

**PLAQUETTE DE FORMATION**

480 H

**6 Mois**

**Devenir développeur**

**Front-End** de A à Z

# Présentation de la formation

La formation en développement web front-end de Nimba Hub, est une opportunité passionnante pour transformer votre passion pour la technologie en une carrière prometteuse. Notre programme de formation a été soigneusement conçu pour répondre aux besoins changeants du marché du travail numérique, en vous offrant une base solide pour devenir un développeur front-end compétent.

Notre formation Front-End est une plongée approfondie dans l'art et la science du développement web. Elle vise à vous équiper des compétences techniques, des connaissances conceptuelles et des compétences pratiques nécessaires pour créer des interfaces web interactives et attrayantes. Vous apprendrez à donner vie à des designs créatifs en utilisant les dernières technologies front-end.

## Notre Objectif

Notre objectif ultime est de **vous préparer à une carrière réussie en tant que développeur web front-end**. Nous croyons fermement que la maîtrise du développement front-end est essentielle pour répondre aux demandes croissantes de l'industrie technologique. Notre programme vise à :



### Vous doter des compétences techniques

Vous acquerrez une compréhension approfondie des langages clés tels qu'HTML, CSS et JavaScript, ainsi que des frameworks modernes.



### Cultiver votre créativité

Vous développerez votre sens artistique pour créer des interfaces utilisateur intuitives et esthétiques.



### Nourrir votre pensée critique

Vous apprendrez à résoudre des problèmes complexes, à analyser les besoins des utilisateurs et à créer des solutions innovantes.



### Vous préparer à la collaboration

Vous travaillerez sur des projets réels en équipe, acquérant ainsi une expérience précieuse de la collaboration au sein de l'industrie.

# Programme de Formation

## Phase 1: Formation en Conception et Développement Web

### Module 1: Introduction au Développement Web

1. Vue d'ensemble de haut niveau du développement web
2. Conseils avant de commencer

### Module 2: Fondamentaux du Web

1. Introduction à HTML
2. Introduction à CSS
3. Travail avec les couleurs

### Module 3: Théorie CSS

1. Conflits entre sélecteurs
2. Héritage et sélecteur universel
3. Le modèle de boîte CSS
4. Types de boîtes
5. Positionnement absolu

### Module 4: Construction des Layouts

1. Les 3 méthodes de création de layouts
2. Utilisation des floats
3. box-sizing: border-box
4. Vue d'ensemble de Flexbox
5. Vue d'ensemble de CSS Grid

### Module 5: Principes de Conception Web

1. Vue d'ensemble de la conception web et des personnalités de site web
2. Règles de conception web : Typographie
3. Règles de conception web : Couleurs
4. Règles de conception web : Images et illustrations
5. Règles de conception web : Icônes

### Module 6: Éléments de Design Avancés

1. Règles de conception web : Ombres
2. Règles de conception web : Border-radius
3. Règles de conception web : Espace blanc (Whitespace)
4. Règles de conception web : Hiérarchie visuelle
5. Règles de conception web : Expérience utilisateur (UX)
6. Le cadre des personnalités de site web

### **Module 1: Introduction à Javascript**

1. Conseils avant de commencer
2. Introduction rapide au JavaScript
3. Types de données
4. Logique booléenne
5. Les versions de JavaScript : ES5, ES6+ et ESNext

### **Module 2: Fondamentaux des fonctions**

1. Fonctions appelant d'autres fonctions
2. Révision des fonctions
3. Comment apprendre à coder
4. Penser comme un développeur
5. Résumé : Quelle structure de données utiliser ?

### **Module 3: Avancé en fonctions et arrays**

1. Fonctions de première classe et fonctions d'ordre supérieur
2. Fermetures (Closures)
3. Transformations de données : map, filter, reduce
4. Résumé : Quelle méthode de tableau utiliser ?

### **Module 4: Le document Object Model (DOM)**

1. Comment fonctionne réellement le DOM
2. Propagation des événements : Montée et capture
3. Chargement efficace des scripts : defer et async

### **Module 5: Programmation orientée objet (POO)**

1. Qu'est-ce que la POO ?
2. POO en JavaScript
3. Héritage prototypal et chaîne de prototypes
4. Object.create et Héritage entre "C
5. Résumé des classes ES6

### **Module 6: Planification de projet web**

1. Projet Mappy : Comment planifier un projet web
2. Projet Mappy : Considérations finales

### **Module 1: Introduction à React**

1. Conseils avant de commencer
2. Pourquoi les frameworks front-end existent-ils ?
3. Qu'est-ce que React ?
4. Les deux options pour configurer un nouveau projet React
5. Section Summary

