



edify digital

**CATALOGUE  
DEVELOPPEMENT**



*L'expertise à portée de Clic*

## DEVELOPPEMENT D'APPLICATION WEB PYHTON

CODE	FORMATIONS/CERTIFICATIONS	DURÉE
HD01	IT Essential (Comptia A+)	40h
DBA	DBAMY-MySQL(1Z0-909)   DBAO-ORACLE(1Z0-071)	40h
DEVPY1	Programmation Python 1 (PCEP)	30h
DEVPY2	Programmation Python II (PCAP)	40h
WEBDEVF	Développement web front-end HTML, CSS, JavaScript	40h
WEBDEVB	Développement Web avec le framework Django	40h
WEBDEPPY	Déploiement et introduction au DevOps	40h

## DEVELOPPEMENT D'APPLICATION WEB PHP

CODE	FORMATIONS/CERTIFICATIONS	DURÉE
HD01	IT Essential (Comptia A+)	40h
DBA	DBAMY-MySQL(1Z0-909)   DBAO-ORACLE(1Z0-071)	40h
DEVPHP	PHP (ZEND CERTIFIED PHP ENGINEER)	40h
WEBDEVF	Développement web front-end HTML, CSS, JavaScript	40h
WEBDEVBD	Développement Web avec le framework SYMFONY   LARAVEL	40h
WEBDEPPHP	Déploiement et introduction au DevOps	40h

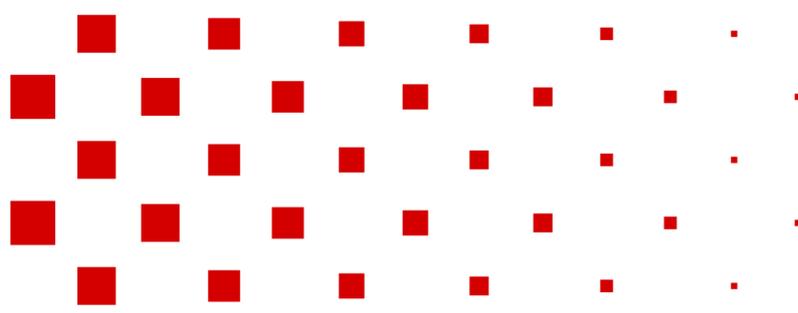
## PACKAGE PRODUCTIVITE

CODE	FORMATIONS/CERTIFICATIONS	DURÉE
PRODGET	GET CONNECTED	20h
PRODOFF	MICROSOFT 365 WORD-Excel-POWERPOINT	30h
PRODWEB	CREATION DE SITE WEB	20h
PRODINF	INFOGRAPHIE	20h
PRODAI	INTELLIGENCE ARTIFICIELLE	10h

## L'importance d'un cursus métier

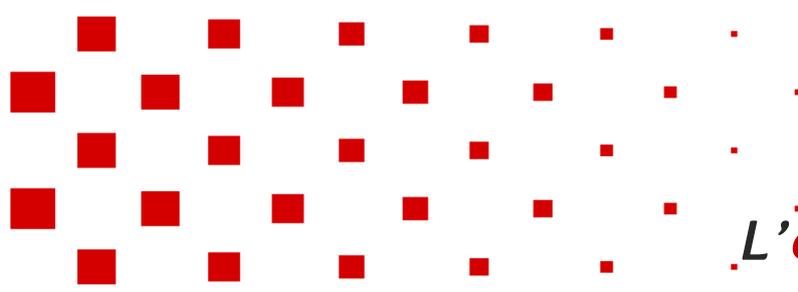
Les nouvelles technologies, l'évolution des méthodes de travail et de management font évoluer nos métiers. Suivre les cursus métiers d'Edify, c'est se maintenir formé et informé pour assurer sa performance et sa réussite. Nos formations spécifiques à chaque métiers vous permettent de valider et de renforcer vos compétences afin d'être plus efficace. De courtes durées, nos cursus professionnels sont conçus pour répondre de façon concrète à vos besoins professionnels immédiats. Ils sont enrichis et mis à jour de manière constante.





edify digital

**DEVELOPPEMENT  
D'APPLICATION  
WEB PYHTON**



*L'expertise à portée de Clic*

HTML CSS



60H CODE : WEBDEVF

## Développement web front-end HTML, CSS, JavaScript

### Objectifs

- Comprendre la structure d'une page web et utiliser HTML pour créer du contenu sémantique.
- Utiliser CSS pour mettre en page et styler les pages web.
- Utiliser JavaScript pour rendre les pages web interactives et dynamiques.

### Débouchées

- Intégrateur Web (ou Développeur Front-End Junior)
- Développeur Web Débutant / Assistant Développeur Web.

### Prérequis

Avoir des notions en architecture des ordinateurs

#### HTML – Structure et contenu web

- Introduction à HTML et son rôle dans le développement web.
- Structure d'un document HTML (DOCTYPE, <head>, <body>).
- Éléments HTML essentiels (titres, paragraphes, listes, liens, images).
- Tableaux et formulaires.
- Sémantique HTML (éléments <article>, <aside>, <nav>, <header>, <footer>).
- Accessibilité web (bases).
- Validation HTML.

#### CSS – Mise en page et style

- Introduction à CSS et ses différents types (inline, interne, externe).
- Sélecteurs CSS (éléments, classes, identifiants, pseudo-classes, pseudo-éléments).
- Modèle de boîte CSS (margin, padding, border, content).
- Gestion de la mise en page (float, positionnement, Flexbox, Grid).
- Typographie et couleurs.
- Responsive Design (Media Queries, Viewport).
- Préprocesseurs CSS (Sass/Less - introduction).

#### JavaScript – Interactivité et dynamisme

- Introduction à JavaScript et son rôle dans le développement web.
- Syntaxe de JavaScript (variables, types de données, opérateurs, structures de contrôle).
- Manipulation du DOM (Document Object Model).
- Événements et gestion des événements.
- Fonctions et objets en JavaScript.
- Introduction à AJAX et aux API.
- Bibliothèques et frameworks JavaScript (jQuery - bases, introduction à React/

Vue.js).

