2 - Programmer en PHP

Introduction

Introduire la notion de Modélisation d'un problème

Ce que vous allez apprendre dans ce cours :

- Identifier les différents types de serveurs web
- Acquérir une bonne connaissance de l'architecture client/serveur
- Maitriser l'environnement de développement

Vous pouvez télécharger le résumé théorique complet du module de compétences **"Développer des sites web dynamiques"** sur le volet droit de votre écran.

Maitriser le langage PHP

1 - Structure générale d'un script PHP

Balises PHP

- Lorsque PHP traite un fichier, il cherche les balises d'ouverture et de fermeture (<?php et ?>) qui délimitent le code qu'il doit interpréter.
- En version 7, PHP accepte deux syntaxes pour les balises :

<?php ... ?>

<? ... ?>

- Avant la version 7, PHP acceptait deux syntaxes supplémentaires pour les balises :

<script language="php"> ... </script>

<% ... %>

- Tout ce qui est à l'extérieur de la balise PHP est transmis tel quel au navigateur; seul les instructions PHP sont interprétées par le moteur PHP.

La fonction echo

echo - Affiche une chaîne de caractères

PHP inclus une balise ouvrante echo courte <?= qui est un raccourci au code plus verbeux <?php echo

Séparation des instructions

PHP requiert que les instructions soient terminées par un point-virgule à la fin de chaque instruction.

2 - Manipulation des Variables/constantes/Affectation

Les variables essentielles

Un nom de variable valide doit respecter les règles suivantes (après le symbole $) :

- Le nom est sensible à la casse : $a et $A sont deux variables distinctes.
- Le premier caractère doit être une lettre ou un souligné _
- À partir du deuxième caractère seul les lettres, chiffres ou soulignés sont acceptés.

Variables prédéfinies

Les variables "superglobales" sont disponibles quel que soit le contexte du script.

Portée des variables

- Une variable globale doit être déclarée à l'intérieur de chaque fonction afin de pouvoir être utilisée dans cette fonction.
  - Il suffit d'utiliser le mot clé global avant la variable pour l'utiliser.
- Une variable statique a une portée locale uniquement, mais elle ne perd pas sa valeur lorsque le script appelle la fonction.
  - Il suffit d'utiliser le mot clé static avant la variable ou l'attribut ou la méthode pour l'utiliser.
- Les propriétés statiques sont accédées en utilisant l'opérateur de résolution de portée ::
- Les variables statiques peuvent être assignées des valeurs qui sont issue d'expression constante, mais les expressions dynamiques, tel que les appels de fonctions, résulteront en une erreur d'analyse.

Variables dynamiques

- Une variable dynamique prend la valeur d'une variable et l'utilise comme nom d'une autre variable, en utilisant le "$$" précédent la variable.
- Les accolades peuvent aussi être utilisées, pour clairement délimiter le nom de la propriété
- Les superglobales ne peuvent pas être utilisées comme variables dynamiques dans les fonctions ou les méthodes des classes.

Les constantes

- Une constante est un identifiant (un nom) qui représente une valeur simple.
- Les constantes sont sensibles à la casse.
- Par convention, les constantes sont toujours en majuscule.
- Les constantes peuvent être définies en utilisant le mot clé const ou en utilisant la fonction define().
- une constante n'est pas préfixée d'un $
- Pour vérifier qu'une constante est définie, utiliser la fonction defined().
- Si une constante indéfinie est utilisée une Error est renvoyée ~~lancée~~.
- Contrairement aux constantes définies en utilisant l'instruction define(), les constantes définies en utilisant le mot-clé const doivent être déclarées au plus haut niveau du contexte, car elles seront définies au moment de la compilation.

Constantes magiques

Une constante magique est une constante prédéfinie dans PHP qui se change en fonction du contexte.

Nom	Description
__LINE__	La ligne courante dans le fichier.
__FILE__	Le chemin complet et le nom du fichier courant avec les liens symboliques résolus. Si utilisé pour une inclusion, le nom du fichier inclus est retourné.
__DIR__	Le dossier du fichier. Si utilisé dans une inclusion, le dossier du fichier inclus sera retourné. C'est l'équivalent de dirname(__FILE__). Ce nom de dossier ne contiendra pas de slash final, sauf si c'est le dossier racine.
__FUNCTION__	Le nom de la fonction, ou {closure} pour les fonctions anonymes.
__CLASS__	Le nom de la classe courante. Le nom de la classe contient l'espace de nom dans lequel cette classe a été déclarée. Lorsqu'elle est utilisée dans une méthode de trait, __CLASS__ est le nom de la classe dans laquelle le trait est utilisé.
__TRAIT__	Le nom du trait. Le nom du trait inclut l'espace de nom dans lequel il a été déclaré.
__METHOD__	Le nom de la méthode courante.
__NAMESPACE__	Le nom de l'espace de noms courant.

