

Programmer en PHP - Créer un projet Console (CLI)

Présentation

PHP est un langage de script largement utilisé pour le développement web côté serveur. Cependant, PHP peut aussi être exécuté en mode console (CLI – Command Line Interface), utile pour apprendre la syntaxe de base, tester des algorithmes ou automatiser des tâches.

Outils

- Télécharger et installer PHP (version 8.x conseillée).
- Vérifier l'installation avec :
 - `php --version`

Prérequis

- Connaître les bases des types de données (nombres, chaînes, booléens).
- Comprendre l'exécution séquentielle (instruction après instruction).

Créer un projet

- Ouvrir un éditeur de texte.
- Créer un nouveau fichier PHP.
- Sauvegarder le fichier dans un dossier dédié.

Exécuter un script en console

- Ouvrir un terminal.
- Naviguer jusqu'au dossier où est enregistré le fichier : `cd chemin/vers/le/dossier`
- Lancer l'exécution : `php mon_script.php`

Affichage en Console

```
<?php
echo "Bonjour !\n"; // Affiche avec retour à la ligne
echo "Entrez votre prénom : "; // Sans retour automatique
?>
```

- `\n` représente un retour à la ligne en PHP.
- `echo` est utilisé pour afficher du texte.

Programmer en PHP - Créer un projet Console (CLI)

Types de Données

int	42, -15, 0
float	3.14, -0.99
string	"Bonjour", "1234"
bool	true, false
array	[1, 2, 3]
object	(instances de classes)

Déclaration et Affectation de Variables

```
<?php
$age = 18;
$nom = "Alice";
$majeur = true;
$taille = 1.75;
?>
```

Règles : les variables commencent toujours par \$.

Opérations sur Variables

Opération	Symbole	Exemple
Addition	+	\$a + \$b
Soustraction	-	\$a - \$b
Multiplication	*	\$a * \$b
Division	/	\$a / \$b
Modulo	%	\$a % \$b
Incrémenter	++	\$a++
Décrémenter	--	\$a--

Programmer en PHP - Créer un projet Console (CLI)

Lecture clavier (Input utilisateur)

```
<?php
echo "Entrez votre prénom : ";
$prenom = trim(fgets(STDIN));
echo "Bonjour, $prenom !\n";
?>
```

- fgets(STDIN) lit une ligne.
- trim() enlève les retours à la ligne ou espaces.

Conditions

```
<?php
if ($age >= 18) {
    echo "Vous êtes majeur.\n";
}
?>
```

```
<?php
if ($age >= 18) {
    echo "Majeur\n";
} else {
    echo "Mineur\n";
}
?>
```

```
<?php
if ($age < 12) {
    echo "Enfant\n";
} elseif ($age < 18) {
    echo "Adolescent\n";
} else {
    echo "Adulte\n";
}
?>
```

Modèle de bloc try-catch en PHP

```
<?php
try {
    // Code à exécuter qui pourrait provoquer une exception
} catch (Exception $e) {
    // Traitement de l'exception
    echo 'Une erreur est survenue : '. $e->getMessage();
}
?>
```

Programmer en PHP - Créer un projet Console (CLI)

Switch

```
<?php
echo "Entrez une note entre 1 et 5 : ";
$note = intval(trim(fgets(STDIN)));

switch ($note) {
    case 1:
        echo "Très mauvais\n";
        break;
    case 5:
        echo "Excellent !\n";
        break;
    default:
        echo "Note correcte\n";
        break;
}
?>
```

Boucle for (connue)

```
<?php
for ($i = 0; $i < 5; $i++) {
    echo "i vaut : $i\n";
}
?>
```

Boucle while (tant que)

```
<?php
$compteur = 0;
while ($compteur < 3) {
    echo "Compteur : $compteur\n";
    $compteur++;
}
?>
```

Boucle do-while (au moins une fois)

```
<?php
do {
    echo "Tapez un nombre supérieur à 10 : ";
    $nb = intval(trim(fgets(STDIN)));
} while ($nb <= 10);
?>
```

Programmer en PHP - Créer un projet Console (CLI)

Définir une fonction

```
<?php
function direBonjour() {
    echo "Bonjour !\n";
}
?>
```

Appeler une fonction

```
<?php
direBonjour();
?>
```

Fonction avec Retour

```
<?php
function additionner($a, $b) {
    return $a + $b;
}

echo additionner(5, 3) . "\n"; // 8
?>
```

Déclaration et initialisation d'un tableau

```
<?php
$notes = [12, 15, 8, 19, 14];
?>
```

Accès et modification

```
<?php
echo $notes[0]; // 12
$notes[2] = 10;
?>
```

Parcourir avec for

```
<?php
for ($i = 0; $i < count($notes); $i++) {
    echo "Note " . ($i+1) . " : " . $notes[$i] . "\n";
}
?>
```

Parcourir avec foreach

```
<?php
foreach ($notes as $note) {
    echo "Note : $note\n";
}
?>
```

Programmer en PHP - Créer un projet Console (CLI)

Collections (Listes dynamiques)

```
<?php
$fruits = [];
$fruits[] = "Pomme";
$fruits[] = "Banane";
$fruits[] = "Cerise";
?>
```

Suppression

```
<?php
unset($fruits[1]); // Supprime "Banane"
?>
```

Vérification d'existence

```
<?php
if (in_array("Pomme", $fruits)) {
    echo "Pomme est dans la liste\n";
}
?>
```

Classe et Objet

```
<?php
class Voiture {
    public $marque;
    public $modele;
}
$v1 = new Voiture();
$v1->marque = "Renault";
$v1->modele = "Clio";
?>
```

Méthodes

```
<?php
class Voiture {
    public function klaxonner() {
        echo "Bip Bip !\n";
    }
}
$v1 = new Voiture();
$v1->klaxonner();
?>
```

Programmer en PHP - Créer un projet Console (CLI)

Constructeur

```
<?php
class Voiture {
    public $marque;
    public function __construct($marque) {
        $this->marque = $marque;
    }
}
$v1 = new Voiture("Peugeot");
?>
```

Concepts avancés

Concept	Exemple	Description
Héritage	class Moto extends Vehicule	Reprendre le code d'une classe
Abstraction	abstract class Animal	Classe modèle non instanciable
Polymorphisme	\$a = new Chien(); \$a->crier();	Appel dynamique du comportement
Interface	interface IVehicule	Définir des obligations

Exemple complet

```
<?php
interface IVehicule {
    public function demarrer();
}

class Moto implements IVehicule {
    public function demarrer() {
        echo "La moto démarre !\n";
    }
}

$m = new Moto();
$m->demarrer();
?>
```