### **Module 2: Fondamentaux des Composants**

1. Les composants comme blocs de construction
2. Qu'est-ce que JSX ?
3. Séparation des préoccupations
4. Props, Immutabilité et Flux de données unidirectionnel
5. Les règles de JSX
6. Récapitulatif de la section

### **Module 3: Gestion de l'état en React**

1. Qu'est-ce que l'état (State) en React ?
2. Mécanismes de l'état
3. L'état vs. Props
4. Penser en React
5. Fondamentaux de la gestion de l'état
6. État dérivé
7. Le prop "children" pour créer un bouton réutilisable
8. Comment diviser une interface utilisateur en composants
9. Catégories de composants
10. Composition des composants
11. Props en tant qu'API de composant
12. Composants, instances et éléments
13. Comment fonctionne le rendu: Aperçu

### **Module 4: Approfondissement du rendu et des événements**

1. Mécanismes de rendu: Phase de rendu et phase de validation
2. Comment fonctionne le "diffing"
3. La prop "key"
4. Règles pour la logique de rendu : composants purs
5. Gestion des événements en React
6. Bibliothèques vs. Frameworks & L'écosystème React



## Module 7: Éléments et Composants

1. Règles de conception web : Éléments et composants - Partie I
2. Changer la direction de flex en colonne
3. Centre vertical avec position absolue et transformation

## Module 8: Patterns et Principes de Design

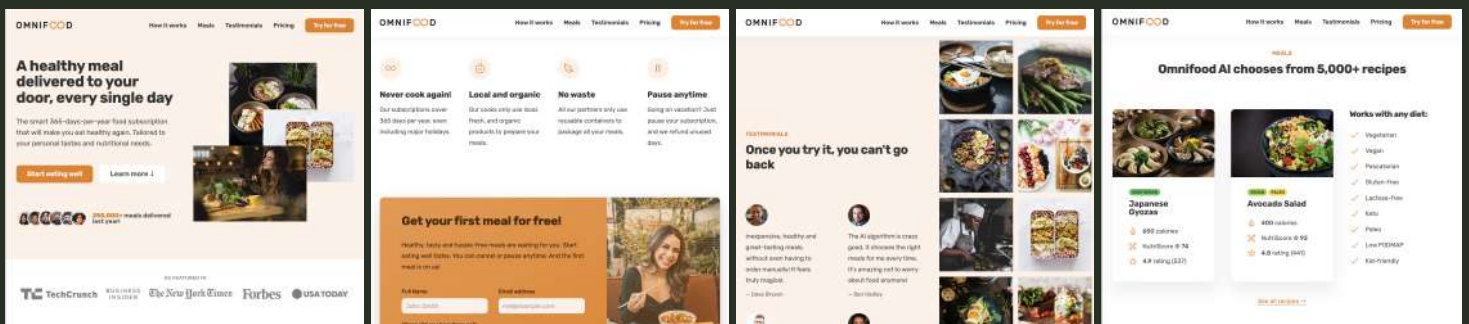
1. Règles de conception web : Éléments et composants - Partie II
2. Les 7 étapes pour un excellent site web
3. Définir et planifier le projet
4. Esquisser les premières idées de mise en page

## Module 9: Design Réactif

1. Principes de design réactif
2. Comment fonctionnent les requêtes média
3. Comment sélectionner les points de rupture (Breakpoints)

### Travaux pratiques (1)

OmniFood est un site web dédié à la présentation d'un service de livraison de nourriture.



### Objectifs du projet :

- Maîtriser la création d'un site web complet.
- Intégrer une mise en page responsive pour assurer une compatibilité avec tous les types d'appareils.
- Apprendre à utiliser des techniques avancées de CSS et des frameworks comme Flexbox ou Grid.

### Résultats attendus :

- Un site web entièrement fonctionnel et esthétiquement plaisant, adaptatif à tous les formats d'écran.

## **Module 5: Hooks et Cycle de Vie**

1. Le cycle de vie d'un composant
2. Une première approche des effets avec `useEffect``
3. Les règles des hooks
4. Introduction aux hooks: `useState``, `useRef``, `useReducer``
5. Quand et comment créer un Hook personnalisé?
6. Comparaison: Composants de classe vs. Composants fonctionnels
7. `useState` vs. `useReducer`

## **Module 6: Avancé et Optimisation**

1. Routage et Applications Monopage (SPAs)
2. Options de style pour les applications React
3. Qu'est-ce que l'API Context ?
4. Penser en React: Gestion avancée de l'état
5. Optimisation des performances et rendus inutiles
6. Comprendre `memo``, `useMemo`` et `useCallback``
7. Ne pas optimiser prématurément!
8. Règles et meilleures pratiques de `useEffect``