Tab. : Les constantes magiques de PHP source https://www.php.net/manual/fr/language.constants.magic.php

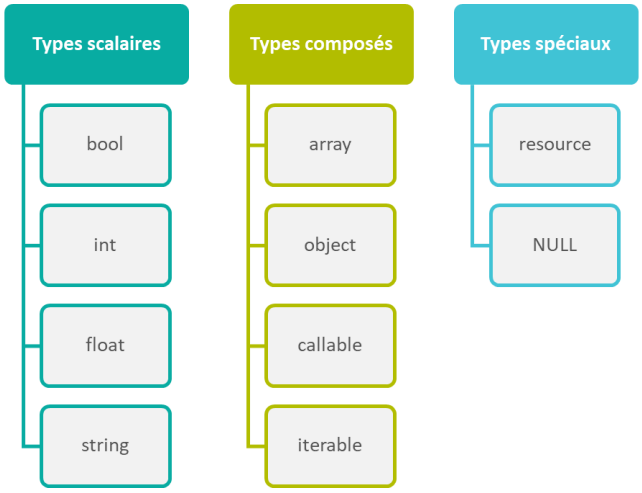
Les opérateurs d'affectation

- L'opérateur d'affectation le plus simple est le signe =
- Il signifie que l'opérande de gauche se voit affecter la valeur de l'expression qui est à droite du signe égal.
- L'affectation par référence se fait grâce au signe &

3 - Manipulation des types de données

Introduction

- Les variables PHP pourront stocker différents types de valeurs comme des nombres ou des tableaux. Par abus de langage, on parle des "types variables" de PHP.
- PHP supporte 10 types basiques :



Type booléen

- Il peut avoir true ou false.
- == est un opérateur qui teste l'égalité et retourne un booléen.
- Pour convertir explicitement une valeur en booléen, utilisez (bool) ou (boolean).

Type entier

- Un entier est un nombre appartenant à l'ensemble $\mathbb{Z}$ = {..., -2, -1, 0, 1, 2, ...}.
- L'opérateur de négation peut être utilisé pour désigner un entier négatif.
- À partir de PHP 7.4.0, les entiers littéraux peuvent contenir des underscores (_) entre les chiffres, pour une meilleure lisibilité des littéraux. Ces underscores sont supprimés par le scanner de PHP. Exemple : $a = 1_203_489 // c'est la même chose que $a = 1203489

Type nombre à virgules

À partir de PHP7.4.0

- LNUM [0-9]+(_[0-9]+)*
- DNUM ([0-9]*(_[0-9]+)*[\.]{LNUM}) | ({LNUM}[\.][0-9]*(_[0-9]+)*)
- EXPONENT_DNUM (({LNUM} | {DNUM}) [eE][+-]? {LNUM})

Type chaine de caractères

- La façon la plus simple de spécifier une chaîne de caractères est de l'entourer de guillemets simples (le caractère ').

