

PHP

A écrire à l'intérieur du code .html

Il est possible de mettre du code HTML a l'intérieur du code PHP

<code><?php Le code PHP se met ici ?></code>	Dis que CECI est du PHP
<code>header('Location: minichat.php');</code>	Redirige vers la page <code>minichat.php</code>
<code>// commentaire</code> <code>/* commentaire sur plusieurs lignes */</code>	commentaire
<code>echo "Ceci est du texte";</code>	Ecrit du texte
<code>include("menus.php");</code>	Inséré le contenu de la page <code>menu.php</code>

<pre> if (\$age <= 12) { } elseif (\$age >= 12) { } else { } </pre>	<p>Si \$age est inférieur ou égale à 12</p> <p>Sinon si \$age est supérieur ou égale à 12</p> <p>Sinon</p>
<pre> \$note = 10; switch (\$note) { case 0: echo "Tu es vraiment un gros nul !!!"; break; case 5: // dans le cas où \$note vaut 5 echo "Tu es très mauvais"; break; case 7: // dans le cas où \$note vaut 7 echo "Tu es mauvais"; break; case 10: // etc. etc. echo "Tu as pile poil la moyenne, c'est un peu juste..."; break; case 12: echo "Tu es assez bon"; break; case 16: echo "Tu te débrouilles très bien !"; break; case 20: echo "Excellent travail, c'est parfait !"; break; default: echo "Désolé, je n'ai pas de message à afficher pour cette note"; } </pre>	<p>Indique sur quelle variable on travaille</p> <p>Si \$note vaut 0</p> <p>Stop</p> <p>Si \$note vaut 5</p> <p>Stop</p> <p>Si \$note vaut 7</p> <p>Stop</p> <p>Si \$note vaut 10</p> <p>Stop</p> <p>Si \$note vaut 12</p> <p>Stop</p> <p>Si \$note vaut 16</p> <p>Stop</p> <p>Si \$note vaut 20</p> <p>Stop</p> <p>Default si aucune case n'est validé</p>
<pre> while (\$continuer_boucle == true) { instructions à exécuter dans la boucle } </pre>	<p>Temps que \$continuer_boucle == True</p>

<pre>for (\$nombre_de_lignes = 1; \$nombre_de_lignes <= 100; \$nombre_de_lignes++) { echo 'Ceci est la ligne n°' . \$nombre_de_lignes . '
'; }</pre>	Pour nombre_de_lignes=1 , nombre_de_ligne inferieur ou égale a 100 , nombre_de_ligne +1
AND - OR	ET - OU
calculCube(valeur, valeur, valeur...)	Appel de la fonction calculCube
<pre>function VolumeCone(\$rayon, \$hauteur) { \$volume = \$rayon * \$rayon * 3.14 * \$hauteur * (1/3); return \$volume; }</pre>	Cree une fonction volumeCone Retourne la variable volume
<pre>try { } catch (Exception \$e) { die('Erreur : ' . \$e->getMessage()); }</pre>	Lance le code de try, si il y a une erreur, affiche le code de catch sans utiliser le code de try Ici : affiche le code d'erreur
crypt('kangourou')	CRYPTER LA CHAÎNE DE CARACTÈRE

Variables:

\$age_du_visiteur = 17;	Cree une variable age_du_visiteur de valeur 17
echo \$age_du_visiteur;	Affiche la variable age_du_visiteur
echo "le visiteur a : " . \$age_du_visiteur . "ans";	Permet de concaténer avec variable
\$nombre = 2 + 4; \$nombre = 5 - 1; \$nombre = 3 * 5; \$nombre = 10 / 2; \$nombre = 3 * 5 + 1; \$nombre = (1 + 2) * 2;	\$nombre prend la valeur 6 \$nombre prend la valeur 4 \$nombre prend la valeur 15 \$nombre prend la valeur 5 \$nombre prend la valeur 16 \$nombre prend la valeur 6
\$nombre = 10 % 5; \$nombre = 10 % 3;	\$nombre prend la valeur 0 car la division tombe juste \$nombre prend la valeur 1 car il reste 1
strlen(\$phrase);	Nombre de caractère de la chaine
str_replace('b', 'p', 'bim bam boum')	Remplace les b par p dans bim bam boum
str_shuffle(\$chaine)	Mélange la chaine aléatoirement
strtolower(\$chaine)	Écrit la chaine en minuscule
Strtoupper(\$chaine)	Écrit la chaine en majuscule
date('Y')	Récupère la date H :Heure i : Minute d : Jour m : Mois Y : Année

