

Présentation du projet

Simulation d'un distributeur automatique de billets en HTML/CSS/JS. L'application permet de consulter son solde, de retirer de l'argent avec un algorithme de distribution optimale des billets (50 €, 20 €, 10 €), et d'afficher l'état du stock.

Fonctionnalités

- Consulter le solde courant
- Retirer un montant — algorithme glouton de distribution des billets
- Vérification : solde suffisant, billets disponibles, montant composable
- État des billets : tableau quantité / valeur par coupure
- Messages dynamiques : succès (vert), erreur (rouge), avertissement (jaune)
- Quitter : désactivation de tous les boutons

Architecture

Fichier	Rôle
TP_HTML.html	Structure de la page (header, menu, zones d'affichage, footer)
style.css	Mise en forme dark blue — palette CSS variables, responsive
script.js	Logique applicative — état, calculs, DOM, événements

FICHIER 1 — TP_HTML.html

■ *TP_HTML.html* — Structure HTML5 de la page

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="UTF-8">
<title>Distributeur de Billets</title>
<!-- Lien vers la feuille de style externe -->
<link rel="stylesheet" href="style.css">
</head>
<body>
<!-- En-tête de la page -->
<header>
<h1>■ Distributeur Automatique</h1>
<p>Projet BTS SIO – Simulation en HTML / CSS / JS</p>
</header>

<!-- Contenu principal -->
<main>
<!-- Menu principal avec les 4 fonctionnalités -->
<div class="menu">
<button id="btn-solde">Consulter le solde</button>
<button id="btn-retirer">Retirer de l'argent</button>
<button id="btn-etat">État des billets</button>
<button id="btn-quitte" class="danger">Quitter</button>
</div>

<!-- Zone affichée uniquement pour le retrait -->
<div id="zone-retrait" class="card" style="display:none;">
<label for="montant">Montant à retirer (€)</label>
```

```

<input id="montant" type="number" min="10" step="10" placeholder="Ex: 130">
<button id="btn-confirmer">Confirmer</button>
</div>

<!-- Zone pour afficher les messages (succès, erreurs, avertissements) -->
<div id="messages"></div>

<!-- Zone pour afficher l'état des billets sous forme de tableau -->
<div id="etat"></div>
</main>

<!-- Pied de page -->
<footer>
<p>© 2025 – Simulation pédagogique BTS SIO</p>
</footer>

<!-- Lien vers le fichier JavaScript -->
<script src="script.js"></script>
</body>
</html>

```

FICHIER 2 — style.css

■ style.css — Feuille de style

```

/* ----- Palette de couleurs ----- */
:root {
--blue-dark: #0a192f; /* Fond principal sombre */
--blue-mid: #112d4e; /* Fond intermédiaire */
--blue-light: #3f72af; /* Boutons et titres */
--blue-accent: #00b4d8; /* Accent lumineux */
--white: #f9f9f9; /* Texte clair */
--success: #06d6a0; /* Vert succès */
--error: #ef476f; /* Rouge erreur */
--warn: #ffd166; /* Jaune avertissement */
}

/* Style général du corps */
body {
font-family: 'Segoe UI', Arial, sans-serif;
background: var(--blue-dark);
color: var(--white);
margin: 0;
}

/* En-tête */
header {
background: var(--blue-mid);
padding: 20px;
text-align: center;
border-bottom: 3px solid var(--blue-accent);
}

h1 {
margin: 0;
color: var(--blue-accent);
}

/* Menu principal */
.menu {
margin: 20px;
display: flex;
gap: 12px;
flex-wrap: wrap;
justify-content: center;
}

```

```

/* Boutons */
button {
padding: 10px 15px;
border-radius: 8px;
border: none;
cursor: pointer;
font-weight: bold;
background: var(--blue-light);
color: var(--white);
transition: 0.3s;
}

/* Effet au survol */
button:hover {
background: var(--blue-accent);
color: var(--blue-dark);
}

/* Bouton quitter en rouge */
button.danger {
background: var(--error);
}

/* Zone de retrait */
#zone-retrait {
margin: 20px auto;
padding: 15px;
max-width: 400px;
background: var(--blue-mid);
border-radius: 10px;
box-shadow: 0 4px 10px rgba(0,0,0,.3);
}