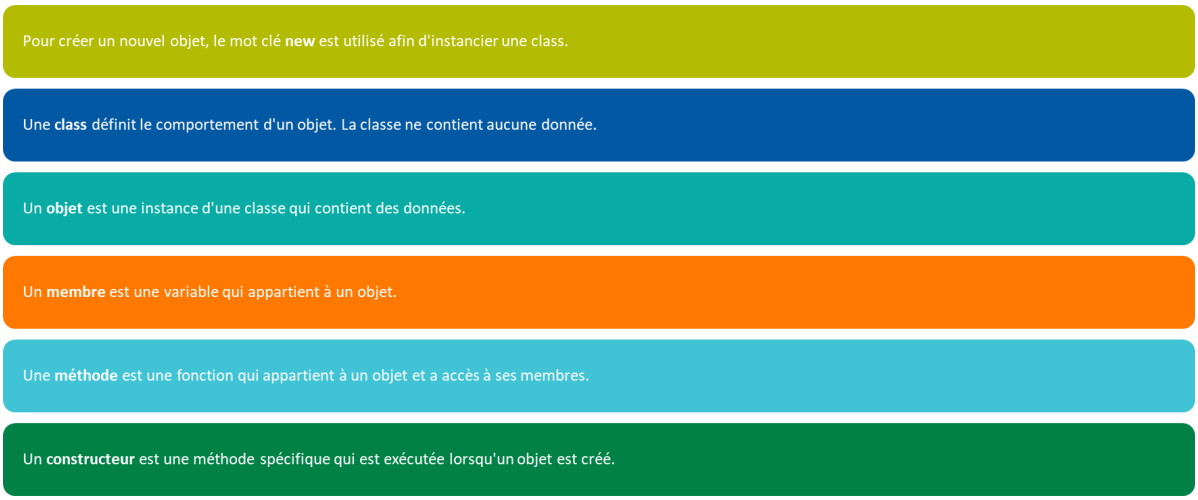
- Si la chaîne de caractères est entourée de guillemets doubles (le caractère ") :

Type tableau

Syntaxe d'un tableau est :

- clé => valeur
- La clé peut être soit un int, soit une chaîne de caractères. Les nombres à virgule flottante seront aussi modifiés en entier.
- La valeur peut être de n'importe quel type.
- Structure de langage array(). La syntaxe courte emplace array() par []
- Le tableau prend un nombre illimité de paramètres, chacun séparé par une virgule

Type objet



Type callable

- Les fonctions de rappel peuvent être des fonctions simples ou des méthodes objet, y compris des méthodes de classe statique.
- Les fonctions de rappel peuvent être identifiées via le type callable.
- Une méthode d'un objet instancié est passée comme un tableau contenant un objet à l'index 0, et le nom de la méthode à l'index 1.
- Les méthodes de classe statiques peuvent également être transmises sans instancier un objet de cette classe en passant le nom de la classe au lieu de l'objet à l'index 0, ou en passant 'NomClass::NomMethode'

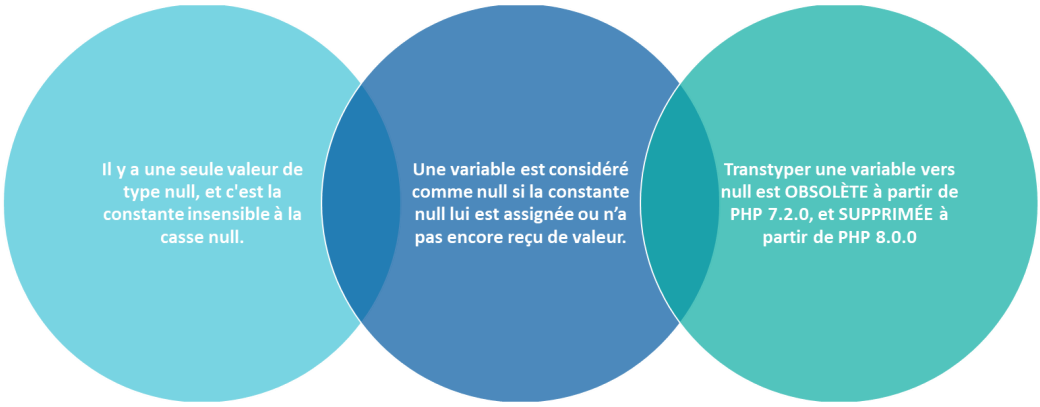
Type itérable

- Un iterable est un pseudo-type introduit en PHP 7.1. Il accepte n'importe quel tableau ou objet implémentant l'interface Traversable. (Interface permettant de détecter si une classe peut être parcourue en utilisant foreach).
- Tous les tableaux sont des itérables.
- Un tableau peut être utilisé comme argument d'une fonction qui nécessite un itérable.

Type ressource

Une ressource est une variable spéciale, contenant une référence vers une ressource externe.

Type NULL



Modification de types

4 - Instructions de sortie

Envoi vers le navigateur

Trois fonctions permettent d'envoyer du texte vers le navigateur :

- echo(string ...$expressions) : void → Affiche une chaîne de caractères.
- print(string $expression) : int → Affiche une chaîne de caractères.
- printf(string $format, mixed ...$values) : int → Affiche une chaîne de caractères formatée.