Variables serveur

<code>\$_SERVER</code>	Valeurs renvoyées par le serveur.
<code>\$_SERVER['REMOTE_ADDR']</code>	Adresse IP du client qui regarde la page
<code>\$_SESSION :</code>	Variables de session. Ce sont des variables qui restent stockées sur le serveur le temps de la présence d'un visiteur.
<code>\$_COOKIE</code>	Contient les valeurs des cookies enregistrés sur l'ordinateur du visiteur.
<code>\$_POST</code>	Données envoyées en paramètres dans l'URL.
<code>\$_FILES :</code>	Fichiers qui ont été envoyés via un formulaire
<code>session_start()</code>	Démarre une session pour un utilisateur (à mettre sur toutes les pages et au tout début DANS LES PREMIERES LIGNES dans <?php ?>)
<code>\$_SESSION['nom'] = "coucou"</code>	Crée une variable qui sera stockée durant toute la présence de l'utilisateur sur n'importe quel page du site
<code>session_destroy()</code>	Ferme la session pour l'utilisateur
<code>setcookie('pseudo', 'M@teo21', time() + 365*24*3600, null, null, false, true)</code>	Crée un cookie pseudo avec la valeur M@teo21 qui expirera dans 365*24*3600 secondes (à mettre au tout début DANS LES PREMIERES LIGNES dans <?php ?>) refaire si on change la valeur du cookie
<code>\$_COOKIE['pseudo']</code>	Affiche le cookie

Tableaux

<code>\$prenoms = array ('François', 'Michel', 'Nicole', 'Véronique', 'Benoît');</code>	Cree un tableau de 5 valeur
<code>\$prenoms[] = 'François';</code> <code>\$prenoms[] = 'Michel';</code> <code>\$prenoms[] = 'Nicole';</code>	Cree prenom[0] prenom[1] prenom[2] (possible de mettre le chiffre directement a l'intérieur)
<code>echo \$prenoms[1];</code>	Affiche le contenu de prenom[1]
<code>\$coordonnees = array ('prenom' => 'François', 'nom' => 'Dupont', 'adresse' => '3 Rue du Paradis', 'ville' => 'Marseille');</code>	Cree un dictionnaire
<code>\$coordonnees['prenom'] = 'François';</code> <code>\$coordonnees['nom'] = 'Dupont';</code> <code>\$coordonnees['adresse'] = '3 Rue du Paradis';</code> <code>\$coordonnees['ville'] = 'Marseille';</code>	Pareil qu'au-dessus en rajoutant ligne par ligne
<code>echo \$coordonnees['ville'];</code>	Affiche le contenu de <code>coordonnees['ville']</code>
<code>foreach(\$prenoms as \$element)</code> { }	Parcours le tableau <code>prenom</code> en insérant au fur et a mesure la valeur dans la variable <code>\$element</code>
<code>foreach(\$coordonnees as \$cle => \$element)</code> { }	Prend la clé ET l'élément du tableau coordonnées
<code>print_r(\$coordonnees);</code>	Affiche le contenu complet de coordonnees (utilisé pour le débogage)
<code>array_key_exists('cle', \$array);</code>	Renvoie TRUE si <code>cle</code> existe dans <code>array</code>
<code>in_array('Cerise', \$fruits)</code>	Renvoie TRUE si l'élément <code>Cerise</code> existe dans <code>fruit</code>
<code>array_search('Banane', \$fruits);</code>	Renvoie la clé de <code>Banane</code> dans <code>fruit</code>
<code>(int) \$variable</code>	Remplace une chaine en entier (met 0 si ce n'est pas un entiere)
<code>isset(\$prenom)</code>	Renvoie TRUE si la variable <code>\$prenom</code> existe

Lire/crire fichier

<code>\$monfichier = fopen('compteur.txt', 'r+');</code>	<p>Ouvre le fichier en lecture et ecriture</p> <p>r Ouvre le fichier en lecture seule.</p> <p>r+ Ouvre le fichier en lecture et écriture.</p> <p>a Ouvre le fichier en écriture seule. Le fichier est automatiquement créé.</p> <p>a+ Ouvre le fichier en lecture et écriture. le fichier est créé automatiquement. Le répertoire doit avoir un CHMOD à 777 ! le texte sera rajouté à la fin.</p>
<code>\$ligne = fgets(\$monfichier);</code>	<p>Lit une ligne de <code>monfichier</code>, refaire pour lire la prochaine</p> <p>fgetc = caractère a caractère</p>
<code>fseek(\$monfichier, 0);</code>	Remet le curseur au début du fichier (remplace la ligne si il y a un truc écrit)
<code>fputs(\$monfichier, 'Texte à écrire');</code>	Ecrit dans le fichier
<code>fclose(\$monfichier);</code>	Ferme le fichier

