

Enseigner la programmation de services web aux étudiants en utilisant une plateforme gratuite peut être une expérience enrichissante et pratique. Voici un plan détaillé pour structurer votre cours et les outils gratuits à utiliser.

Plan du Cours

1. Introduction aux Services Web

- **Objectif:** Comprendre les bases des services web, RESTful APIs, et le protocole HTTP.
- **Contenu:** Concepts de client-serveur, requêtes HTTP (GET, POST, PUT, DELETE), et JSON.

2. Configuration de l'Environnement de Développement

- **Objectif:** Installer et configurer les outils nécessaires.
- **Contenu:**
 - Installation de Node.js et npm.
 - Installation d'un éditeur de code (Visual Studio Code recommandé).
 - Introduction à Git et GitHub pour le contrôle de version.

3. Création d'un Serveur Web Basique avec Node.js et Express

- **Objectif:** Écrire un serveur web simple qui répond aux requêtes HTTP.
- **Contenu:**
 - Initialisation d'un projet Node.js (`npm init`).
 - Installation et utilisation d'Express pour créer un serveur web.
 - Écriture de routes basiques (`GET`, `POST`).

4. Gestion des Données avec un Fichier

- **Objectif:** Apprendre à lire et écrire des données dans un fichier JSON.
- **Contenu:**
 - Utilisation des modules `fs` de Node.js pour la gestion des fichiers.
 - Création de routes pour sauvegarder et lire des données depuis un fichier JSON.

5. Déploiement Gratuit sur Render

- **Objectif:** Déployer l'application sur une plateforme gratuite.
- **Contenu:**
 - Création d'un compte sur Render.
 - Déploiement de l'application à partir d'un dépôt GitHub.

6. Projets Pratiques

- **Objectif:** Appliquer les connaissances acquises dans des projets concrets.
- **Contenu:**
 - Projet 1: Service web de gestion de tâches.
 - Projet 2: Service web de gestion de contacts.

Outils et Ressources Gratuits

1. **Node.js et Express:** Pour la création du serveur web.
2. **Visual Studio Code:** Un éditeur de code gratuit et puissant.
3. **GitHub:** Pour le contrôle de version et l'hébergement du code.
4. **Render:** Une plateforme gratuite pour déployer des applications web.

Étape par Étape

1. Introduction aux Services Web

- **Session 1:** Présentation des concepts de base des services web.
- **Exercice:** Analyser des exemples de requêtes HTTP.

2. Configuration de l'Environnement de Développement

- **Session 2:** Installation de Node.js, npm, et Visual Studio Code.
- **Exercice:** Configurer un projet Node.js et installer Express.

3. Création d'un Serveur Web Basique

- **Session 3:** Écrire un serveur Express avec une route **GET**.
- **Exercice:** Ajouter des routes **POST** et **DELETE**.

4. Gestion des Données avec un Fichier

- **Session 4:** Utilisation des modules **fs** pour lire et écrire dans un fichier JSON.
- **Exercice:** Créer un service web qui sauvegarde des données soumises via **POST** et les renvoie via **GET**.

5. Déploiement Gratuit sur Render

- **Session 5:** Création d'un compte Render et déploiement de l'application.
- **Exercice:** Déployer une application web simple et tester son fonctionnement en ligne.

6. Projets Pratiques

- **Session 6 et au-delà:** Projets pratiques, retour sur les concepts et développement de projets plus complexes.
- **Projet 1:** Créer un service web de gestion de tâches.
- **Projet 2:** Créer un service web de gestion de contacts.

Exemple de Code

Serveur Basique avec Node.js et Express

1. Initialiser le projet:

```
mkdir mon-service-web  
cd mon-service-web
```

```
npm init -y
npm install express
```

2. Créer le serveur (src/app.js):

```
const express = require('express');
const fs = require('fs');
const app = express();
const port = 3000;

app.use(express.json());

app.get('/', (req, res) => {
  res.send('Hello World!');
});

app.post('/save', (req, res) => {
  const data = req.body;
  fs.writeFileSync('data.json', JSON.stringify(data));
  res.send('Données sauvegardées avec succès.');
```

```
});

app.get('/data', (req, res) => {
  const data = fs.readFileSync('data.json');
  res.send(JSON.parse(data));
});

app.listen(port, () => {
  console.log(`Serveur en écoute sur le port ${port}`);
});
```

3. Démarrer le serveur:

```
node src/app.js
```

Déploiement sur Render

1. Créer un dépôt GitHub:

```
git init
git add .
git commit -m "Initial commit"
git remote add origin https://github.com/votreutilisateur/mon-service-web.git
git push -u origin master
```

2. Déployer sur Render:

- Allez sur [Render](#), créez un compte et liez votre dépôt GitHub.
- Suivez les instructions pour déployer votre application.

Conclusion

En suivant ce plan, vous pourrez enseigner efficacement la programmation de services web en utilisant des outils et des plateformes gratuits. Cela permettra aux étudiants d'acquérir des compétences pratiques en développement web et en déploiement d'applications.