<html>
<head>
        <meta charset="utf-8" />
        <title>Le titre de la page</title>
</head>
<body>
        <div>
                <p>Un peu de texte <a>et un lien</a></p>
        </div>
</body>
</html>
```

5.1.1. Comment sélectionner des éléments dans le document ?

Ces méthodes seront appliquées sur document.

➡ getElementById(unId)

Permet de sélectionner ***le noeud dont l'id est passé en paramètre***.

Par exemple :

Code source

```
<!doctype html>
<html>
<head>
	<meta charset="utf-8" />
	<title>Le titre de la page</title>
</head>
<body>
	<div>
		<p id="texte">Un peu de texte <a>et un lien</a></p>
		<input type="button" onclick="modifTexte();">
	</div>
</body>
</html>
```

Sélection avec getElementById /Modification avec innerHTML

```
<script type="text/javascript">
	function modifTexte(){
		var leTexte=document.getElementById("texte");
		alert(leTexte.innerHTML);
		leTexte.innerHTML="valeur modifiée....";
	}
</script>
```

➡ getElement**s**ByTagName(uneBalise)

Permet de sélectionner ***les noeuds dont le nom de balise est égal au nom passé en paramètre***.

Testez ce code, complétez le avec du code html, que fait-il?

```
var lesP = document.getElementsByTagName('p');
for (var i = 0 ; i < lesP.length ; i++) {
	lesP[i].style.color="red";
}
```

➡ getElement**s**ByName(unNom)

Permet de sélectionner ***les noeuds dont la valeur de l'attribut name est égal à la valeur passée en paramètre***.

Testez ce code, complétez le avec du code html, que fait-il?

```
var lesP = document.getElementsByName('formations');
for (var i = 0 ; i < lesP.length ; i++) {
	lesP[i].style.color="red";
}
```

➡ querySelector(chaineDeSelectionCSS)
Permet de récupérer le premier noeud correspondant à la sélection.
querySelector("#menu .item span"); permettra de récupérer le noeud

➡ querySelectorAll(chaineDeSelectionCSS)

Exemple : testez ces codes.
code html

```
<div id="menu">
        <div class="item">
                <span>Élément 1</span>
                <span>Élément 2</span>
        </div>
        <div class="publicite">
                <span>Élément 3</span>
                <span>Élément 4</span>
        </div>
</div>
<div id="contenu">
        <span>Introduction au contenu de la page...</span>
</div>
```

code javascript :

```
var query = document.querySelector('#menu .item span'),
queryAll = document.querySelectorAll('#menu .item span');
alert(query.innerHTML); // Affiche : "Élément 1"
alert(queryAll.length); // Affiche : "2"
alert(queryAll[0].innerHTML + ' - ' + queryAll[1].innerHTML); //
Affiche : "Élément 1 - Élément 2"
```

5.1.2. Comment lire/modifier un attribut ?

Il faut être sur un élément du document et lui appliquer ensuite une de ces deux méthodes :

➡ getAttribute(chaineDeSelectionCSS)
➡ setAttribute(chaineDeSelectionCSS)

Exemple :

```
<input type="text" value="" id="leNom" />
<input type="button" onclick="modif();" />


<script type="text/javascript">
        function modif(){
                var nom=document.querySelector("#leNom");
                nom.setAttribute("type","submit");
        }
</script>
```

Que réalise ce script ?

5.1.3. Accéder à la valeur d'un élément et la modifier

innerHTML : innerHTML appliqué à un élément du document vous permettra de lire/modifier le texte de cet élément.

Prenons l'exemple de

```
<div id="erreur"></div>
```

En appliquant le code suivant :

```
document.getElementById("#erreur").innerHTML="bonjour";
```

Le code source du document affiché dans le navigateur deviendra dynamiquement :

```
<div id="erreur">bonjour</div>
```

et l'affichage sera ainsi modifié !

Rajoutons le code suivant pour afficher la valeur text de l'élément dont l'id = "erreur".