5 - Contrôles de flux et boucles

Introduction

- L'ordre de flux est l'ordre dans lequel les instructions d'un programme sont exécutées.
- Les instructions qui changent l'ordre d'exécution sont:
  - Les itérations.
  - Les instructions conditionnelles.
  - Les ruptures de séquences.

- Les instructions de contrôle sont utilisées pour contrôler l'exécution d'un programme:
  - Instructions de boucle : elles sont utilisées pour exécuter un bloc de code à x reprises.
  - Instructions de sélection : elles vous permettent d'exécuter un bloc de code spécifique dans plusieurs blocs de code en fonction de critères de sélection.
  - Instructions de saut : Ces instructions sont utilisées pour passer d'un bloc de code à un autre.

Syntaxe alternative

Pour les fonctions de contrôle if, while, for, foreach et switch. PHP propose une autre manière de rassembler des instructions à l'intérieur d'un bloc.

Le principe est de remplacer l'accolade d'ouverture par deux points (:) et l'accolade de fermeture par, respectivement, endif;, endwhile;, endfor;, endforeach;, ou endswitch;

Vous ne pouvez pas utiliser différentes syntaxes dans le même bloc de contrôle.

L'instruction if else

- Si l'expression vaut true, PHP exécutera l'instruction et si elle vaut false, l'instruction sera ignorée.
- Les instructions après le else ne sont exécutées que si l'expression du if est false.

L'instruction elseif

- L'expression elseif est exécutée seulement si le if précédent et tout autre elseif précédent sont évalués comme false, et que votre elseif est évalué à true.
- Le « elseif » et « else if » sont traités de la même façon seulement quand des accolades sont utilisées.

L'instruction switch

- L'instruction switch équivaut à une série d'instructions if.
- Un cas spécial est default. Ce cas est utilisé lorsque tous les autres cas ont échoué.

- PHP continue d'exécuter les instructions jusqu'à la fin du bloc d'instructions du switch, ou bien dès qu'il trouve l'instruction break.
- Dans une commande switch, une condition n'est évaluée qu'une fois, et le résultat est comparé à chaque case.

L’instruction match

De la même manière qu'une instruction switch, l’instruction match a une expression de sujet qui est comparée à plusieurs alternatives.

L’instruction while

PHP exécute l'instruction tant que l'expression de la boucle while est évaluée comme true. Si l'expression du while est false avant la première itération, l'instruction ne sera jamais exécutée.

L’instruction do-while

La principale différence par rapport à la boucle while est que la première itération de la boucle do-while est toujours exécutée.

L’instruction for

- expr1 : est évaluée (exécutée), au début de la boucle.
- expr2 : est évaluée, au début de chaque itération. Si l'évaluation vaut true, la boucle continue et les commandes sont exécutées. Si l'évaluation vaut false, l'exécution de la boucle s'arrête.
- expr3 : est évaluée (exécutée), à la fin de chaque itération.
- Les expressions peuvent éventuellement être laissées vides ou peuvent contenir plusieurs expressions séparées par des virgules.

L’instruction foreach

- foreach ne fonctionne que pour les tableaux et les objets.
- La forme suivante passe en revue le tableau iterable_expression. À chaque itération, la valeur de l'élément courant est assignée à $value.
- La forme suivante assignera en plus la clé de l'élément courant à la variable $key à chaque itération.

L’instruction break

- L'instruction break permet de sortir d'une structure for, foreach, while, do-while ou switch.
- break accepte un argument numérique optionnel qui vous indiquera combien de structures emboîtées doivent être interrompues.

L’instruction continue

- L'instruction continue est utilisée dans une boucle pour contourner l'instruction de l'itération actuelle et poursuivre l'exécution sous la condition évaluée, en commençant l'itération suivante.

- continue accepte un argument numérique optionnel qui vous indiquera combien de structures emboîtées doivent être interrompues.
- La structure continue s'applique aux structures switch et se comporte de la même manière que break.

6 - Formulaires simples

Formulaires HTML

Lorsqu'un formulaire est envoyé à un script PHP, toutes les variables du formulaire seront automatiquement disponibles dans le script.

Formulaires XForms

XForms est un dialecte XML permettant de créer des formulaires en ligne conçus pour être utilisés avec HTML, XHTML, WML ou SVG.