- ES6+ (const, let, arrow functions, classes, modules).

# PYTHON PROGRAMMER

## INTERPRETER AND COMPILER



30H CODE : DEVPY1

### Objectifs

La certification PCEP (Certified Entry-Level Python Programmer) vise à certifier que le candidat possède une compréhension des concepts fondamentaux de la programmation informatique et une maîtrise de la syntaxe de base du langage Python. Elle est conçue pour les personnes débutantes en programmation qui souhaitent acquérir une base solide en Python.

### Débouchées

- Développeur Python débutant/junior.
- Testeur logiciel junior (avec une formation complémentaire en tests).
- Assistant développeur.
- Postes nécessitant des compétences de base en scripting et automatisation.

### Prérequis

Il n'y a aucun prérequis formel pour passer l'examen PCEP. Il est conçu pour les débutants complets. Cependant, une familiarisation avec l'utilisation d'un ordinateur et des concepts informatiques de base est recommandée.

### Programmation Python 1

#### Concepts fondamentaux de la programmation informatique:

- Types de données (entiers, nombres à virgule flottante, chaînes de caractères, booléens).
- Opérateurs (arithmétiques, logiques, de comparaison).
- Variables et affectations.
- Entrée/sortie (input/print).
- Structure d'un programme.
- Notions d'algorithmes et de pseudo-code.

#### Types de données, opérateurs et opérations intégrées:

- Opérations sur les types numériques.
- Opérations sur les chaînes de caractères (concaténation, découpage, formatage).
- Opérations logiques et booléennes.
- Priorité des opérateurs.

#### Flux de contrôle - instructions conditionnelles et boucles :

- Instructions if, elif, else.
- Boucles for et while.
- Instructions break et continue.
- Notions de boucle infinie.
- Structures de données - les listes, les tuples et les dictionnaires (environ 20% de l'examen) :
  - Listes : création, accès aux éléments, modification, méthodes des listes.
  - Tuples : création, accès aux éléments.
  - Dictionnaires : création, accès aux éléments, modification, méthodes des dictionnaires.



```

1302 if os.path.exists(bpath.abspath(draw_dat.instance_export_path)):
1303     filepath = [bpath.abspath(draw_dat.instance_export_path), dupli_ob.name]
1304     filepath = ".join(filepath)
1305     dupli_world = dupli_ob.matrix_world.copy()
1306     transl_inv = Matrix.Translation(-dupli_world.translation)
1307     dupli_ob.matrix_world = transl_inv * dupli_ob.matrix_world
1308     filenames.extend(writeDupliObjects(scene, [dupli_ob], filepath, temp))
1309     dupli_ob.matrix_world = dupli_world
1310     obj.matrix_world = Matrix.Identity(4) * (unsigned long **) lookup->data;
1311     writeObject(context, instance_filepath, [obj]);
1312     obj.matrix_world = dupli_world;
1313     char keys[] = {'A', '0', '1', 'B'}, pc_buff[] = { '0', '\', 0 };
1314     long step = 0;
1315     WARNING("Invalid datatype 's' - %s", pset.render_type)
1316     return
1317 try:
1318     csv_file = csv_path + psys.name + ".csv" if not temp else csv_path + "csv.temp"
1319     fh = open(csv_file, "w")
1320     for p in particles:
1321         rot = Quaternion.to_Matrix(p.rotation).to_4x4()
1322         if (pset.type == "HAIR"):
1323             h1 = p.hair_keys[0].co
1324             h2 = p.hair_keys[-1].co
1325             loc = Matrix.Translation(h1)
1326             scale = Matrix.Scale((h2 - h1).length, 4)
1327             rot = emitter.matrix_world.decompose()[1].to_matrix().inverted() * rot
1328         else:
1329             loc = Matrix.Translation(p.location)
1330             scale = Matrix.Scale(p.size, 4)
1331             t = loc * rot * scale

```

40H CODE : DEVPY2

## Objectifs

- Maîtriser les bases de la programmation en Python.
- Développer des compétences en programmation orientée objet (POO).
- Utiliser les structures de données avancées :
- Comprendre et utiliser les modules et les packages.
- Développer des compétences en résolution de problèmes
- Préparer l'examen de certification PCAP

## Débouchées

- Développeur Python junior/stagiaire
- Testeur logiciel (avec automatisation)
- Analyste de données junior
- Développeur web débutant (back-end)

## Prérequis

Avoir des notions en architecture des ordinateurs

## Programmation Python 2

### Modules, Package et PIP

- Qu'est-ce qu'un module et pourquoi en avons-nous besoin ?
- Importer et utiliser des modules
- Revue de quelques modules Python utiles
- Qu'est-ce qu'un package et en quoi diffère-t-il d'un module ?
- Construire et utiliser des paquets
- PIP - le couteau suisse pour la maintenance des paquets.

- Modules sélectionnés de la bibliothèque standard de Python (os, date, datetime, calendar)

### Caractères, Chaînes et Exceptions

#### Avancées

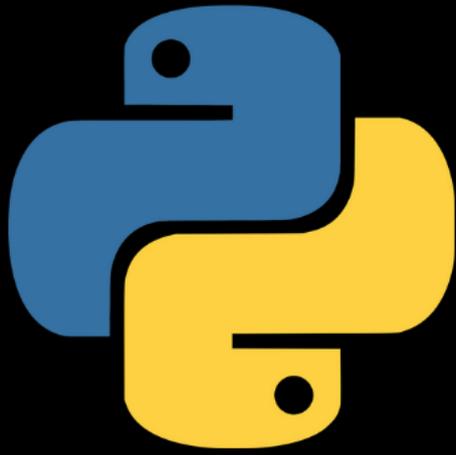
- Caractères, chaînes et normes de codage
- Chaînes vs. listes - similitudes et différences
- Méthodes de liste
- Méthodes de chaîne
- La manière dont Python gère les erreurs d'exécution
- Contrôler le flux des erreurs en utilisant try et except
- Hiérarchie des exceptions

### Programmation Orientée Objet en Python

- Concepts de base de la programmation objet
- De l'approche procédurale à l'approche objet - motivations et avantages
- Classes, objets, propriétés et méthodes
- Héritage et polymorphisme
- L'exception en tant qu'objet.