```
alert(document.getElementById("#erreur").innerHTML);
```

5.2.Les événements

La gestion des événements est très importante pour tout ce qui concerne l'affichage de pages dans un navigateur web.

Voici les principaux événements survenant sur des éléments HTML :
(tableaux tirés du site du zéro).

Nom de l'événement	Action pour le déclencher
click	Cliquer (appuyer puis relâcher) sur l'élément
dblclick	Double-cliquer sur l'élément
mouseover	Faire entrer le curseur sur l'élément
mouseout	Faire sortir le curseur de l'élément
mousedown	Appuyer (sans relâcher) sur le bouton gauche de la souris sur l'élément
mouseup	Relâcher le bouton gauche de la souris sur l'élément
mousemove	Faire déplacer le curseur sur l'élément
keydown	Appuyer (sans relâcher) sur une touche clavier sur l'élément
keyup	Relâcher une touche clavier sur l'élément
keypress	Frapper (appuyer puis relâcher) sur une touche clavier sur l'élément
focus	"Cibler" l'élément
blur	Annuler le "ciblage" de l'élément
change	Changer la valeur d'un élément spécifique aux formulaires (input, checkbox, etc...)
select	Sélectionner le contenu d'un champ de texte (input, textarea, etc...)

Ceux survenant sur un formulaire :

Nom de l'événement	Action pour le déclencher
submit	Envoyer le formulaire
reset	Réinitialiser le formulaire

5.2.1. Comment intercepter un événement utilisateur ?

il suffit de choisir l'événement que l'on désire intercepter et de le faire précéder de "on", "onclick" par exemple.

5.2.2. Gestion du premier événement

Voyons l'exemple suivant :

```
<form>
        <input type="text" value="Votre nom ?" onfocus="this.value='';" />
</form>
```

this : représente ici l'élément html sur lequel survient l'événement.

On peut aussi déporter le code javascript this.value='' dans une fonction javascript.

```
<form>
	<input type="text" value="Votre nom ?" onfocus="saisie(this);" />
</form>
<script type="texte/javascript">
	function saisie(unElement){
		unElement.value='';
	}
</script>
```

5.2.3. Exercice.

Reprendre ce code en ajoutant la modification de la couleur de fond de l'élément qui reçoit le focus en jaune.

Pour modifier le style de l'élément dont l'id est toto : (attention il y a un piège)

```
document.getElementById("toto").style.backgroundColor="jaune";
```

5.2.4. Annuler l'action d'une balise.

Prenons l'exemple d'un formulaire qui contient des champs de saisie, vous pouvez modifier le bouton de soumission du formulaire en ajoutant :

```
<form action="traitement.php">
	<input type="text" value="Votre nom ?" onfocus="saisie(this);" />
	<input type="submit" value="valider" onclick="alert('validation annulée!');return false;" />
</form>
```

Utilité de return false ?

```
bloque l'action de la balise, ici la redirection de page.
```

5.3.Cas pratique : formulaire html valide.

Vous allez compléter le formulaire de saisie que vous avez créé dans le TP de la page 24/25.

5.3.1. Les champs et les erreurs.

Chaque champs de saisie de formulaire sera suivi d'un champ caché vide (div ou span à vous de choisir)

5.3.2. Le focus sur les champs

lorsque un champ du formulaire reçoit le focus, le fond de ce champ est colorié en jaune.

lorsque le champ du formulaire perd le focus (événement blur), si le champ contient "" le fond est colorié en rouge et un message d'erreur apparaît dans le champ caché associé, en vert sinon et le message d'erreur associé disparait.

5.3.3. Soumission du formulaire

Le formulaire est soumis si tous les champs sont saisis, sinon un message d'erreur apparait (avec une instruction alert).

5.3.4. Complément

En plus de vérifier si les champs sont renseignés, vous allez vérifier leur contenu :

pour l'identifiant : au moins 6 caractères,
pour le mot de passe : concordance des contenus,
pour le courriel : vérification de la présence d'1 seul "@",
pour le téléphone : valeur numérique et 10 chiffres

Le formulaire ne sera validé que si toutes ces conditions sont remplies.

6. Jquery : améliorer l'ergonomie de vos pages.

Jquery est une bibliothèque de fonctions développée en **javascript**, on utilise aussi le terme de framework ou aussi IPA "Interface de Programmation Applicative" ou API "Application Programming Interface".

6.1.Inclure la bibliothèque.

La bibliothèque Jquery est téléchargeable sur le site www.jquery.com.
Télécharger la version "production" et enregistrez ce fichier sous "**jquery.js**".

Dès maintenant, vous pourrez utiliser Jquery dans vos pages web en ajoutant simplement le contenu du script javascript téléchargé. Cela se fait en utilisant la balise script dont la source sera le fichier jquery.js téléchargé.

