|  |
| --- |
| * + 1. **PROSIT 4 [Maquette Statique]** |
| **[Rami Mohamed Amine]** | |

|  |  |
| --- | --- |
| **Rôle** | **Nom Prénom** |
| **Animateur** | WISSAM RECHAM |
| **Scribe** | SELATNA MOHAMED AMINE |
| **Gestionnaire** | DADOU DJAMEL |
| **Secrétaire** | SIDALI HALLAOUA |

08/02/2025 A2 – Groupe 1

1. **Mots clés :**

* **Expérience utilisateur**

1. **Mots à définir :**

* **Expression régulière :** Une **expression régulière** (ou regex) est une séquence de caractères qui définit un **motif de recherche** dans du texte. Elle permet de rechercher, valider, extraire ou remplacer des chaînes de caractères en fonction de règles précises. Par exemple, l'expression régulière ^\d{3}-\d{2}-\d{4}$ correspond à un format de numéro de sécurité sociale américain (XXX-XX-XXXX).
* **Evènement JS :** Un **événement en JavaScript** est une action ou un changement d'état détecté par le navigateur et qui peut être géré par du code. Cela inclut des actions utilisateur (clic, pression de touche, déplacement de souris) ou des événements système (chargement de page, fin d’une requête réseau). Par exemple, un **événement click** peut être capturé avec addEventListener("click", function() { /\* action \*/ }); pour exécuter une fonction lorsqu’un élément est cliqué.

1. **Analyse du contexte :**

* **Un client demande de rendre le site statique interactif**

1. **Définition de la problématique :**

* **Comment rendre le site interactif en utilisant JS et en respectant les bonnes pratiques JS**

1. **Les contraintes :**

* **Cahier de charges**

1. **Plan d’actions :**

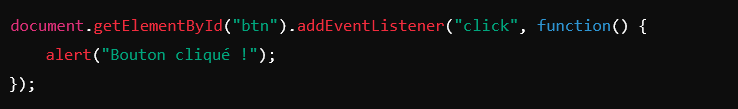
**Etudier les principes de javascript :**

JavaScript est un langage de programmation dynamique et interprété, principalement utilisé pour le développement web, mais aussi pour des applications serveur grâce à Node.js. Il est conçu pour être léger et flexible, permettant aux développeurs de créer des expériences interactives et réactives sur le web.

Modèle événementiel

L'un des principes fondamentaux de JavaScript est son modèle basé sur les événements. Il permet d'exécuter du code en réponse à des actions des utilisateurs, comme un clic, une pression de touche ou le chargement d'une page. Cela est possible grâce aux gestionnaires d'événements.

Exemple :

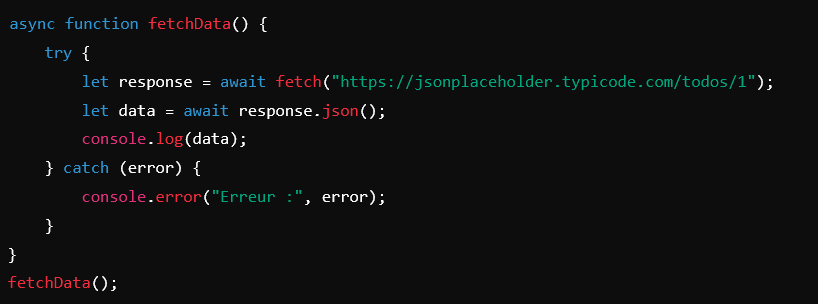
****

Ici, un événement click est attaché à un bouton avec l’ID btn. Lorsqu’un utilisateur clique dessus, une alerte s'affiche.

Exécution asynchrone et non bloquante

JavaScript gère l'exécution de code de manière asynchrone grâce aux **promises** et à async/await. Cela permet d'effectuer des opérations longues (comme des requêtes réseau) sans bloquer l'interface utilisateur.

Exemple avec fetch pour récupérer des données d'une API :

****

Ici, fetchData attend que la requête soit terminée avant de continuer, sans bloquer l’exécution du reste du programme.

Programmation orientée objet avec les prototypes

JavaScript repose sur un modèle basé sur les **prototypes** plutôt que sur les classes traditionnelles. Cependant, ES6 a introduit la syntaxe class pour rendre le code plus lisible et plus proche d’autres langages orientés objet.

Exemple avec une classe :

****

Ici, Personne est une classe qui permet de créer des objets avec des propriétés nom et age, ainsi qu'une méthode sePresenter.

Interopérabilité et exécution multiplateforme

JavaScript peut être exécuté sur tous les navigateurs modernes et même sur des serveurs grâce à **Node.js**. Cela en fait un langage puissant aussi bien pour le **frontend** que pour le **backend**.

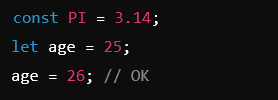
Exemple d’un simple serveur avec **Node.js et Express** :



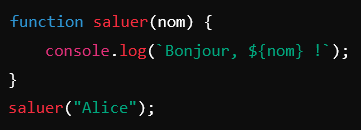
**Voir les bonnes pratiques de JS :**

Voici quelques bonnes pratiques pour écrire du code JavaScript plus propre et efficace :

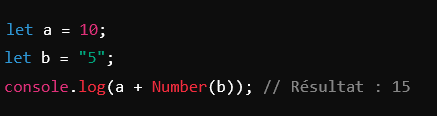
**Utiliser const et let plutôt que var** :

****

**Écrire un code clair** : choisir des noms explicites et bien indenter.

****

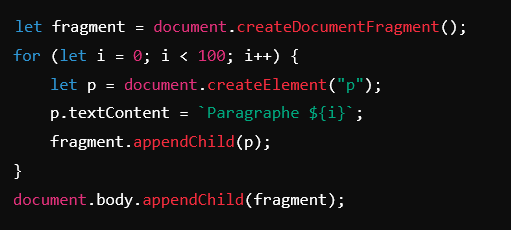
**Faire attention aux types de données** :

****

**Utiliser async/await pour gérer les requêtes asynchrones** :

****

**Éviter de modifier directement le DOM à chaque opération** :

****

**Toujours tester son code et utiliser un linter comme ESLint** pour détecter les erreurs.

**Adapter le site selon les besoins du client :**

Voir le fichier index.html