## **Module 7: Gestion d'état avancée et Outils**

1. Introduction à Redux
2. Middleware Redux et thunks
3. Qu'est-ce que Redux Toolkit (RTK) ?
4. Redux vs. Context API 5. Présentation de Tailwind CSS
6. Qu'est-ce que Supabase ?
7. Qu'est-ce que React Query ?
8. Vue d'ensemble de la réutilisabilité en React

## **Module 8: Projets Pratiques**

1. Planification d'une application : "Fast React Pizza Co."
2. Planification d'une application : "The Wild Oasis"

## Module 7: JavaScript asynchrone, AJAX et APIs

1. Comment fonctionne le web : Demandes et réponses
2. Promesses et l'API Fetch
3. Asynchronisme en coulisses : La boucle d'événements

## Module 8: Développement JavaScript moderne

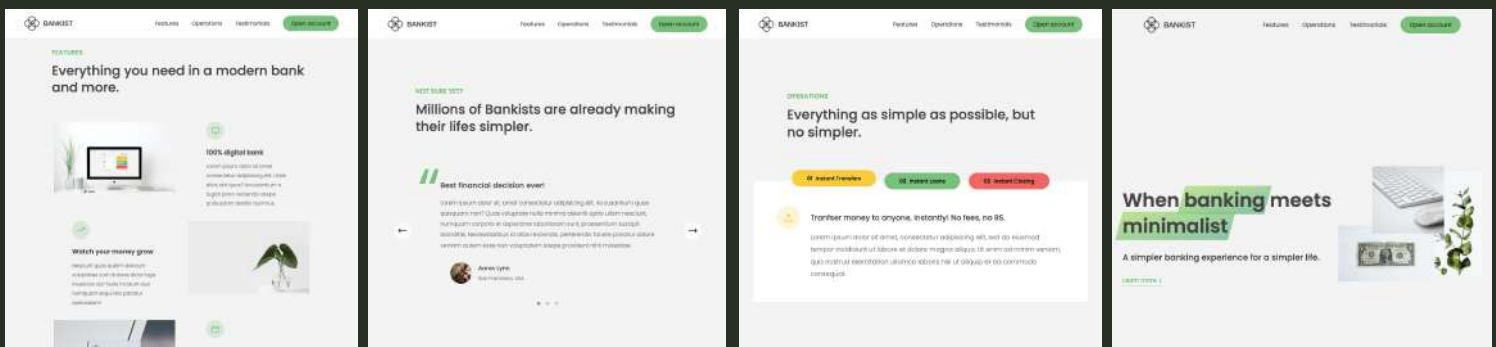
1. Vue d'ensemble du développement JavaScript moderne
2. Vue d'ensemble des modules en JavaScript
3. Programmation JavaScript moderne, propre et déclarative

## Module 9: Architecture et Projet

1. Forkify : Vue d'ensemble et planification du projet
2. L'architecture MVC
3. Gestionnaires d'événements dans MVC : Modèle éditeur-abonné
4. Projet Forkify : Considérations finales

## Travaux pratiques (2)

Une application pour gérer et suivre les dépenses personnelles.



## Objectifs du projet :

- Apprendre les bases du JavaScript.
- Comprendre les structures de données comme les tableaux et les objets.
- Mise en œuvre de la logique pour ajouter/supprimer des dépenses.

## Résultats attendus :

- Une application simple mais fonctionnelle pour suivre et gérer les dépenses.



### Travaux pratiques (3)

Projet : Une introduction à React à travers la création d'une application simple

#### Objectifs du projet :

- Comprendre les notions de composants.
- Se familiariser avec JSX, les props, le state et les formulaires dans React

#### Résultats attendus :

- Une application React basique mais fonctionnelle, montrant la puissance des composants et du state.
- 

### Travaux pratiques (4)

Projet usePopcorn: Un projet intermédiaire pour approfondir la maîtrise de React

#### Objectifs du projet :

- Apprendre le data fetching.
- Explorer le hook useEffect et la création de custom hooks

#### Résultats attendus :

- Une application montrant comment utiliser les données externes et intégrer des effets et logiques personnalisées avec les hooks.
- 

### Travaux pratiques (5)

Projet ReactQuiz: Une application de quiz.

#### Objectifs du projet :

- Combinaison des concepts précédents.
- Introduction au reducer, optimisation de la performance

#### Résultats attendus :

- Une application de quiz interactive, avec une bonne gestion de l'état et des performances
- 

### Travaux pratiques (6)

Projet WORLDWISE: Une application SPA (Single Page Application)

#### Objectifs du projet :

- Utiliser React Router pour la navigation.
- Introduction à Redux et Redux Toolkit pour la gestion avancée de l'état.

#### Résultats attendus :

- Une application SPA moderne avec une navigation fluide et une gestion d'état avancée.

**1000 TechLeaders**