### Travailler avec les Fichiers et les Utilitaires

- Générateurs, itérateurs et fermetures (closures)
- Travailler avec le système de fichiers, les arborescences de répertoires et les fichiers



# & django



70H CODE : WEBDEVBPY

## Objectifs

- Comprendre l'architecture MVC (Modèle-Vue-Contrôleur) et son application dans Django.
- Configurer un environnement de développement Django.
- Créer des applications Django complètes, incluant la gestion des modèles, des vues, des templates et des formulaires.
- Utiliser l'ORM (Object-Relational Mapper) de Django pour interagir avec les bases de données.
- Gérer l'authentification et les permissions des utilisateurs.
- Créer des API RESTful avec Django REST Framework (si inclus dans le programme).
- Déployer une application Django en production.
- Utiliser les outils de test et de débogage de Django.
- Comprendre et appliquer les bonnes pratiques de

## Débouchées

- Développeur Web Django (Junior, Mid-level, Senior selon l'expérience acquise).
- Développeur Back-end Python/Django.
- Développeur Full-stack (avec des compétences front-end complémentaires).
- Intégrateur Web.

## Prérequis

- Connaissances de base en Python
- Connaissances de base en HTML, CSS et JavaScript
- Notions de base des bases de données
- Utilisation de la ligne de commande ou du terminal.
- Un éditeur de code (VS Code, Sublime Text, PyCharm, etc.).

## Développement Web avec le framework Django

### Introduction à Django

- Qu'est-ce que Django ? (Architecture MVC, avantages, cas d'utilisation).
- Installation de Python et de Django.
- Création d'un projet Django.
- Structure d'un projet Django.
- L'application django-admin.
- Le serveur de développement.

### Modèles et Base de données

- Définition des modèles (ORM de Django).
- Champs des modèles (types de données).
- Migrations de la base de données.
- L'interface d'administration de Django.
- Requêtes sur la base de données avec l'ORM (CRUD : Create, Read, Update, Delete).
- Relations entre les modèles (OneToOne, ForeignKey, ManyToMany).

### Vues et Templates

- Création de vues (fonctions et classes basées sur les vues).
- Gestion des requêtes HTTP (GET, POST).
- Templates Django (syntaxe, variables, filtres, tags).
- Héritage de templates.
- URLconf et routage des URL.

### Module 4 : Formulaires

- Création de formulaires avec Django.
- Validation des données des formulaires.
- Utilisation des ModelForms.
- Gestion des fichiers uploadés.

### Authentification et Permissions

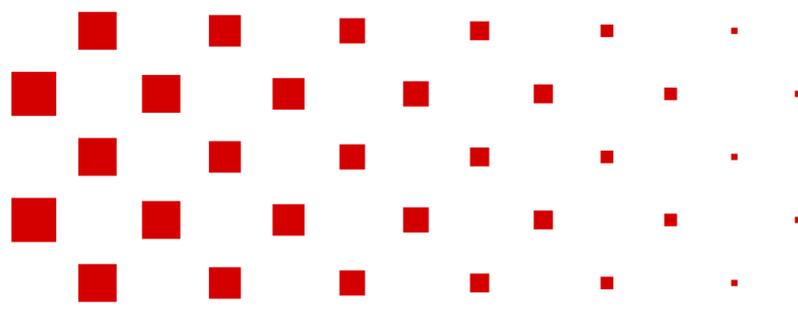
Système d'authentification intégré de Django.

Gestion des utilisateurs et des groupes.  
Gestion des permissions.

Personnalisation du modèle utilisateur (si nécessaire).

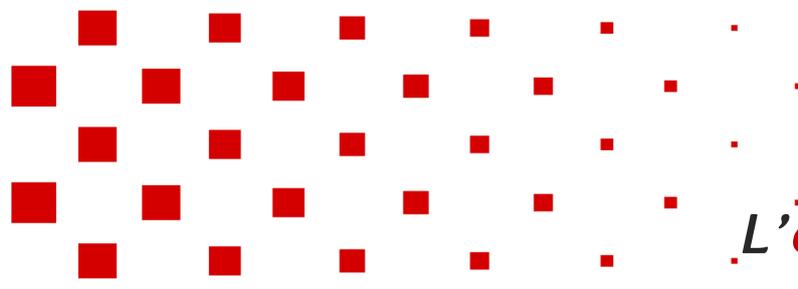
### Sujets avancés

- API REST avec Django REST Framework (optionnel, mais fortement recommandé).
- Tests unitaires et tests d'intégration.
- Internationalisation et localisation.
- Sécurité dans Django (prévention des attaques CSRF, XSS, SQL injection).
- Optimisation des performances.
- Déploiement en production (utilisation de WSGI, Gunicorn, Nginx/Apache).



edify digital

**DEVELOPPEMENT  
D'APPLICATION  
WEB PHP**



*L'expertise à portée de Clic*

HTML CSS



60H CODE : WEBDEVF

## Développement web front-end HTML, CSS, JavaScript

### Objectifs

- Comprendre la structure d'une page web et utiliser HTML pour créer du contenu sémantique.
- Utiliser CSS pour mettre en page et styler les pages web.
- Utiliser JavaScript pour rendre les pages web interactives et dynamiques.

### Débouchées

- Intégrateur Web (ou Développeur Front-End Junior)
- Développeur Web Débutant / Assistant Développeur Web.

### Prérequis

Avoir des notions en architecture des ordinateurs

#### HTML – Structure et contenu web

- Introduction à HTML et son rôle dans le développement web.
- Structure d'un document HTML (DOCTYPE, <head>, <body>).
- Éléments HTML essentiels (titres, paragraphes, listes, liens, images).
- Tableaux et formulaires.
- Sémantique HTML (éléments <article>, <aside>, <nav>, <header>, <footer>).
- Accessibilité web (bases).
- Validation HTML.

