

concept	définition	PHP
affectation	attribution d'une valeur (par exemple 5) à une variable (par exemple l'entier \$a)	<code>\$a = 5;</code>
affichage dans la page HTML	insertion dans la page HTML d'une chaîne de caractères	<code>echo("Titre de la page");</code>
aléatoire	nombre entier pseudo-aléatoire entre un minimum et un maximum	<code>rand(5,15); //fournit un entier entre 5 et 15 inclus random_int(5,15); //même chose mais adapté pour un usage en cryptographie</code>
appel de fonction	utilisation d'une fonction sur d'éventuelles données en entrée	<code>algo1(\$entree1,42); \$s = algo1(18, 42); //on stocke dans une variable \$s le résultat renvoyé.</code>
base de données SQL : création d'une connexion à la base avec PDO	création d'une connexion \$link à la base ma_base, sur sa propre machine avec l'identifiant root et le mot de passe vide (valeurs par défaut dans Wampserver), en s'assurant que toutes les chaînes de caractères transférées seront encodées en UTF-8	<code>\$link = new PDO('mysql:host=localhost;dbname=ma_base', 'root', '', array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8")); // pour le serveur de l'UPEM, remplacer localhost par sqletud.u-pem.fr</code>
base de données SQL : envoi d'une requête SQL simple avec PDO	envoi, suite à l'ouverture d'une connexion \$link à une base de données, d'une requête SQL pour sélectionner les lignes d'une table users dont le champ prenom commence par "Phil" et affichage des résultats dans une liste à puces	<code>\$sql = "SELECT prenom, nom FROM users WHERE prenom LIKE 'Phil%'"; // On prépare la requête avant l'envoi : \$req = \$link -> prepare(\$sql); \$req -> execute(); // On crée une liste non numérotée avec les résultats echo ''; while(\$data = \$req -> fetch()){ // On affiche chaque résultat sous forme d'un item de la liste echo ''. \$data['prenom']. ' '. \$data['nom']. ''; } \$req = null; echo '';</code>
base de données SQL : envoi d'une requête SQL simple avec PDO intégrant une variable transmise en GET	envoi, suite à l'ouverture d'une connexion \$link à une base de données, d'une requête SQL pour insérer dans le champ courriel d'une table users le contenu d'un champ input de formulaire en GET ayant pour attribut name la valeur "ad_courriel"	<code>// On vérifie que l'adresse de courriel a bien été transmise en GET if(isset(\$_GET["ad_courriel"])){ \$sql = "INSERT INTO users(courriel) VALUES (:courriel)"; // On prépare la requête avant l'envoi : \$req = \$link -> prepare(\$sql); // On exécute la requête en insérant la valeur transmise en GET \$req -> execute(array(":courriel" => \$_GET["ad_courriel"])); \$req = null; // On affiche l'adresse inscrite en évitant une injection de code JS echo "L'adresse ".htmlentities(\$_GET["ad_courriel"])." est inscrite !"; }</code>
base de données SQL : fermeture d'une connexion à la base avec mysqli	fermeture de la connexion \$link à une base de données	<code>\$link = null;</code>
booléen	valeur vraie ou fausse	<code>\$resultat = true; \$resultat = false;</code>
boucle	structure de programmation qui permet d'exécuter plusieurs fois une suite d'instructions	<code>while(...){ // instructions à répéter }</code>
boucle de parcours d'un tableau	structure de programmation qui permet de visiter toutes les cases d'un tableau associatif \$tab	<code>foreach (\$tab as \$key => \$value){ echo("La case n°". \$key. " a la valeur ". \$value); }</code>
chaîne de caractères	ensemble de caractères consécutifs, toujours notée entourée de guillemets droits doubles ou simples	<code>\$chaine1 = "toto"; \$chaine2 = 'blabla';</code>
commentaire	ligne d'un code source qui n'est pas exécutée	<code>//Commentaire</code>
commentaire multi-ligne	lignes consécutives d'un code source qui ne sont pas exécutées	<code>/* Mon commentaire sur plusieurs lignes */</code>
concaténation	opération qui juxtapose deux chaînes de caractères l'une après l'autre pour les réunir en une seule	<code>"toto"."1" est la chaîne de caractères "toto1"</code> En PHP, il est possible (mais déconseillé, car ça ne fonctionne pas dans d'autres langages) d'intégrer des variables directement dans une chaîne de caractères "Mon prénom est \$prenom !"; est équivalent à "Mon prénom est" . \$prenom. " !";
déclaration d'une variable	définition du nom d'une variable	Inutile en PHP car les variables se reconnaissent facilement, elles commencent par \$.