7 - Transmission de variables (GET, POST)

$_GET

- Elle donne les valeurs des informations indiquées dans l'url.
- Les informations après le point d'interrogation ? d'une URL sont en réalité des données que l'on fait transiter d'une page à une autre.
- Les paramètres sont séparé par le symbole &

$_POST et $_REQUEST

- PHP $_POST est une variable super globale PHP qui est utilisée pour collecter des données de formulaire après avoir soumis un formulaire HTML avec method="post".
- $_POST est également utilisé pour passer des variables.
- Lorsque nous envoyons des variables via un formulaire, la variable est récupérée de cette façon $_POST['Variable'] et non plus $Variable

8 - Variables d'environnement

$_SERVER

$_SERVER est un tableau contenant des informations comme les en-têtes, dossiers et chemins du script.

Liste dans https://www.php.net/manual/fr/reserved.variables.server.php

Variable prédéfinies

- $_FILES - Variable de téléchargement de fichier via HTTP:
  - $_FILES['nom_de_la_variable']['name'] : Le nom original du fichier qui provient de la machine du client
  - $_FILES['nom_de_la_variable']['type'] : Le type MIME du fichier
  - $_FILES['nom_de_la_variable']['size'] : La taille du fichier en bytes (soit 8 bits ou un octet)
  - $_FILES['nom_de_la_variable']['tmp_name'] : Le nom temporaire du fichier stocké sur le serveur
  - $_FILES['nom_de_la_variable']['error'] : Le code erreur associé à l'upload
- $_SESSION - Variables de session. Liste des fonctions dans https://www.php.net/manual/fr/ref.session.php
- $_ENV - Variables d'environnement
- $_COOKIE - Cookies HTTP

9 - Redirection entre pages

header

- header = Envoie un en-tête HTTP brut
- La syntaxe est header(string $header, bool $replace = true, int $response_code = 0): void

10 - Fonctions sur les chaines de caractères et les dates

Fonctions sur les chaînes de caractères

Référence officielle : https://www.php.net/manual/fr/ref.strings.php

Fonctions sur les Date/Heure

Référence officielle : https://www.php.net/manual/fr/ref.datetime.php

Traiter les données en PHP

1. Traitement des tableaux (simple, Associatif)

Tableaux indexés ou simples

Un tableau indexé contient des indices numériques qui indexent le contenu souhaité. Ces indexes commencent par défaut de 0 et s'incrémentent de 1 à chaque fois.

Tableaux associatifs

Un tableau associatif contient des clés qui sont des chaînes de caractères.

Constantes pré-définies

Liste : https://www.php.net/manual/fr/array.constants.php

Fonctions sur les tableaux

Liste : https://www.php.net/manual/fr/ref.array.php

Opérateurs de tableaux

Source : https://www.php.net/manual/fr/language.operators.array.php

2. Manipulation de fichier (chargement, Suppression, téléchargement)

Ouvrir un fichier

- fopen = Ouvre un fichier ou une URL
- Filename : est de la forme "protocole://"
- mode : spécifie le type d'accès désiré au flux
- Liste des protocoles et des gestionnaires supportés : https://www.php.net/manual/fr/wrappers.php
- use_include_path : paramètre optionnel peut être défini à 1 ou à true pour chercher le fichier dans l'include_path.

Paramètre mode de fopen()

Tab : Liste des modes pour la fonction fopen() en utilisant le paramètre mode. Source : https://www.php.net/manual/fr/function.fopen.php

Autre opérations sur un fichier

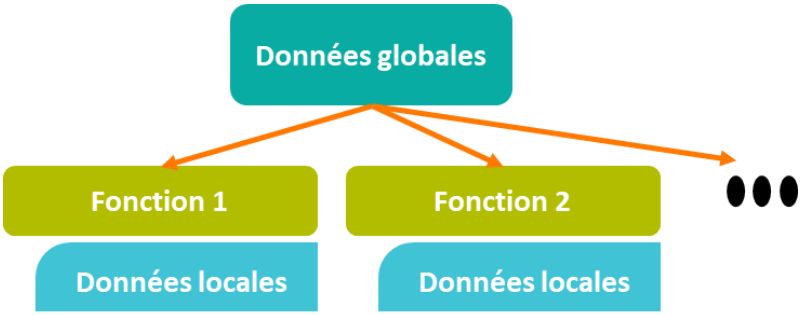
Liste : https://www.php.net/manual/fr/ref.filesystem.php

