Création de l'interface Web administrateur

Projet MEUNOT — Système d'occultation automatique d'un conteneur à verre

Présentation du problème

La société MEUNOT souhaite automatiser l'ouverture et la fermeture de l'orifice d'un conteneur à verre afin de limiter les nuisances sonores pendant certaines heures.

Le système comporte une porte motorisée qui doit s'ouvrir uniquement pendant des plages horaires autorisées.

Un administrateur doit pouvoir programmer ces horaires depuis son ordinateur ou son smartphone, en se connectant au réseau Wi-Fi créé par la carte Raspberry Pi Pico W.

Votre mission : concevoir l'interface Web permettant à l'administrateur de gérer le système.

Cette première séance est consacrée à la création de la page Web. La communication avec la carte Pico W sera réalisée ultérieurement.

Fonctionnalités attendues

L'interface doit permettre de :

- Afficher le nom du système et son état.
- Programmer trois plages horaires d'ouverture et de fermeture par jour.
- Enregistrer les horaires saisis.
- Disposer de deux boutons de commande manuelle : OUVRIR et FERMER.
- Afficher un message confirmant les actions de l'administrateur.

Avant de commencer la programmation, vous allez imaginer l'apparence de l'interface Web qui permettra à un administrateur de piloter le système MEUNOT.
Cette interface devra être accessible depuis un ordinateur ou un smartphone connecté au réseau Wi-Fi du système.

Travail demandé :

Sur une feuille blanche, **réaliser** à main levée un croquis de votre future interface Web.

Vous êtes libres de choisir l'organisation des éléments, les couleurs, les formes des boutons et le style graphique.

Votre interface doit cependant permettre à l'administrateur de :

- Visualiser l'état de la porte (ouverte ou fermée).
- Programmer jusqu'à trois plages horaires d'ouverture et de fermeture.
- Enregistrer les horaires programmés.
- Commander manuellement l'ouverture et la fermeture de la porte.
- Visualiser les messages de confirmation des commandes.

Quelques questions pour guider votre réflexion

- Où placer les informations les plus importantes ?
- Comment distinguer les boutons OUVRIR et FERMER ?
- Comment organiser les trois plages horaires pour qu'elles soient faciles à comprendre ?
- Quelles couleurs utiliser pour rendre l'interface agréable et lisible ?
- Comment rendre l'interface simple à utiliser sur un smartphone ?

Conseil : imaginez que vous êtes l'administrateur du système. Vous devez pouvoir comprendre et utiliser l'interface sans avoir besoin d'un mode d'emploi.

Travail à produire

Un croquis annoté sur feuille indiquant :

- L'emplacement des différents éléments.
- Les couleurs envisagées.
- La forme et la disposition des boutons.
- L'organisation générale de la page.

Conservez votre croquis : vous devrez essayer de le reproduire à la fin de la séance !

Préparation de l'environnement

Dans votre espace personnel, **créer** un dossier nommé Poubelle_IHM.

Dans ce dossier, **créer** les trois fichiers suivants :

```
Poubelle_IHM/
|
 ├── index.html
 ├── style.css
 └── script.js
```

Ces trois fichiers ont des rôles différents :

Fichier	Langage	Rôle
index.html	HTML	Organiser les éléments de la page
style.css	CSS	Modifier les couleurs, les dimensions et la présentation
script.js	JavaScript	Réagir aux actions de l'utilisateur

Étape 1 — Créer la structure HTML

Ouvrir le fichier index.html et **recopier** le code suivant :

```
<!DOCTYPE html>
<html lang="fr">

<head>
    <meta charset="UTF-8">
    <title>MEUNOT - Administration</title>
    <link rel="stylesheet" href="style.css">
</head>
```

```
<body>

    <h1>MEUNOT</h1>
    <h2>Administration du conteneur à verre</h2>

    <h3>État du système</h3>
    <p id="etat">Porte fermée</p>

    <h3>Programmation des horaires</h3>

    <p>Plage horaire 1</p>
    Ouverture :
    <input type="time" id="ouv1">
    Fermeture :
    <input type="time" id="ferm1">

    <p>
        <button onclick="enregistrer()">
            Enregistrer les horaires
        </button>
    </p>

    <p id="message"></p>

    <h3>Commande manuelle</h3>

    <button onclick="ouvrir()">OUVRIR</button>
    <button onclick="fermer()">FERMER</button>

    <script src="script.js"></script>

</body>
</html>
```

Aide : comprendre les balises HTML

- <h1> : affiche un titre principal.
- <p> : affiche un paragraphe.
- <input> : permet à l'utilisateur de saisir une valeur.
- type="time" : affiche un champ de saisie d'heure.
- <button> : crée un bouton.
- id="ouv1" : donne un identifiant unique à un élément.
- onclick="ouvrir()" : demande d'exécuter la fonction JavaScript ouvrir() lorsque l'utilisateur clique sur le bouton.