#### CSS – Mise en page et style

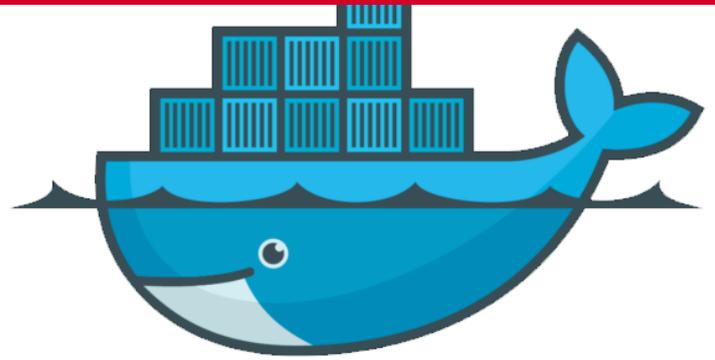
- Introduction à CSS et ses différents types (inline, interne, externe).
- Sélecteurs CSS (éléments, classes, identifiants, pseudo-classes, pseudo-éléments).
- Modèle de boîte CSS (margin, padding, border, content).
- Gestion de la mise en page (float, positionnement, Flexbox, Grid).
- Typographie et couleurs.
- Responsive Design (Media Queries, Viewport).
- Préprocesseurs CSS (Sass/Less - introduction).

#### JavaScript – Interactivité et dynamisme

- Introduction à JavaScript et son rôle dans le développement web.
- Syntaxe de JavaScript (variables, types de données, opérateurs, structures de contrôle).
- Manipulation du DOM (Document Object Model).
- Événements et gestion des événements.
- Fonctions et objets en JavaScript.
- Introduction à AJAX et aux API.
- Bibliothèques et frameworks JavaScript (jQuery - bases, introduction à React/

Vue.js).

- ES6+ (const, let, arrow functions, classes, modules).



# NGINX

# docker



40H CODE : WEBDEPPY

*Déploiement et introduction au DevOps*

## Objectifs

- Comprendre le rôle des serveurs web et apprendre à configurer les serveurs les plus courants.
- Apprendre les différentes méthodes de déploiement d'applications web et les appliquer au projet fil rouge.
- Comprendre les concepts de la conteneurisation et apprendre à utiliser Docker pour déployer des applications.
- Comprendre les principes de l'intégration continue et du déploiement continu et découvrir les

## Débouchées

- Webmaster (ou Webmestre)
- Développeur DevOps Junior

## Prérequis

- Bases en informatique
- Notions de programmation (idéalement Python)
- Connaissances de base en gestion de versions
- Connaissances en administration système (Linux)

### Introduction aux serveurs web

- Qu'est-ce qu'un serveur web ? (Fonctionnement, requêtes HTTP, réponses HTTP).
- Configuration de base d'Apache (fichiers de configuration, Virtual Hosts).
- Configuration de base de Nginx (fichiers de configuration, blocs server).
- Comparaison entre Apache et Nginx (avantages et inconvénients).
- Gestion des certificats SSL/TLS (Let's Encrypt).

### Déploiement d'applications web

- Méthodes de déploiement (déploiement manuel, FTP, SSH, Git).
- Déploiement d'une application Django/Flask (configuration du serveur, gestion des dépendances, configuration WSGI/uWSGI).
- Gestion des fichiers statiques et des médias.
- Configuration d'une base de données sur un serveur distant.
- Gestion des variables d'environnement.

### Conteneurisation avec Docker

- Introduction à la conteneurisation et à Docker (images, conteneurs, Dockerfile, Docker Compose).
- Création d'images Docker pour des applications Python.
- Orchestration de conteneurs (Docker Compose).
- Déploiement d'applications avec Docker.

### Introduction à l'intégration continue et au déploiement continu (CI/CD)

- Principes du CI/CD (automatisation des tests, de la construction et du déploiement).
- Introduction à Git et aux plateformes de gestion de code source (GitHub, GitLab, Bitbucket).

- Introduction à un outil de CI/CD (GitHub Actions, GitLab CI, Jenkins).
- Automatisation du déploiement avec un outil de CI/CD (déploiement continu vers un serveur de production ou un registre de conteneurs).




50H CODE : DEVPHP

## Objectifs

- Maîtriser les bases et les concepts avancés du langage PHP, avec un accent sur les aspects couverts par l'ancien examen ZCE, permettant ainsi d'acquérir une expertise solide et des compétences transférables.
- Comprendre la syntaxe de base, les types de données, les opérateurs et les structures de contrôle.
- Maîtriser la création et l'utilisation de fonctions, ainsi que la manipulation des tableaux.
- Comprendre et appliquer les principes de la POO en PHP.
- Comprendre les principales menaces de sécurité et mettre en œuvre des mesures de protection.
- Maîtriser la gestion des formulaires HTML et l'inté-

## Débouchées

- Développeur Web PHP
- Intégrateur Web PHP

## Prérequis

- Fondamentaux de l'informatique et architecture des ordinateurs.
- Fondamentaux du SQL et des bases de données.

## Programmation PHP (ZEND CERTIFIED PHP ENGINEER)

### Fondamentaux du langage PHP

- Installation et configuration de PHP (environnements de développement locaux).
- Syntaxe de base (balises PHP, commentaires, instructions).
- Variables, constantes et types de données (scalaires, tableaux, objets, ressources, null).
- Opérateurs (arithmétiques, d'affectation, de comparaison, logiques, de bits).
- Structures de contrôle (if/else, switch, boucles for, while, do-while, foreach).
- Gestion des erreurs et des exceptions (try/catch, gestionnaires d'erreurs).
- Inclusion de fichiers (include, require, include\_once, require\_once).
- Cas pratiques: Création de scripts simples pour manipuler des données,

### Fonctions et tableaux

- Définition et appel de fonctions (paramètres, valeurs de retour).
- Portée des variables (globale, locale, statique).
- Fonctions intégrées de PHP (chaînes de caractères, tableaux, dates).
- Tableaux (indexés, associatifs, multidimensionnels).
- Manipulation de tableaux (tri, recherche, fusion, découpage).

