**PHP\_Objet**

SELF :: = pour utiliser la constante dans ma classe.

Différence entre fonction et procédure. La procédure ne retourne rien.

Implémenter une méthode = y mettre du code.

Classe abstraite pour avoir de la rigueur dans le code.

 if ($this->score) {

            $info .= 'Le score de la partie est : ' . $this->score . '. ';

        } else {

            $info .= 'Le score de la partie n\'est pas défini. ';

        }

        if ($this->gagner) {

            $info .= 'La partie est gagnée. ';

        } else {

            $info .= 'La partie n\'est pas gagnée. ';

        }

Procédures de création des classes

1. Créer les classes
2. Créer les attributs simples (entier, date, etc)
3. Créer les attributs et type objets
4. Créer les accesseurs get et & set
5. Créer les constructeurs
6. Créer les méthodes du diagramme UML
7. Déclarer les fonctionalités

Avec PHP 8

Avnat chaque déclaration de classe :

Déclarer le namespace : namespace App\models (=>bien structurer les dossiers du code avant).

Pour appeler les fichiers dans index.php, utiliser : use App\Models\Joueur ;

Installer xdebug :

* Créer un fichier phpinfo.php dans le dossier local
* Executer phpinfo()
* Tout copier
* Ouvrir le lien <https://xdebug.org/wizard>
* Suivre les instructions

**Instructions**

1. Download [php\_xdebug-3.2.0-8.1-vs16-x86\_64.dll](https://xdebug.org/files/php_xdebug-3.2.0-8.1-vs16-x86_64.dll)
2. Move the downloaded file to C:\xampp\php\ext, and rename it to php\_xdebug.dll
3. Update C:\xampp\php\php.ini and add the line:  
   zend\_extension = xdebug
4. Restart PHP's built-in HTTP server (php -S)

Methode static

Le fait de déclarer des propriétés ou des méthodes comme statiques vous permet d'y accéder sans avoir besoin d'instancier la classe.

:: [opérateur de résolution de portée](https://www.php.net/manual/fr/language.oop5.paamayim-nekudotayim.php)