A partir d'un lien qu'on envoie :

<pre>Dis-moi bonjour !</pre>	amène a la page bonjour.php et envoie les paramètre nom=Dupont et prenom = Jean
<code>\$_GET['prenom']</code>	Reprend la valeur prenom qui est dans l'adresse web

Pour les formulaires :

<pre><form method="post" action="cible.php"> <p> </p> </form></pre>	Method : post : on peut mettre autant que l'on veut (conseillé) Get : se met dans l'adresse web Action : lien page validation
<code><input type="submit" value="Valider" /></code>	Le bouton submit enverra les information et amene l'utilisateur a la page qui contiendra ces information s'utile avec <code>\$_POST[variable]</code>
<code>htmlspecialchars(\$_POST['prenom'])</code>	Toujours protéger avec cette fonction les POST et GET
Apres envoie de fichier : <code>\$_FILES['monfichier']['name']</code> <code>\$_FILES['monfichier']['type']</code> <code>\$_FILES['monfichier']['size']</code> <code>\$_FILES['monfichier']['tmp_name']</code> <code>\$_FILES['monfichier']['error']</code>	Apres envoie de fichier Contient le nom du fichier envoyé par le visiteur. Indique le type du fichier envoyé. Si c'est une image gif par exemple, le type sera image/gif. Indique la taille du fichier envoyé. (En octet, limité à 8Mo max) Cette variable contient l'emplacement temporaire du fichier Code d'erreur, 0 = aucun

BASE DE DONNE

utilisateur : root sans mot de passe en local

Pour remettre la base sur internet : aller sur le phpMyAdmin de l'hébergeur. Et chercher l'adresse de phpMyAdmin. Une fois dessus, se rendre dans l'onglet Importer.

<pre>try { \$bdd = new PDO('mysql:host=localhost;dbname=test;charset=utf8', 'root', 'password'); } catch (Exception \$e) { die('Erreur : ' . \$e->getMessage()); }</pre>	Connection à la base de donné host : adresse de la base dbname : nom de la base charset : encodage utilisateur mot de passe
<pre>\$reponse = \$bdd->query('Tapez votre requête SQL ici');</pre>	reponse contiendra les données de la requête SQL
<pre>\$donnees = \$reponse->fetch();</pre>	Données prend les valeurs de la base (refaire pour passer à la donnée suivante)
<pre>while (\$donnees = \$reponse->fetch()) { } </pre>	Parcoure toute les données avec données prend la nouvelle valeur à chaque boucle
<pre>array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION);</pre>	Permet d'afficher les erreurs lié à la base
<pre>echo \$donnee["nom"];</pre>	affiche en fonction du nom de la colonne (conseillé)
<pre>\$reponse->closeCursor();</pre>	A mettre à chaque fois que l'on a fini de traité un SELECT sql
<pre>\$bdd->exec(REQUETE)</pre>	SANS VARIABLE, faire ceci pour toute requete
<pre>PROTECTION \$req = \$bdd->prepare('SELECT nom FROM jeux_video WHERE possesseur = :possesseur AND prix <= :prix_max'); \$req->execute(array('possesseur' => \$_GET['possesseur'], 'prix_max' => \$_GET['prix_max']));</pre>	PROTECTION Prépare la requête avec les variable Modifie la valeur des variable de la requête
<pre>\$req = \$bdd->prepare('INSERT INTO jeux_video(nom, possesseur, console, prix, nbre_joueurs_max, commentaires) VALUES(:nom, :possesseur, :console, :prix, :nbre_joueurs_max, :commentaires)'); \$req->execute(array('nom' => \$nom, 'possesseur' => \$possesseur, 'console' => \$console, 'prix' => \$prix, 'nbre_joueurs_max' => \$nbre_joueurs_max, 'commentaires' => \$commentaires));</pre>	Insertion de donné, se fais de la même façon
<pre><?php \$req = \$bdd->prepare('UPDATE jeux_video SET prix = :nvprix, nbre_joueurs_max = :nv_nb_joueurs WHERE nom = :nom_jeu'); \$req->execute(array('nvprix' => \$nvprix, 'nv_nb_joueurs' => \$nv_nb_joueurs, 'nom_jeu' => \$nom_jeu)); ?></pre>	Modifie une donnée, se fais de la même façon
<pre>\$bdd->exec(DELETE FROM jeux_video WHERE nom='Battlefield 1942')</pre>	Supprime une donnée