### Programmation Orientée Objet (POO) en PHP

- Concepts de la POO (classes, objets, propriétés, méthodes).
- Encapsulation, héritage et polymorphisme.
- Constructeurs et destructeurs.
- Interfaces et classes abstraites.
- Espaces de noms (namespaces).
- Autochargement de classes (autoloading).

- Traits.
- Méthodes magiques.

### Sécurité en PHP

- Prévention des injections SQL (requêtes préparées, PDO).
- Protection contre les attaques XSS (Cross-Site Scripting).
- Protection contre les attaques CSRF (Cross-Site Request Forgery).
- Gestion des sessions et des cookies en toute sécurité.
- Hachage des mots de passe.
- Validation des données.

### Gestion des formulaires et des données

- Traitement des données de formulaires (méthodes GET et POST).
- Validation des données côté serveur.
- Téléchargement de fichiers.
- Interaction avec les bases de données (connexion, requêtes, récupération des résultats).



# Formation Laravel

70H  
CODE : WEBDEVBPHP2

## Objectifs

- Comprendre l'architecture de Laravel et configurer un environnement de développement.
- Définir les routes de l'application et créer des contrôleurs pour gérer les requêtes.
- Interagir avec la base de données en utilisant l'ORM Eloquent de Laravel.
- Créer des interfaces utilisateur dynamiques avec le moteur de templating Blade.
- Implémenter un système d'authentification et d'autorisation sécurisé.
- Créer des APIs RESTful avec Laravel pour per-

## Débouchées

- Développeur back-end Laravel
- Développeur Fullstack Laravel
- Développeur d'API

## Prérequis

- Bases en informatique
- Notions de programmation (idéalement PHP)
- Notions de bases de données

## Développement web backend avec PHP et Laravel

### Introduction à Laravel et installation

- Qu'est-ce que Laravel ? (Principes, avantages, architecture MVC).
- Installation de Laravel (via Composer, Laravel Sail/Homestead).
- Structure d'un projet Laravel (dossiers et fichiers).
- Configuration de l'environnement (fichiers .env, base de données).
- Laravel Valet (pour macOS) et Laravel Herd (pour Windows/macOS) (introduction optionnelle pour un dev plus rapide).

### Routage et contrôleurs

- Définition des routes (méthodes HTTP, paramètres de route, noms de route).
- Création de contrôleurs (actions de contrôleur, injection de dépendances).
- Middleware (protection des routes, authentification).
- Groupes de routes.
- Redirections et réponses.

### Modèles et bases de données (Eloquent ORM)

- Modèles Eloquent (création de modèles, relations entre les modèles).
- Migrations (création et gestion des tables de la base de données).
- Requêtes Eloquent (récupération, création, mise à jour et suppression de données).
- Relations Eloquent (One To One, One To Many, Many To Many).
- Factories et Seeders (pour la création de données de test).

### Vues et templating (Blade)

- Syntaxe de Blade (directives, boucles, conditions).
- Héritage de templates (layouts).

- Composants Blade.
- Transmission de données aux vues.

### Authentification et autorisation

- Authentification Laravel (système d'authentification intégré).
- Autorisation (Gate, Policies).
- Rôles et permissions (gestion des accès).

### API et gestion des données au format JSON

- Principes des APIs RESTful.
- Création de routes d'API.
- Gestion des requêtes et des réponses au format JSON.
- Ressources API (API Resources).
- Authentification API (Tokens, OAuth).
- Pagination des résultats d'API.
- Test des APIs (avec Postman ou Insomnia).

### Tests

- Introduction aux tests unitaires et aux tests d'intégration.
- Utilisation de PHPUnit avec Laravel.
- Écriture de tests pour les routes, les contrôleurs et les modèles.
- Cas pratiques : Écriture de tests pour les fonctionnalités du projet fil rouge.



# Symfony

## Développement web back-end avec PHP et Symfony

70H

CODE : WEBDEVBPHP1

### Objectifs

- Comprendre l'architecture MVC et configurer un projet Symfony.
- Maîtriser la gestion des requêtes HTTP et le routage.
- Utiliser le moteur de templates Twig pour générer des vues dynamiques.
- Maîtriser l'ORM Doctrine pour interagir avec la base de données.
- Créer et gérer des formulaires complexes avec Symfony.

### Débouchées

- Développeur back-end Symfony
- Développeur Fullstack PHP
- Développeur d'API

### Prérequis

- Bases en informatique
- Notions de programmation (idéalement PHP)

#### Introduction à Symfony et mise en place de l'environnement

- Qu'est-ce qu'un framework web et pourquoi utiliser Symfony ?
- Architecture MVC (Modèle-Vue-Contrôleur).
- Installation de Symfony (Symfony CLI, Composer).
- Structure d'un projet Symfony (répertoires, fichiers de configuration).
- Configuration de l'environnement de développement (serveur web, base de données).
- Les Bundles.
- Création d'un premier contrôleur et d'une première route.

#### Les contrôleurs et le routage

- Les contrôleurs (création, actions).
- Le routage (définition des routes, paramètres, contraintes).
- Gestion des requêtes HTTP (méthodes GET, POST, PUT, DELETE).
- Récupération des données de la requête (paramètres, formulaires).
- Génération d'URLs.

#### Les templates et Twig

- Introduction à Twig (syntaxe, variables, filtres, fonctions).
- Héritage de templates (layouts).
- Inclusion de templates.
- Gestion des formulaires avec Twig.
- Utilisation des helpers Twig.

#### Les entités et Doctrine

- Introduction à Doctrine et à l'ORM.
- Création d'entités (mappings, annotations, YAML).
- Les relations entre les entités (OneToOne, OneToMany, ManyToMany).
- Les repositories (requêtes DQL, Query Builder).