Travail à **réaliser** :

1. Enregistrer le fichier et ouvrir index.html dans le navigateur.
2. Vérifier que les 2 champs horaires sont présents.
3. Tester la saisie d'une heure.
4. Modifier le titre de la page pour afficher « MEUNOT — Gestion des horaires ».
5. Ajouter 4 autres champs horaires

Étape 2 — Améliorer l'apparence avec CSS

L'interface fonctionne pour l'instant sans mise en forme particulière.
Nous allons modifier son apparence avec le langage CSS.

Ouvrir le fichier style.css et **recopier** :

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f2f5;
    margin: 30px;
    text-align: center;
}

h1 {
    color: #176b91;
    font-size: 36px;
}

h2 {
    color: #444;
}

h3 {
    margin-top: 30px;
    color: #176b91;
}

input {
    padding: 8px;
    margin: 5px;
    border-radius: 5px;
    border: 1px solid #aaa;
}

button {
    background-color: #176b91;
    color: white;
    padding: 12px 20px;
    border: none;
    border-radius: 6px;
    cursor: pointer;
    margin: 5px;
}

button:hover {
    background-color: #104b66;
}

#etat {
    font-size: 22px;
    font-weight: bold;
}
```

Aide : comprendre le CSS

Une règle CSS est constituée d'un **sélecteur** et de **propriétés**.

Exemple :

```
h1 {
    color: blue;
    font-size: 30px;
}
```

Ce code indique que tous les titres <h1> doivent être bleus et mesurer 30 pixels.

Le symbole # permet de sélectionner un élément à partir de son identifiant.

Par exemple, #etat correspond à l'élément HTML possédant id="etat".

Travail à **réaliser** :

1. Enregistrer le fichier CSS.
2. Actualiser la page dans le navigateur.
3. Modifier la couleur du titre principal.
4. Modifier la couleur des boutons.
5. Augmenter la taille du texte indiquant l'état de la porte.

Étape 3 — Découvrir JavaScript (15 min)

Le HTML permet de créer des boutons, mais il ne suffit pas pour programmer leur comportement.
Pour cela, nous allons utiliser JavaScript.

Ouvrir le fichier script.js.

Recopier :

```
function ouvrir() {
    alert("Commande d'ouverture !");
}

function fermer() {
    alert("Commande de fermeture !");
}
```

Aide : comprendre une fonction

Une fonction regroupe plusieurs instructions qui peuvent être exécutées lorsqu'un événement se produit.

Exemple :

```
function bonjour() {
    alert("Bonjour !");
}
```

Lorsque la fonction bonjour() est appelée, une fenêtre affiche « Bonjour ! ».
Dans notre page, l'attribut onclick="ouvrir()" appelle la fonction ouvrir().

Travail à **réaliser** :

1. Actualiser la page.
2. Cliquer sur OUVRIR.
3. Cliquer sur FERMER.
4. Modifier les messages affichés pour indiquer « Ouverture demandée » et « Fermeture demandée ».

Étape 4 — Modifier le contenu de la page avec JavaScript

Nous souhaitons maintenant que l'état de la porte change directement sur la page, sans afficher de fenêtre.

Remplacer les deux fonctions précédentes par :

```
function ouvrir() {
    document.getElementById("etat").textContent =
        "Porte ouverte";
```

```
}

function fermer() {
    document.getElementById("etat").textContent =
        "Porte fermée";
}
```

Aide : comprendre le programme

L'instruction :

```
document.getElementById("etat")
```

permet de retrouver dans la page HTML l'élément qui possède l'identifiant etat.

La propriété :

```
.textContent
```

permet de modifier le texte affiché dans cet élément.

Travail à **réaliser** :

1. Tester les boutons OUVRIR et FERMER.
2. Vérifier que l'état de la porte change.
3. Modifier les textes pour afficher « Porte en cours d'ouverture » et « Porte en cours de fermeture ».

Attention : pour le moment, le programme simule le fonctionnement de la porte. Aucun moteur n'est commandé.

Étape 5 — Récupérer les horaires saisis

L'administrateur doit pouvoir programmer les heures d'ouverture et de fermeture.
Nous allons maintenant récupérer ces informations avec JavaScript.

Dans script.js, **ajouter** la fonction suivante :