```
<!-- par exemple :
        on inclut le fichier jquery.js se trouvant dans le dossier js/
        Si ce fichier est absent, affichage du message "Fichier API non trouvé".
-->
<script src="./js/jquery.js">
        alert("Fichier API non trouvé");
</script>
```

6.2.Document prêt.

Les codes javascript exploitant les fonctionnalités de **Jquery** seront placées dans des fichiers séparés :

```
<!-- fichier html principal -->
<head>
        <script src="./js/jquery.js">
        </script>
        <script src="./js/monscript.js">
        </script>
        <!-- suite du head -->
</head> .
```

Code de votre script : "monscript.js".

```
//fichier javascript exploitant jquery
//jquery est déjà inséré dans le document principal.

$(document).ready(function() {
          // Handler for .ready() called.
        $("*").click(function(){
                alert('le clic est détourné !!!!!');
        });
});
```

Explications :

- $ est équivalent à jquery, cela permet d'utiliser les fonctionnalités de jquery.
- $(document) permet de sélectionner le document et donc la page web active.

- $(document).ready permet de préparer du code javascript qui sera exécuté lorsque le document sera "ready" ç-à-d lorsque le document sera chargé dans le navigateur web.
- $(document).ready(function(){}); permet en plus de créer ce que l'on appelle une fonction anonyme (écrite à la volée). Ici elle ne prend pas de paramètres.
-
```
$(document).ready(function(){
        $("*").click(function(){
                alert("détournement du clic");
        });
});
```

Permet de **détourner le clic** sur n'importe quel élément (*) du document actif, quand le document est chargé (ready), si on clique sur n'importe quel élément de la page web, une page d'alerte est affichée.

Important :

Il faut placer vos codes jquery dans $(document).ready(function(){});

Pour intercepter un événement sur un élément du document, il faut d'abord sélectionner cet élément.

6.3. Sélection d'un élément d'un document web (= sélection CSS).

Ce qui va vous permettre de sélectionner un élément dans un document web c'est ce que l'on appelle le sélecteur CSS.

Observer les parties suivantes et testez ensuite le code de la partie 6.3.5.

$(this) représente le noeud DOM sur lequel survient l'événement.

6.3.1. * : permet de sélectionner tous les éléments d'un document web

```
$("*").gestionEvénement(function(){});
```

Exemple :

```
$("*").click(function(){
        alert("détournement du clic");
});
```

6.3.2. Balise html

```
$("balisehtml").gestionEvénement(function(){});
```

Exemple :

```
$("p").hover(function(){
        $(this).css("background-color","red");
});
```

6.3.3. Class

```
$(".classhtml").gestionEvénement(function(){});
```

Exemple :

```
$(".rouge").hover(function(){
        $(this).css("background-color","blue");
});
```

6.3.4. Id

$("#classhtml").gestionEvénement(function(){});

Exemple :

```
$("#rouge").hover(function(){
        $(this).css("background-color","green");
});
```

6.3.5. Codes associés.

Recopiez les codes 6.3.1 à 6.3.4 dans le fichier "monscript.js", insérez le code html dans index.html.

```
<!DOCTYPE html>
<html>
<head>
        <meta charset="UTF-8">
        <title>Code 6.3</title>
        <script  src="./js/jquery.js">

        </script>
        <script src="./js/monscript.js">
        </script>
</head>
<body>
        <p> premier paragraphe
        </p>
        <p> second paragraphe
        </p>
        <p class="rouge"> troisieme paragraphe avec class rouge
        </p>
        <p id="vert"> quetrieme paragraphe avec id vert
        </p>
        <span> du texte avec span...</span>
</body>
</html>
```

6.3.6. $(this).

Ce sélecteur spécial permet de sélectionner l'élément sur lequel se produit l'événement.

6.4.Capture d'un événement : le clic de la souris.

Décomposons le code qui va permettre de capturer l'événement "clic" de la souris sur un élément de la page web :

```
$(sélecteur).click(function(donnees){
        instruction(s);
});
```

Exemple :

```
$("span").click(function(e){
        alert(e.which); //retourne 1 si le bouton gauche est pressé, 2 milieu
        alert(e.pageX); //retourne l'abscisse depuis le coin supérieur gauche
        alert(e.pageY); //retourne l'ordonnée depuis le coin supérieur gauche
```