PCRE LANGUAGE SIMPLIFIE

# #	Toujours délimité un REGEX entre des dièses
preg_match("#guitare#", "J'aime jouer de la guitare")	Permet de faire une recherche avec un langage REGEX et ce en quoi on fais la recherche ici : renvoie TRUE car guitare est dans la chaine
#guitare piano#	Verifie sur il y a guitare OU piano
#Quoi \?#	Metre "\" pour tous les metacaractère : # ! ^ \$ () [] { } ? + * . \
#^Bonjour#	Bonjour devra être le premier mot de la chaine
#zéro\$#	zéro devra obligatoirement être le dernier mot de la chaine
#gr[ioa]s#	Recherche gris OU gros OU gras
#[a-z]#	Une lettre (de A à Z)
#[0-9]#	Un chiffre (de 0 à 9)
#[a-z0-9]#	Une lettre OU un chiffre
#[^0-9]#	Recherche tout sauf un chiffre
#a?#	Reconnais 0 ou 1 "a" PAS PLUS de 1
#a+#	Au moins 1 "a" A PARTIR DE 1
#a*#	0 ou plusieurs a
#Ay(ay oy)*#	VRAI pour AY et autant de ay ou de oy que l'on veut
#a{3}#	Fonctionne que pour aaa
#a{3,5}#	Fonction pour aaa, aaaa, aaaaa seulement
#a{3,}#	Fonctionne pour autant de a que l'on veut
(j'	= [0-9] = [^0-9] = [a-zA-Z0-9_] = [^a-zA-Z0-9_] Indique une tabulation Indique une nouvelle ligne Indique un retour chariot Indique un espace blanc Indique ce qui n'est PAS un espace blanc (\t \n \r) Indique n'importe quel caractère. Sauf entrée "\n"
#REGEX#i	Le i demande a que le REGEX ne sois plus sensible à la casse, il ne fera plus la différence entre majuscule/minuscule
#REGEX#s	Le point "." acceptera tout, même les entré "\n"
#REGEX#U	Le REGEX s'arrêtera le + tôt possible ([b]/[b] , il s'arretera au premier [/b] qu'il rencontrera
\$texte = preg_replace('#\[b\](.+)\[/b\]#i', '\$1', \$texte);	Une parenthèse = une variable chiffrer (\$1 pour la premier, \$2 pour la deuxième...) \$0 = toute la REGEX dans l'ordre : fais une recherche, remplace ce qu'il en a entre parenthèse par un autre code, (ici met le même code en gras avec strong) la partie ou l'on fais la recherche (chaine de caractère)
(?:nsti)	La parenthèse ne comptera pas pour une variable et sera sauté

LES OBJETS POO

<pre> class Membre { private \$pseudo; private \$email; private \$signature; private \$actif; public function envoyerEMail(\$titre, \$message) { mail(\$this->email, \$titre, \$message); } public function bannir() { \$this->actif = false; \$this->envoyerEMail('Vous avez été banni', 'Ne revenez plus !'); } public function getPseudo() { return \$this->pseudo; } public function setPseudo(\$nouveauPseudo) { if (!empty(\$nouveauPseudo) AND strlen(\$nouveauPseudo) < 15) { \$this->pseudo = \$nouveauPseudo; } } } </pre>	<p>Cree la classe membre</p> <p>Variable pseudo Variable email Variable signature Variable actif</p> <p>Cree la fonction envoyerEMail avec paramètre titre et message Envoie le mail</p> <p>Cree la fonction bannir</p> <p>Met actif a FALSE Appel la fonction mail avec titre et message</p> <p>Fonction getPseudo</p> <p>Retourne le pseudo</p> <p>Fonction setPseudo avec paramètre nouveau pseudo</p> <p>Vérifie si le pseudo n'existe pas déjà et la longueur < 15</p> <p>Changement du pseudo</p>
<pre> include_once('Membre.class.php'); \$membre = new Membre(); \$membre->setPseudo('M@teo21'); echo \$membre->getPseudo() . ', je vais te bannir !'; \$membre->bannir(); </pre>	<p>Inclue le fichier membre.class.php qui contien la classe demandé</p> <p>Cree un nouveau membre du nom membre Ajoute un pseudo M@teo21 Envoie au membre qu'il va être banni Le banni</p>