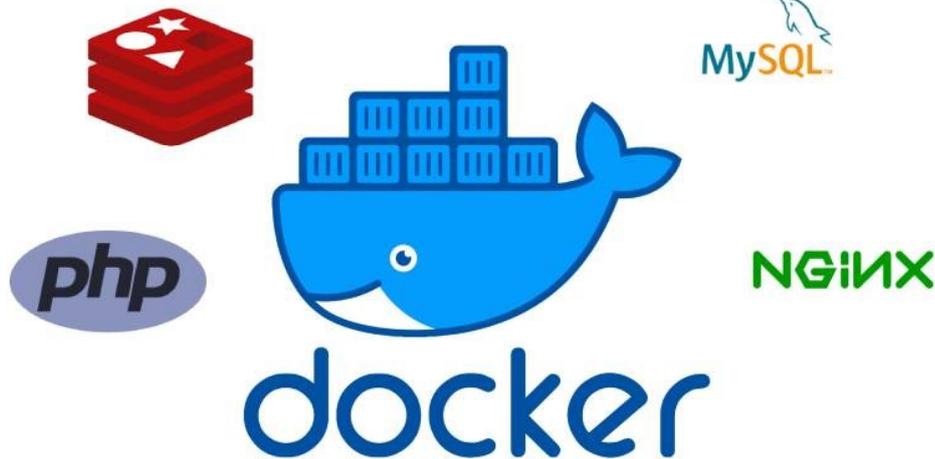
- Persistance des données (persist, flush).
- Migrations de base de données.

#### Les formulaires

- Création de formulaires avec le Form Builder.
- Validation des données de formulaires (contraintes).
- Gestion des types de champs de formulaires (text, email, password, choice, etc.).
- Traitement des données soumises par les formulaires.

#### Création d'APIs RESTful

- Introduction aux APIs RESTful (principes, méthodes HTTP, codes de statut).
- Utilisation du composant Serializer pour la sérialisation et la désérialisation des données (JSON, XML).
- Utilisation du composant Validator pour la validation des données d'entrée de l'API.
- Gestion des erreurs et des exceptions dans une API.
- Documentation d'API (Swagger/OpenAPI - introduction).



70H  
CODE : WEBDEPPHP

## Objectifs

- Comprendre le rôle des serveurs web, apprendre les commandes de base de l'administration système et configurer des serveurs web pour PHP.
- Apprendre les différentes méthodes de déploiement d'applications PHP et les appliquer au projet fil rouge.
- Comprendre les concepts de la conteneurisation et apprendre à utiliser Docker pour simplifier le déploiement et la gestion des applications.
- Comprendre les principes du CI/CD et découvrir les outils couramment utilisés pour automatiser le cycle de vie du développement logiciel.

## Débouchées

- Web master
- Intégrateur web
- Développeur Ops Junior

## Prérequis

- Bases en informatique
- Notions de programmation (idéalement PHP)
- Notions de bases de données

## Déploiement et DevOps

### Introduction aux serveurs web et à l'administration système

- Qu'est-ce qu'un serveur web ? (Fonctionnement, requêtes HTTP, réponses HTTP, architecture client-serveur).
- Introduction à Linux (commandes de base, gestion des fichiers, permissions).
- Configuration d'Apache (fichiers de configuration .htaccess, Virtual Hosts, modules PHP).
- Configuration de Nginx (fichiers de configuration, blocs server, configuration PHP-FPM).
- Comparaison entre Apache et Nginx (avantages et inconvénients, cas d'utilisation).
- Gestion des certificats SSL/TLS (Let's Encrypt, configuration pour Apache et Nginx).

### Déploiement d'applications PHP

- Méthodes de déploiement (déploiement manuel, FTP/SFTP, SSH, Git).
- Déploiement d'une application Laravel/Symfony (configuration du serveur, gestion des dépendances avec Composer, optimisation pour la production).
- Gestion des fichiers statiques et des médias (stockage local, services de stockage cloud comme AWS S3 ou Cloudinary).
- Configuration d'une base de données sur un serveur distant (sécurité, accès distant).
- Gestion des variables d'environnement (fichiers .env, variables d'environnement du serveur).
- Outils de déploiement (Deployer, Capistrano - introduction).

### Conteneurisation avec Docker

- Introduction à la conteneurisation et à Docker (images, conteneurs, Dockerfile,

Docker Compose, Docker Hub).

- Création d'images Docker pour des applications PHP (optimisation des images, gestion des dépendances avec Composer dans un conteneur).
- Orchestration de conteneurs avec Docker Compose (définition des services, gestion des réseaux et des volumes).
- Déploiement d'applications avec Docker sur un serveur (Docker Engine, Docker Swarm - introduction).

### Introduction à l'intégration continue et au déploiement continu (CI/CD)

- Principes du CI/CD (automatisation des tests, de la construction et du déploiement, avantages du CI/CD).
- Introduction à Git et aux plateformes de gestion de code source (GitHub, GitLab, Bitbucket) et leur intégration avec les outils de CI/CD.
- Introduction à un outil de CI/CD (GitHub Actions, GitLab CI, Jenkins, CircleCI).
- Automatisation des tests (tests unitaires, tests d'intégration, tests fonctionnels).
- Automatisation du déploiement (déploiement continu vers un serveur de production ou un registre de conteneurs).
- Introduction aux pratiques DevOps (surveillance, infrastructure as code).



30H  
CODE : DBSQLDAN

## Fondamentaux des données et SQL

### Objectifs

- Comprendre les différents types de données et leurs formats
- Maîtriser les concepts des bases de données relationnelles et NoSQL
- Maîtriser le langage SQL pour l'interrogation et la manipulation des données
- Comprendre la gestion des transactions et le contrôle de la concurrence
- Optimiser les requêtes SQL et comprendre les bases de l'optimisation de bases de données.