```
    e.preventDefault(); //permettra d'annuler le submit dans un formulaire
});
```

Commentaires :

Lors du clic sur une balise span, il sera possible de détourner le click mais aussi de :

- d'afficher quel bouton de souris a été pressé, quelle est la position X, Y du pointeur de la souris, etc...
- d'annuler la soumission du formulaire en appliquant la méthode preventDefault();

6.5.Les événements (http://www.w3schools.com/jquery/jquery_ref_events.asp) :

Vous pouvez gérer d'autres événements survenants dans vos pages web :

- blur() : perte du focus,
- change() : changement de valeur,
- click, dblclick, mouseXXX (mouseout, mousemove, etc.)
- focus,
- hover,
- keyup,
- live, permet de manipuler les éléments du document ainsi que ceux qui sont rajoutés dynamiquement, ==> on
- mousedown,
- resize,
- scroll,
- select,
- submit,
- toggle : affiche ou cache un élément,
- etc...

6.6.Les effets.

Il est possible de créer des effets sur les éléments d'un document tel que :

- hide, show : cacher un élément,

```
$("p").click(function(){
  $(this).hide();
});
```

- hide, show : cacher un élément,

```
$("p").click(function(){
  $(this).hide("slow");
});
```

- slideToggle, slideUp, slideDown : cacher/découvrir un élément en faisant une animation,

```
$("p").click(function(){
  $(this).slideToggle("slow");
});
```

- fadeTo, fadeIn, fadeOut : cacher/découvrir un élément en faisant une animation de fondu,

```
$("button").click(function(){
  $("div").fadeTo("slow",0.25);
});
$("button").click(function(){
  $("div").fadeOut(4000);
});
```

- animate : animer des éléments.

```
  $("button").click(function(){
    $("div").animate({height:300},"slow");
    $("div").animate({width:300},"slow");
    $("div").animate({height:100},"slow");
    $("div").animate({width:100},"slow");
  });


///// code CSS à insérer au fichier css
animate({width:"70%",opacity:0.4,marginLeft:"0.6in",fontSize:"3em"});
/////
```

6.7.Modifier le contenu de la page web.

Il est possible de modifier le contenu de la page web avec les outils suivants :

- $(selecteur).html(valeur) // remplace la valeur html équiv. à innerHTML
- $(selecteur).append(valeur) //ajoute après la valeur html
- $(selecteur).prepend(valeur) //ajoute avant la valeur html
- $(selecteur).before(valeur) //ajoute avant l'élément sélectionné html
- $(selecteur).after(valeur) ////ajoute après l'élément sélectionné html

- $(selecteur).val() //accède en lecture à la valeur de la balise
- $(selecteur).val(value) //modifie avec la valeur value la valeur de la balise

6.8.Modifier le CSS de la page web.

Il est possible de lire/modifier avec css.

- $(selecteur).css(valeur), // atteint pour afficher ou manipuler la valeur CSS
- $(selecteur).css(propriété,valeur), // remplace la valeur CSS par la valeur transmise
- $(selecteur).css({propriétés:valeurs}), // remplace les valeurs CSS par les valeurs transmises
- $(selecteur).width ou height // permet de manipuler la largeur et hauteur.

6.9.Ajax.

Ajax permet à un script javascript ***d'exécuter un script sur un serveur web*** et d'en ***récupérer*** ***l'affichage***. Ce résultat peut ensuite être placé dans un élément HTML de la page courante.

6.9.1. Exemple :

Choix d'un login dans un champ de formulaire. Un message écrit en rouge est affiché lorsque le login choisi n'est pas disponible, en vert sinon.

Comment procéder?

Lors de chaque frappe dans le champ de saisie, la valeur du champ est transmise à un script php qui vérifie si cette valeur existe ou non déjà dans la table utilisateur de la base de données.

6.9.2. Le champ de saisie et la zone ou sera affichée le message (vert ou rouge)

Code se trouvant dans le fichier "saisie.html"

```
<input type="text" name="identifiant" id="identifiant" value=""/>
<span id="erreur"></span>
```

6.9.3. La Table

Vous utiliserez la table utilisateur et plus particulièrement le champ "identifiant", clé primaire.

6.9.4. Le script php "verifId.php"

Le script php va :

- recevoir en paramètre (dans le tableau associatif $_POST) l'identifiant saisi et le placer dans $id,
- ensuite exécuter la requête SQL : select * from utilisateur where identifiant='$id',
- si la requête retourne une ligne, le script affiche "idUtilise",
- sinon le script affiche "idNonUtilise"