Utiliser l'orientée objet en PHP

1. Intérêt de programmer en Orienté Objet en PHP

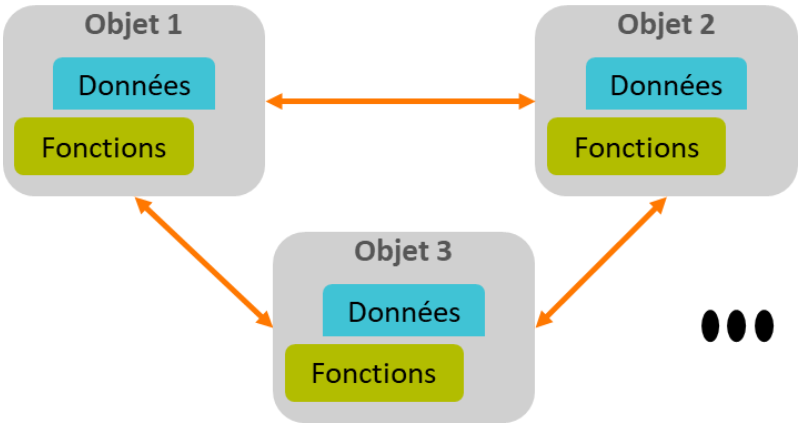
Programmation procédurale

- Le programme est divisé en procédures ou fonctions.
- Chaque fonction contient des données différentes.
- Une procédure permet d'effectuer des opérations sur les données généralement contenues dans des variables.
- Les opérations sont exécutées selon leur ordre d'écriture dans le script.

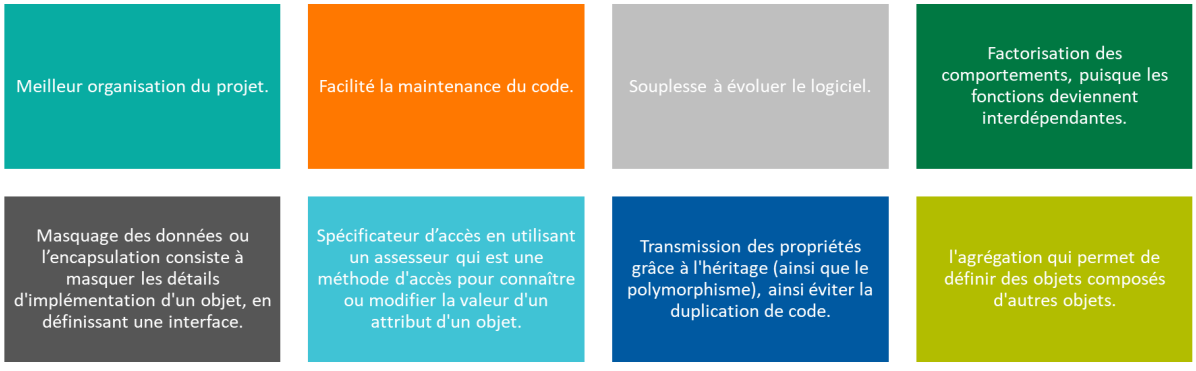


Programmation orientée objet

- Le programme est divisé en objets.
- Les objets agissent comme une seule unité.
- Un objet est une entité qui va pouvoir contenir un ensemble de fonctions et de variables.
- Regrouper des tâches précises au sein d'objets pour obtenir une nouvelle organisation du code



Avantages d'une programmation orientée objet (POO)



Les objets et les classes

Une classe :

- Est un ensemble de variables et de fonctions (attributs et méthodes).
- Contient le nom des propriétés qu'on pourra manipuler
- Les propriétés, ce sont des variables internes où on stocke des valeurs.
- Les méthodes sont des fonctions internes à la classe.
- Un objet est une instance de la classe pour pouvoir l'utiliser.

2. Application des concepts de base de la programmation orientée objet en PHP

Syntaxe de base

- Une définition de classe basique commence par le mot-clé class, suivi du nom de la classe.

Le nom de la classe :

- Peut être ^[a-zA-Z_\x80-\xff][a-zA-Z0-9_\x80-\xff]*$
- Ne doit pas être un mot réservé en PHP

Liste des mots réservé en PHP : https://www.php.net/manual/fr/reserved.php

La pseudo-variable $this est :

- Disponible lorsqu'une méthode est appelée depuis un contexte objet.
- La valeur de l'objet appelant.