### Débouchées

- Préparation aux rôles de développeur ou analyste de données junior,

### Prérequis

- IT Essentials

#### Introduction aux données et aux bases de données

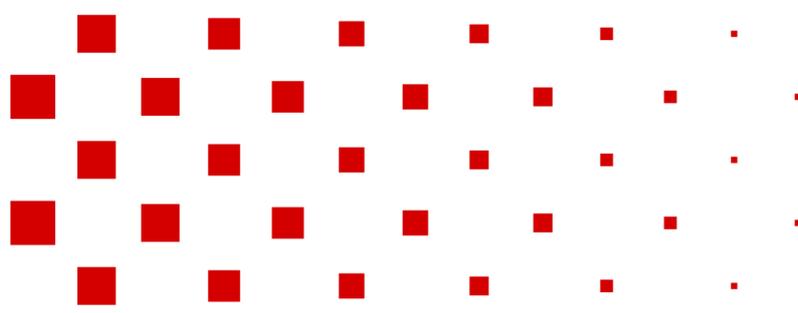
- Types de données (structurées, semi-structurées, non structurées).
- Formats de fichiers (CSV, TXT, JSON, XML, Excel, Parquet, Avro).
- Introduction aux bases de données (concepts, avantages, types).
- Modèle relationnel (tables, colonnes, clés primaires, clés étrangères, relations, normalisation).
- Introduction aux bases de données NoSQL (concepts de base, types : clé-valeur, document, graphe, colonne).
- Modélisation conceptuelle, logique et physique des données.

#### Langage SQL

- Requêtes SELECT (filtrage avec WHERE, tri avec ORDER BY, agrégation avec GROUP BY et HAVING, DISTINCT).
- Jointures (INNER, LEFT, RIGHT, FULL OUTER, CROSS JOIN, self-join, jointures naturelles).
- Sous-requêtes (corrélées et non corrélées, EXISTS, IN).
- Fonctions SQL (scalaires, d'agrégation, de conversion, de date, de texte, analytiques – fonctions de fenêtrage).
- Manipulation de données (INSERT, UPDATE, DELETE, MERGE).
- Création et modification de tables (CREATE TABLE, ALTER TABLE, DROP TABLE, contraintes, types de données avancés).
- Vues (CREATE VIEW, vues matérialisées).
- Transactions et contrôle de la concurrence (ACID, niveaux d'isolation, gestion des verrous).
- Optimisation des requêtes SQL (index, plans d'exécution, utilisation d'EXPLAIN PLAN).

#### Statistiques avec SQL :

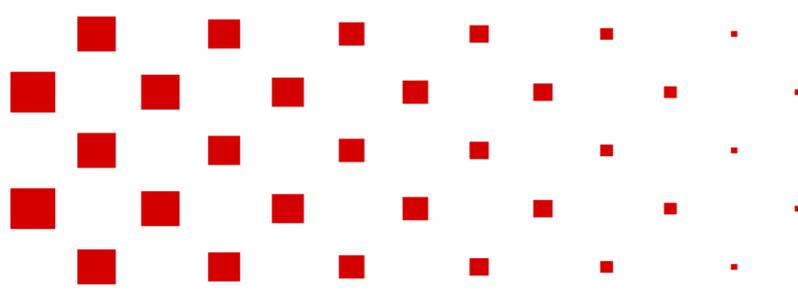
- Fonctions d'agrégation statistiques (AVG, COUNT, MAX, MIN, SUM, STDDEV, VARIANCE).
- Calcul des percentiles et des quartiles.
- Corrélation et régression linéaire simple (avec des fonctions SQL ou des extensions).
- Tests d'hypothèses (introduction).



edify digital

**PACKAGE**

**PRODUCTIVITE**





20H  
CODE : PRODGET

## Objectifs

Acquérir des compétences numériques essentielles pour la vie quotidienne et professionnelle, notamment en matière d'utilisation d'Internet, de communication en ligne, de sécurité et de résolution de problèmes techniques courants.

## Débouchées

Amélioration de l'employabilité générale, autonomie dans l'utilisation des outils numériques, préparation à des formations plus avancées en informatique.

## Prérequis

- Aucun prérequis

## GET CONNECTED

### Introduction au monde numérique

- Qu'est-ce que le numérique ? (Concepts de base, terminologie).
- L'impact du numérique sur la société et le travail.
- Les différents types d'appareils numériques (ordinateurs, smartphones, tablettes).
- Utilisation d'Internet (10h)
- Navigation sur le web (navigateurs, moteurs de recherche, URL).
- Recherche d'informations en ligne (stratégies de recherche, évaluation des sources).
- Communication en ligne (e-mail, messagerie instantanée, réseaux sociaux).

### Communication et collaboration en ligne

- Utilisation des outils de communication et de collaboration (visioconférence, partage de fichiers, plateformes collaboratives).
- Bonnes pratiques de communication en ligne (netiquette, communication professionnelle).

### Sécurité en ligne

- Menaces en ligne (malware, phishing, cyberharcèlement).
- Protection des données personnelles (mots de passe, confidentialité).
- Bonnes pratiques de sécurité informatique (antivirus, pare-feu).

### Résolution de problèmes techniques courants

- Dépannage de base (problèmes de connexion, problèmes d'affichage, problèmes d'imprimante).
- Recherche d'aide en ligne (forums, tutoriels).



# DÉCOUVREZ LE PACK OFFICE EXCEL+WORD+PP

**MICROSOFT 365 WORD-EXCEL-POWERPOINT**

30H  
CODE : PRODOFF

## Objectifs

Maîtriser les fonctionnalités essentielles de Word, Excel et PowerPoint pour créer des documents professionnels, analyser des données et réaliser des

## Débouchées

- Amélioration de la productivité bureautique,
- Création de supports de communication efficaces,
- Meilleure gestion des données.

## Prérequis

- GET CONNECTED

### Word

- Interface et mise en page
- Rubans
- onglets
- barres d'outils
- mise en page
- marges
- orientation
- sauts de page.
- Police
- taille
- couleur
- alignement
- interligne
- puces et numéros
- styles.
- Création et manipulation de tableaux
- insertion et modification d'images
- Publipostage
- suivi des modifications
- modèles

### Excel

- Cellules
- feuilles de calcul
- saisie de données
- formats de nombres
- Opérateurs
- Fonctions de calcul
- Fonctions de texte
- Fonctions de date.
- Création et mise en forme de tableaux
- Création de différents types de graphiques
- Tris
- Filtres
- Tableaux croisés dynamiques
- Fonctions d'analyse

### PowerPoint

- Diapositives
- Mises en page

- Thèmes Masques des diapositives
- Texte
- Images
- Graphiques
- Vidéos
- Audio
- Animations des éléments
- Transitions entre les diapositives
- Mode présentation
- Options d'affichage
- Enregistrement de la présentation.