égalité	avoir la même valeur ; on utilise === pour indiquer que les valeurs sont égales et qu'elles sont de même type	<code>1==1; "toto"=="toto"; \$var1 === \$var2</code>
entier	"nombre entier"	<code>\$numero = 42;</code>
expression régulière	code permettant de représenter de manière compacte un ensemble de chaînes de caractères, et de reconnaître (à l'aide de la fonction preg_match) si une chaîne de caractères donnée fait partie de cet ensemble	<code>function checkEmail(\$email){ \$regex = '/[a-z0-9.+]+@[a-z.-]+\.[a-z]{2,}/'; if(preg_match(\$regex, \$email)){ echo("Adresse de courriel OK"); } else { echo("Adresse de courriel erroné"); } }</code>
flottant	nombre à virgule	<code>\$environPi = 3.14156;</code>

concept	définition	PHP
fonction	suite d'instructions pour résoudre un problème, c'est-à-dire fournir un résultat en sortie à partir de variables en entrée. On peut utiliser une fonction en faisant un "appel de fonction" (appel de la fonction f2 dans l'exemple ci-contre) en fournissant les entrées adaptées (\$entree1 et \$entree2 dans l'exemple) et en récupérant le résultat renvoyé en sortie (dans la variable \$var1, dans l'exemple).	<pre>function algo1(\$entree1, \$entree2){ \$var1 = f2(\$entree1, \$entree2) + \$entree2; return \$var1; }</pre>
initialisation	première affectation dans une variable	Voir <i>affectation</i> ; s'il y a déjà eu une initialisation dans une variable \$var1, l'appel de fonction isset(\$var1) renvoie true, sinon il renvoie false.
instruction PHP	ordre donné à un serveur PHP, élément d'un algorithme codé en PHP	Chaque instruction est terminée par ";" et toutes les instructions se trouvent dans un bloc commençant par <?php et se terminant par ?>.
longueur	pour un tableau \$t, nombre de cases ; pour une chaîne de caractères \$c, nombre de caractères	count(\$t); // ou sizeof(\$t) strlen(\$c);
méthode GET	transmission de valeurs d'un formulaire dans l'URL http://url_du_site.com/page.php? prenom=John&nom=Doe	<pre><!-- Code HTML : --><form action="page.php" method="get"> <input type="text" name="prenom" value="John"/>
 <input type="text" name="nom" value="Doe"/>
 <input type="submit" value="Envoyer"/></form></pre>
méthode POST	transmission de valeurs d'un formulaire dans la requête HTTP (méthode POST) : si on choisit "Choix 2" dans la liste déroulante, la valeur envoyée en POST à la page index.php pour la variable choix sera la chaîne de caractères "2"	<pre><!-- Code HTML : --><form action="index.php" method="post"> <select name="choix"><option value="1">Choix 1</option> <option value="2">Choix 2</option></select> <input type="submit" value="Envoyer"/></form></pre>
modulo	($a \text{ modulo } b$) est le reste dans la division de a par b . On peut utiliser le modulo pour tester si un entier n est pair.	<pre>\$a % \$b //Test de parité de \$n : if(\$n % 2 == 0){ ... }</pre>
remplacement de sous-chaîne	génère une chaîne de caractères dans laquelle toutes les occurrences d'une sous-chaîne ont été remplacées par une autre chaîne de caractères	<pre>str_replace("ss", "s", "poisson"); // renvoie "poison" str_replace(array("c", "o"), array("n", "u"), "cool !"); // renvoie "nuul !"</pre>
renvoyer	donner le résultat d'un algorithme.	return 42;
sous-chaîne	partie d'une chaîne de caractères, constituée de caractères consécutifs	<pre>substr("abcdef", 2, 2); // renvoie "cd" (premier caractère numéroté 0) substr("abcdef", 1, -1); // renvoie "bcde"</pre>
tableau	ensemble de variables (de préférence, de même type) numérotées à partir de 0.	<pre>\$t1 = array(); \$t1[0] = 18; \$t1[1] = 42; // notation plus compacte : \$t1 = array(18,42);</pre>
tableau associatif	ensemble de variables (de préférence, de même type) stockées dans des cases nommées par des valeurs (par exemple par des chaînes de caractères)	<pre>\$t1 = array(); \$t1["a"] = "ah!"; \$t1["o"] = "oh!"; // notation plus compacte : \$t1 = array("a"=>"ah!", "o" => "oh!");</pre>
test	structure de programmation qui permet d'exécuter une instruction de manière conditionnelle	<pre>if(...){ // bloc d'instructions du "alors" } else { // bloc d'instructions du "sinon" }</pre>
type	ensemble de valeurs possibles pour une variable	PHP n'est pas un langage typé : il devine les types (booléen, entier, flottant, chaîne de caractères, tableau, etc.) tout seul, sans qu'on les lui indique et peut faire des conversions entre types si on teste l'égalité de deux variables de types différents
valeur	contenu d'une variable, par exemple entier 42, chaîne de caractères "42", ou tableau à deux cases numérotées 4 et 2	<pre>42 "42" array(4,2)</pre>
variable	élément ayant un nom fixé, qui permet de stocker une valeur, qui peut changer suite à une affectation	Tout nom de variable commence par un \$ en PHP (voir <i>déclaration</i> , <i>initialisation</i> , <i>affectation</i> , <i>type</i> , <i>valeur</i>).
variable transmise en GET	variable récupérable au chargement de la page, dont la valeur a été transmise dans l'URL, par exemple http://url_du_site.com/page.php? prenom=John&nom=Doe	<pre>echo("<h1>Page web de ".\$_GET["prenom"]." ".\$_GET["nom"]."</h1>"); // affiche comme titre de la page : "Page web de John Doe"</pre>
variable transmise en POST	variable récupérable au chargement de la page, dont la valeur a été transmise dans la requête HTTP (voir l'exemple de "méthode POST")	<pre>echo("<i>Vous avez choisi le ".\$_POST["choix"]." !</i>"); // affiche en italique "Vous avez choisi le 2 !"</pre>