- Le mot-clé new est utilisé pour créer une instance d'une classe.
- Le mot-clé extends permet d'étendre les constantes, méthodes et les propriétés à une autre class.

Déclaration des attributs

Types pour déclarer un attribut :

- **public**
- **private**
- **protected**

L'encapsulation est un principe fondamental de la POO. Il vise à masquer les attributs aux utilisateurs du code. C'est la visibilité private ou protected qui est recommandée.

Déclaration des méthodes

Types pour déclarer une méthode :

- **public**
- **private**

Les fonctionnalités d'un objet

- Le mot-clé **extends** permet de définir une classe dérivée.

- Le mot-clé **function** permet de déclarer une méthode.
- Le mot-clé **parent** permet d'accéder au parent de l'objet courant.
- **$this->** est obligatoire pour accéder aux membres de l'objet courant.
- Mot-clé **final** empêche les classes enfants de redéfinir une méthode ou constante en préfixant la définition avec final. Si la classe elle-même est définie comme finale, elle ne pourra pas être étendue.
- Une propriété peut être déclarée avec le modificateur **readonly**, qui empêche la modification de la propriété après l'initialisation.

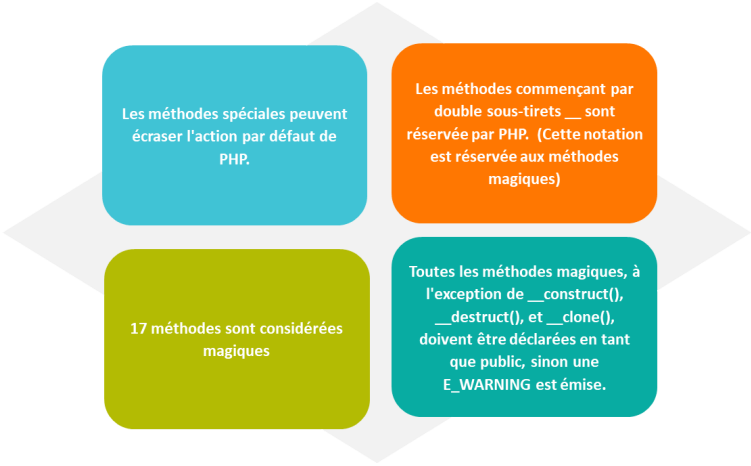
Fonctions Classes / Objets

Quelques exemples :

- __autoload
- class_alias
- class_exists
- enum_exists

3. Utilisation des méthodes magiques en PHP

Méthodes magiques



__construct()

Syntaxe : __construct(mixed ...$values = ""): void

Les classes qui possèdent une méthode constructeur appellent cette méthode à chaque création d'une nouvelle instance de l'objet.

Si vous voulez utiliser un constructeur parent, il sera nécessaire de faire appel à parent::__construct() depuis le constructeur enfant.

Un constructeur peut définir un nombre d'arguments, qui peuvent être requis, avoir un type, et peuvent avoir une valeur par défaut.

__destruct()

Syntaxe : __destruct(): void

La méthode destructeur est appelée dès qu'il n'y a plus de référence sur un objet donné.

Si vous voulez utiliser un destructeur parent, il sera nécessaire de faire appel à parent::__destruct() depuis le destructeur enfant.

Le destructeur sera appelé même si l'exécution du script est stoppée en utilisant la fonction exit().

Surcharge de propriétés

- __set()
- __get()
- __isset()
- __unset()

__sleep() et __serialize()

- __sleep()
- __serialize()

Si __serialize() et __sleep() sont tout les deux définie dans le même objet, alors seulement __serialize() sera appelé.

__wakeup() et __unserialize():

- __wakeup()
- __unserialize()

Si __unserialize() et __wakeup() sont tout les deux définie dans le même objet, alors seulement __unserialize() sera appelée.

Surcharge de méthodes

- __call()
- __callStatic()

Autres méthodes magiques

- __toString()
- __invoke()
- __set_state()
- __clone()

Références et ressources

- https://www.php.net/manual/fr/
- https://openclassrooms.com/fr/courses

- https://www.phpfacile.com
- https://php.developpez.com/