```
<?php
	include_once("./BonChemin/param.php");
	include_once "./BonChemin/gestionUtilisateur.php";
	try{
		$co=new gestionUtilisateur(USR ,MDP ,BDD );

		//évite les injections de code
		$id=strip_tags($_POST['identifiant']);
		$id=htmlspecialchars($id);

		$req="select * from utilisateur where identifiant='$id'";
		$res=$co->requeteSelection($req);
		if($ligne=$res->fetch()){	//ici on récupère le premier tuple
			echo 'idUtilise';
		}else{
			echo 'idNonUtilise';
		}
	}
	catch (PDOException $ex){
		echo "Erreur lors de la connexion à la bd...";
	}
?>
```

6.9.5. Le code javascript exploitant tout cela

Code à placer dans le fichier "saisie.html".

```
// co
$(document).ready(function() {
		//capture de l'événement keyup
		$("#identifiant").keyup(function(){
			//exéction du script verifId à chaque fois que l'on presse une touche
			$.ajax({
				type:"post",
				url: './js/verifId.php',
				data: "identifiant="+$(this).val(),
//		data: "identifiant="+$(this).val()+"&couleur=bleu",
				success: function(data) {
				//data contient la valeur affichée par le script php
				if (data=="idUtilise"){
					$("#erreur").html("Identifiant déjà utilisé");
					$("#erreur").css("color","red");
				}else{
					$("#erreur").html("Identifiant disponible");
					$("#erreur").css("color","green");
				};
				}
			});
		});
});
```

6.9.6. Exercice :

Complétez le fichier saisie.html pour gérer la non redondance du courriel. ç-à-d, proposez à l'utilisateur de saisir un courriel mais à chaque frappe de clavier, vérifier que le courriel ne soit pas déjà présent dans la table utilisateur.

S'il est présent, affichez en rouge le message "courriel deja utilisé, sinon affichez l'image suivante :

 (téléchargez là sur http://icones.pro/?s=accepter)

6.10.Plug-in.

De nombreux développeurs écrivent des plug-in pour jquery, ces plug-in peuvent être téléchargés sur le site des développeurs pour ensuite être inclus dans vos pages web.

6.10.1. Calendrier (datepicker, utilisant jqueryui).

Le plug-in Calendar, c'est un plug-in qui permet d'ajouter un champ de type calendrier dans une page web.

Pour réaliser cela, il faut ajouter à votre page web :

- Les fichiers javascript de jquery et de jqueryui (user interface library), ces fichiers se trouvent sur internet (il est possible aussi de les sauvegarder dans un dossier de votre site web).

```
<!--
        on inclut ici les bibliothèques jquery, jqueryui ainsi que le css de jqueryui
 -->
<script  src="./js/jquery.js"></script>
<script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/jquery-ui.min.js"></script>
<link href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css"
rel="stylesheet" type="text/css"/>
```

- Un champ de saisie possédant par exemple un id

```
<p>
        date<input type="text" id="madate" name="madate"/>
</p>
```

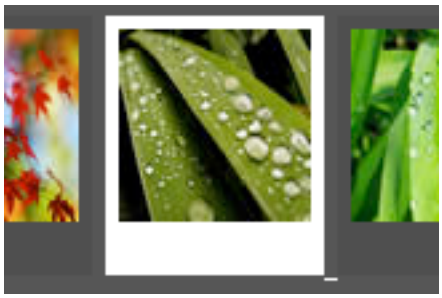
L'instruction jquery réalisant l'affichage du calendrier pour le champ de saisie créé.

