

### **PUBLIC :**

ICAN 3web1 - option code

### **DURÉE :**

51 heures

### **PRÉ-REQUIS :**

* savoir réaliser un CRUD en PHP,
* avoir les bases en algorithmie

### **FORMATEUR :**

Yoann Coualan

Développeur web fullstack et Directeur général de Jedy Formation

<https://www.linkedin.com/in/yoanncoualan/>

Quelques exemples de projets : <https://www.yoanncoualan.com/>

### **MÉTHODE PÉDAGOGIQUE :**

Les cours se déroulent toujours de la même manière :

* J’explique au vidéoprojecteur une notion ou comment faire quelque chose, puis les étudiants reproduisent,
* Une fois que je leur ai montré suffisament de nouvelles choses, je les met sur un exercice reprenant tout ce qu’on vient de voir.

On répète les phases 1 et 2 tant qu’il leur reste des choses à apprendre (sûrement durant tout le module) et à chaque début de cours, on fait des rappels sur ce qui à été vu lors du cours précédent.

## APRÈS CE MODULE LES ÉTUDIANTS SAURONT :

* utiliser GIT,
* développer en PHP Orienté Objet,
* Utilisation d’une structure MVC,
* utiliser le framework Symfony4,
* utiliser Twig,
* utiliser une méthode REST,
* Sécuriser et optimiser leur code.

## OUVRAGES DE RÉFÉRENCE (LIVRES, ARTICLES, REVUES, SITES WEB)

* <https://git-scm.com/>
* <https://secure.php.net/>
* <http://symfony.com/>
* <http://twig.sensiolabs.org/>

### **MODALITÉ DE VALIDATION DES ACQUIS :**

Les étudiants seront évalués tout au long du module à l’aide d’exercices ramassés et notés.

Ils seront également notés en fin de module à l’aide d’un partiel, idéalement de 3 heures.

Les notes seront rendues détaillées avec le barème et les commentaires.

## **PROGRAMME :**

Le niveau en PHP est meilleur que l’année dernière pour le moment, du coup il y a une partie du cours que j’ai pu supprimer. En revanche je vais passer plus de temps avec eux pour faire des révisions en début d’année.

L’année dernière les étudiants utilisaient des notions par habitude, sans savoir ce qu’il se passait vraiment derrière, ce qui a posé des problèmes une fois sur Symfony.

**Cour 1 à 3 :**

Rappels du PHP et PDO à l’aide d’exercices.

**Cour 4 :**

Présentation de GIT, à quoi ça sert ?

Comment fonctionne GIT

Créer son premier répertoire GIT

Mettre en commun son travail (avec qui le partager, comment le partager de manière sécurisé, comment ne pas effacer ce que les autres font, etc ...) Entrainement avec des exercices de mise en pratique.

**Cour 5 :**

Comment avoir l’historique des mises à jour (en ligne puis en local) Comment revenir en arrière

Les branches

Bonnes pratiques pour collaborer sans risque de perte de données Entrainement avec des exercices de mise en pratique.

**Cour 6 :**

Présentation du PHP Orienté Objet : qu’est ce que c’est ? Avantages / inconvénients

Initiation au POO : les class, les méthodes, créer un objet, lui affecter des données, lire un objet, les setters et getters Entrainement avec des exercices de mise en pratique.

**Cour 7 :**

Le constructeur

Le scope

L’héritage

Entrainement avec des exercices de mise en pratique. Les constantes

L’opérateur de résolution de portée

Le type static

Entrainement avec des exercices de mise en pratique.

**Cour 8 :**

Exercice réalisé ensemble pour mettre en pratique le POO dans un cas concret.  Mise en avant des différences entre le procédural et l’objet.

**Cour 9 :**

Présentation de Symfony : pourquoi Symfony ? Avantages / inconvénients Initiation à Symfony : installation, présentation des répertoires et des composants Qu’est ce qu’un Bundle ?

Présentation du modèle MVC.

**Cour 10 :**

Créer un Bundle

Créer une première page et lui envoyer des données Présentation du Twig, afficher ses données Entrainement avec des exercices de mise en pratique.

**Cour 11 :**

Le système de template et d’héritage

Entrainement avec des exercices de mise en pratique. Paramétrer son application

Créer des entités

**Cour 12 :**

Récupérer une entité, envoyer le résultat à la vue et l’afficher Les boucles et conditions en Twig

Entrainement avec des exercices de mise en pratique.

La traduction

Entrainement avec des exercices de mise en pratique.

**Cour 13 :**

Créer son premier formulaire : préparer le formulaire, l’envoyer à la vue, l’afficher, détecter son envois, vérifier les données reçues, les sauvegarder, prévenir l’internaute du succès ou de l’échec.

Entrainement avec des exercices de mise en pratique.

**Cour 14 :**

Personnaliser son formulaire : design, labels, messages d’erreurs Entrainement avec des exercices de mise en pratique.

Ajouter automatiquement des données

Joindre deux entités

**Cour 15 :**

Uploader des médias

Entrainement avec des exercices de mise en pratique. Supprimer un élément en base de données Automatiser la suppression d’éléments annexes Entrainement avec des exercices de mise en pratique.

**Cours 16 :**

L’AJAX avec Symphony

Entrainement avec des exercices de mise en pratique. Les utilisateurs

Sécuriser une page

**Cour 17 :**

La méthode REST : qu’est ce que c’est ? Avantages / inconvénients Entrainement avec des exercices de mise en pratique.

Si les étudiants avancent aussi vite que l’année dernière, on pourra également aborder les API : Qu’est-ce qu’une API ? Créer sa propre API et utiliser une API existante.

Les API font partie du quotidien des développeurs back, elles permettent d’utiliser de manière sécurisé des services tiers et permettent une totale automatisation des taches et un important gain de temps.