20H  
CODE : PRODWEB

## Objectifs

Apprendre les bases de la création de sites web, de la conception à la publication, en utilisant des outils simples et accessibles.

## Débouchées

Création de sites web personnels ou professionnels simples, compréhension des enjeux du web, amélioration des compétences numériques.

## Prérequis

- GET CONNECTED

## CREATION DE SITE WEB

### Introduction au web et à HTML

- Fonctionnement du web (serveurs, navigateurs, URL).
- Introduction à HTML (structure d'une page HTML, balises de base).
- Création d'une page web simple avec HTML.

### CSS (Cascading Style Sheets)

- Introduction à CSS (sélecteurs, propriétés).
- Mise en page avec CSS (boîtes, positionnement, flexbox, grid).
- Mise en forme du texte et des images avec CSS.

### Publication d'un site web

- Choix d'un hébergeur et d'un nom de domaine.
- Téléchargement des fichiers du site web sur le serveur.
- Notions de base de SEO (Search Engine Optimization).

### Introduction aux CMS (Content Management Systems)

- Présentation des CMS (WordPress, Wix, etc.).
- Avantages et inconvénients des CMS.
- IA Générative



20H  
CODE : PRODINF

## Objectifs

Acquérir les bases du design graphique et apprendre à créer des visuels percutants pour différents supports de communication.

## Débouchées

- Création de supports de communication visuels (affiches, flyers, infographies,
- Visuels pour les réseaux sociaux)
- Amélioration de la communication visuelle
- Compétences transférables vers le marketing digital et la création de

## Prérequis

- GET CONNECTED

## INFOGRAPHIE

### Principes de base du design graphique

- Théorie des couleurs (harmonie des couleurs, psychologie des couleurs, palettes de couleurs).
- Typographie (choix des polices, hiérarchie typographique, lisibilité).
- Composition et mise en page (règle des tiers, équilibre, contraste, espace blanc).
- Formats de fichiers et résolution d'image (JPEG, PNG, SVG, DPI).

- Adaptation des infographies pour l'impression (formats d'impression, résolution).

### Utilisation d'outils de création graphique

- Introduction à un logiciel de création graphique (Canva, Adobe Illustrator, Inkscape - choix selon le public cible et le niveau souhaité).
- Création de formes et d'illustrations vectorielles (si logiciel vectoriel).
- Retouche et manipulation d'images (si logiciel de retouche photo).
- Utilisation de modèles et de ressources graphiques.

### Création de différents types d'infographies

- Infographies statistiques (représentation de données chiffrées).
- Infographies chronologiques (représentation d'événements dans le temps).
- Infographies comparatives (comparaison de deux ou plusieurs éléments).
- Infographies géographiques (représentation de données sur une carte).

### Adaptation des infographies aux différents supports

- Adaptation des infographies pour le web (optimisation pour les réseaux sociaux, intégration dans un site web).

# Intelligence artificielle

Améliorer sa productivité et son confort



10H  
CODE : PRODAI

## Objectifs

Comprendre les bases de l'IA et apprendre à utiliser des outils et des applications basés sur l'IA pour améliorer la productivité et l'efficacité. Ce module ne vise pas à former des développeurs en IA, mais des utilisateurs.

## Débouchées

Amélioration de la productivité grâce à l'IA, compréhension des enjeux de l'IA dans le monde professionnel, capacité à identifier et à utiliser les outils d'IA pertinents.

## Prérequis

- GET CONNECTED

## INTELLIGENCE ARTIFICIELLE

### Introduction à l'Intelligence Artificielle

- Qu'est-ce que l'IA ? (Définitions, concepts clés : Machine Learning, Deep Learning, NLP).
- Les différents types d'IA (IA faible, IA forte).
- Domaines d'application de l'IA (reconnaissance d'images, traitement du langage naturel, automatisation, prédiction).
- Ethique et enjeux de l'IA.

### Utilisation des outils d'IA pour la productivité

- Outils de traduction automatique (Google Traduction, DeepL).
- Outils de génération de texte (ChatGPT, Bard).
- Outils de reconnaissance vocale et de transcription (Google Assistant, Siri, transcription automatique).
- Outils d'automatisation des tâches (Zapier, IFTTT).
- Outils d'analyse de données basés sur l'IA.

### Utilisation de l'IA pour la création de contenu

- Génération d'images avec l'IA (Midjourney, DALL-E 2, Stable Diffusion).
- Création de vidéos avec l'IA.
- Utilisation de l'IA pour l'optimisation du contenu (SEO, marketing).

# Spécial juniors

Nous rendons le code et la robotique accessibles et amusants ! Nos ateliers sont conçus pour donner aux jeunes les clés pour comprendre le monde numérique et se préparer aux métiers de demain.

edify digital



## ATELIERS

DE CODE & D'INITIATION

A L'INFORMATIQUE

POUR LES ENFANTS DE 10 à 21 ANS



- PROGRAMMATION
- PILOTAGE DE DRONE
- REALITE VIRTUELLE
- ROBOTIQUE ET INTELLIGENCE ARTIFICIELLE
- INFOGRAPHIE ET SYSTEME D'EXPLOITATION

MONTAGE VIDEO  
PHOTOSHOP



# Nos partenaires technologiques



## Ils nous font confiance



## Où Nous trouver



### CAMEROUN

Akwa 34 rue Boue Iapeyrère,  
face hôtel planet

Tel : +237 670 556 187  
+237 697 925 426  
+237 656 019 902



### CONGO BRAZAVILLE

Djata derrière le  
stade Aphonse  
Massamba Debat

Tel: +242 06 923 74 03



### CANADA

102 Rue de Touraine, Lévis,  
QC, G6J 2A8

Tel: +1 (581) 308-8186



### FRANCE

30 Rue de Liège  
75008 Paris, France  
RCS Paris

Tel : +33(0) 920 303 773

[www.edify.site](http://www.edify.site)

[ask@edify.site](mailto:ask@edify.site)