```
<script type="text/javascript">
        $(document).ready(function() {
                $("#madate").datepicker();
        });
</script>
```

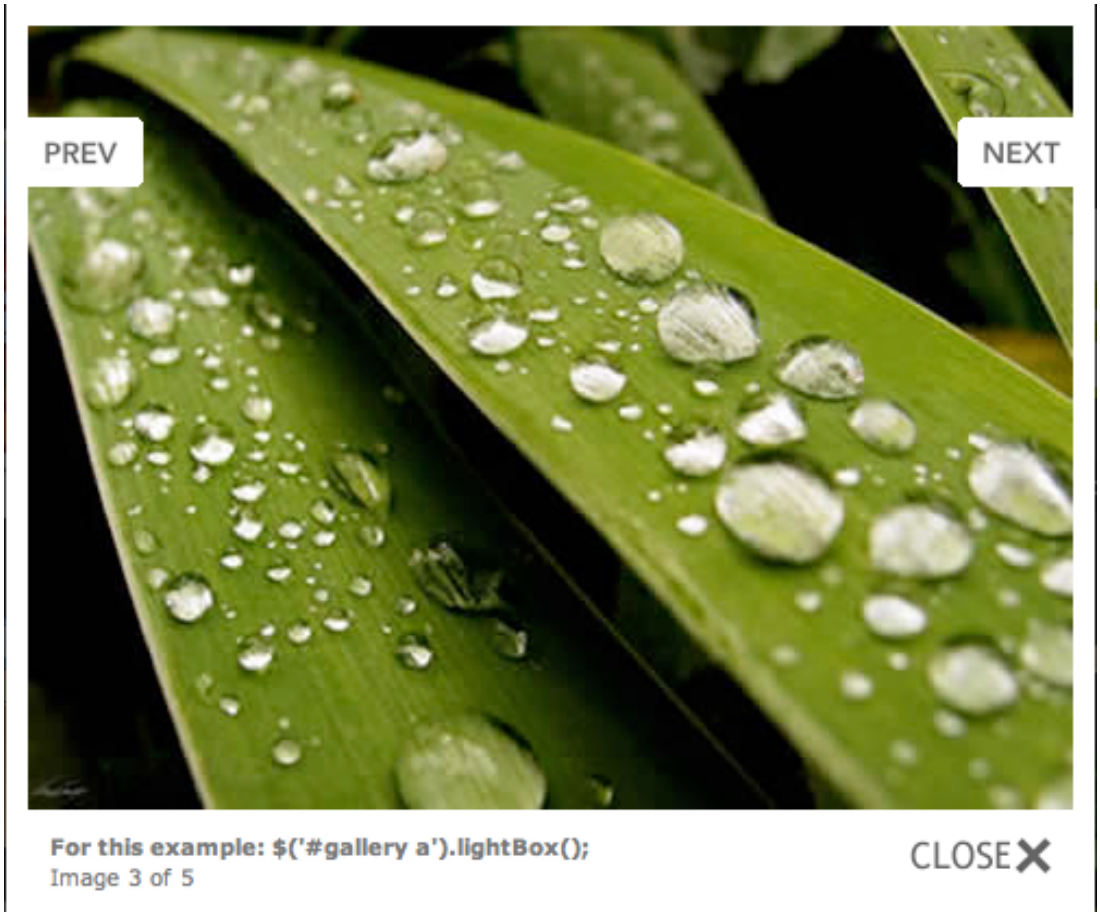
6.10.2. Effet d'image - lightbox (utilisant jquery).

Cet effet vous permettra d'agrandir une image avec une animation d'échelle.

Exemple : image miniature que laquelle vous allez cliquer.



L'image est agrandie.



Pour réaliser cela, il faut ajouter à votre site web les fichiers suivants :

- Téléchargez le fichier compressé se trouvant à l'adresse : http://leandrovieira.com/download/7/
- Décompressez ce fichier et copiez les dossiers suivants dans votre site web :
  - css (contient le fichier css necessaire à lightbox),
  - images (contient les images personnalisables de lightbox, les images PREV et NEXT par exemple),
  - js (contient une version de jquery ainsi que jquery.lightbox-0.5.js),
  - photos (contient des photos exemples que l'on va utiliser).

Ensuite, ajoutez à votre page web le code suivant :

- on inclut les bibliothèques nécessaires ainsi qu'un lien vers le css de lightbox :

```
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.lightbox-0.5.js"></script>

<link rel="stylesheet" type="text/css" href="css/jquery.lightbox-0.5.css" media="screen" />
```

- Affichage d'images miniatures :

```
<a href="./photos/image1.jpg" class="lb"><img src="./photos/thumb_image1.jpg" width="72" height="72"
alt="" /></a>
<a href="./photos/image2.jpg" class="lb"><img src="./photos/thumb_image2.jpg" width="72" height="72"
alt="" /></a>
<a href="./photos/image3.jpg"><img src="./photos/thumb_image3.jpg" width="72" height="72" alt="" /></a>
```

- L'instruction jquery réalisant l'affichage des images.

```
<script type="text/javascript">
        $(document).ready(function() {
                $('a.lb').lightBox(); // Selectionne toutes le balises <a class="lb">  dans la page

        });
</script>
```