```
function enregistrer() {

    let ouverture1 =
        document.getElementById("ouv1").value;

    let fermeture1 =
        document.getElementById("ferm1").value;

    document.getElementById("message").textContent =
        "Plage 1 : " + ouverture1 + " à " + fermeture1;
}
```

Aide : comprendre les variables

Une variable permet de mémoriser une information.

Exemple :

```
let heure = "08:30";
```

La variable heure contient ici le texte "08:30".

L'instruction :

```
document.getElementById("ouv1").value
```

récupère la valeur saisie dans le champ horaire identifié par ouv1.

Le symbole + permet ici d'assembler plusieurs textes.

Exemple :

```
let nom = "MEUNOT";
let message = "Bonjour " + nom;
```

La variable message contient alors "Bonjour MEUNOT".

Travail à **réaliser** :

1. Saisir 08:00 dans le premier champ d'ouverture.
2. Saisir 12:00 dans le premier champ de fermeture.
3. Cliquer sur Enregistrer les horaires.
4. Vérifier que les horaires apparaissent sous le bouton.

À vous de **compléter** le programme

Modifier la fonction enregistrer() pour récupérer également les horaires des plages 2 et 3.

Vous pouvez vous aider de cet exemple :

```
let ouverture2 =
    document.getElementById("ouv2").value;
```

Créer les variables manquantes :

- fermeture2
- ouverture3
- fermeture3

Modifier ensuite le message pour afficher les trois plages horaires.

Résultat attendu :

Plage 1 : 08:00 à 12:00

Plage 2 : 14:00 à 18:00

Plage 3 : 19:00 à 20:00

Indice : pour afficher plusieurs lignes, vous pouvez utiliser trois éléments HTML différents ou ajouter des retours à la ligne avec \n dans un élément utilisant white-space: pre-line en CSS.

Étape 6 — Vérifier le fonctionnement

Effectuer les tests suivants :

Test	**Résultat attendu**	**Validé**

Cliquer sur OUVRIR	Affichage du message d'ouverture	☐
Cliquer sur FERMER	Affichage du message de fermeture	☐
Saisir les horaires de la plage 1	Les valeurs sont récupérées	☐
Saisir les horaires de la plage 2	Les valeurs sont récupérées	☐
Saisir les horaires de la plage 3	Les valeurs sont récupérées	☐
Cliquer sur Enregistrer	Les trois plages sont affichées	☐

Étape 7 — Programmer votre propre interface

Votre mission : passer du croquis à la réalisation

Vous avez maintenant créé une première interface fonctionnelle en HTML, CSS et JavaScript.

Vous allez utiliser ce que vous venez d'apprendre pour **modifier son apparence afin qu'elle ressemble le plus possible à votre croquis initial**.

Vous pouvez modifier librement :

- Les couleurs de fond, des textes et des boutons.
- La taille des titres et des différents éléments.
- La forme des boutons (arrondis, rectangulaires, etc.).
- La disposition des plages horaires.
- L'emplacement des commandes OUVRIR et FERMER.
- Les espacements et les bordures.

Vous pouvez également ajouter des éléments graphiques simples, comme des encadrés, des titres supplémentaires ou des symboles.

Travail à **réaliser :**

1. Reprenez votre croquis réalisé en début de séance.
2. Repérez les différences entre votre croquis et l'interface actuellement affichée.
3. Modifiez progressivement le HTML et le CSS pour vous rapprocher de votre proposition.
4. Actualisez régulièrement la page pour observer les changements.
5. Vérifiez que les boutons et la programmation des horaires fonctionnent toujours.

Attention : ne modifiez pas les identifiants id des éléments HTML ni les noms des fonctions JavaScript, car cela pourrait empêcher le programme de fonctionner.

Pour aller plus loin — Travail facultatif

Si vous avez terminé, améliorez votre interface.

Défi 1 : afficher le texte « Porte ouverte » en vert et « Porte fermée » en rouge.

Défi 2 : vérifier qu'une heure de fermeture est postérieure à son heure d'ouverture.

Exemple :

```
if (ouverture1 >= fermeture1) {
    alert("Erreur dans les horaires !");
}
```

Cette comparaison fonctionne pour des heures au format HH:MM sur une même journée, lorsque les deux valeurs sont renseignées.

Défi 3 : modifier la présentation pour que l'interface soit agréable à utiliser sur un smartphone.