/* Champ de saisie */
input {
padding: 8px;
border-radius: 6px;
border: 1px solid var(--blue-accent);
margin-right: 10px;
width: 150px;
}

/* Zone messages */
#messages {
margin: 20px;
text-align: center;
}

/* Styles des messages */
.status.ok { color: var(--success); font-weight: bold; }
.status.err { color: var(--error); font-weight: bold; }
.status.warn { color: var(--warn); font-weight: bold; }

/* Tableau état des billets */
#etat {
margin: 20px;
}

table {
width: 100%;
border-collapse: collapse;
margin-top: 10px;
background: var(--blue-mid);
border-radius: 8px;
overflow: hidden;
}

```

```

th, td {
border: 1px solid var(--blue-light);
padding: 10px;
text-align: center;
}

th {
background: var(--blue-light);
color: var(--white);
}

/* Pied de page */
footer {
background: var(--blue-mid);
text-align: center;
padding: 10px;
border-top: 3px solid var(--blue-accent);
margin-top: 30px;
}

```

FICHER 3 — script.js

■■■ script.js — Logique JavaScript

```

// ----- Initialisation -----
let solde = 1000;
let atm = {50: 10, 20: 20, 10: 30};
let actif = true;
const DENOMS = [50, 20, 10];

// ----- Utilitaires -----
function formatEuro(n) {
return new Intl.NumberFormat("fr-FR", { style: "currency", currency: "EUR", maximumFractionDigits: 0
}).format(n);
}

function setMessage(msg, type="ok") {
const div = document.createElement("div");
div.className = `status ${type}`;
div.innerHTML = msg;
const zone = document.getElementById("messages");
zone.innerHTML = "";
zone.appendChild(div);
}

function appendMessage(msg, type="ok") {
const div = document.createElement("div");
div.className = `status ${type}`;
div.innerHTML = msg;
document.getElementById("messages").appendChild(div);
}

function totalAtm() {
return DENOMS.reduce((s, b) => s + atm[b]*b, 0);
}

// ----- Fonctions principales -----
function consulterSolde() {
setMessage(`<strong>Solde actuel :</strong> ${formatEuro(solde)}`, "ok");
}

function consulterEtatBillets() {
let html = "<table><tr><th>Billet</th><th>Quantité</th><th>Valeur</th></tr>";
let total = 0;
DENOMS.forEach(b => {
const q = atm[b];

```

```

const val = q*b;
total += val;
html += `<tr><td>${b} €</td><td>${q}</td><td>${formatEuro(val)}</td></tr>`;
});
html += `<tr><td colspan="2"><strong>Total</strong></td><td>${formatEuro(total)}</td></tr></table>`;
document.getElementById("etat").innerHTML = html;
appendMessage("État des billets mis à jour.", "warn");
}

function retirerArgent() {
document.getElementById("zone-retrait").style.display = "block";
setMessage("Indiquez le montant à retirer puis cliquez sur Confirmer.", "warn");
}

function calculerDistribution(montant) {
let restant = montant;
const plan = {};
for (const b of DENOMS) {
const dispo = atm[b];
const maxUtilisable = Math.min(Math.floor(restant/b), dispo);
if (maxUtilisable > 0) {
plan[b] = maxUtilisable;
restant -= maxUtilisable*b;
}
}
return restant === 0 ? plan : null;
}

function confirmerRetrait() {
const val = Number(document.getElementById("montant").value);
if (!Number.isInteger(val) || val <= 0) {
setMessage("Montant invalide.", "err"); return;
}
if (val > solde) {
setMessage("Solde insuffisant.", "err"); return;
}
if (val > totalAtm()) {
setMessage("Distributeur insuffisant.", "err"); return;
}
const plan = calculerDistribution(val);
if (!plan) {
setMessage("Impossible de composer ce montant avec les billets disponibles.", "err"); return;
}
// Mise à jour
solde -= val;
for (const b of Object.keys(plan)) atm[b] -= plan[b];
const breakdown = Object.entries(plan).map(([b,q]) => `${q} × ${b} €`).join(" + ");
setMessage(`Retrait réussi : ${formatEuro(val)}<br>Billets donnés : ${breakdown}<br>Nouveau solde :
${formatEuro(solde)}`, "ok");
consulterEtatBillets();
document.getElementById("montant").value = "";
document.getElementById("zone-retrait").style.display = "none";
}

function quitterProgramme() {
actif = false;
setMessage("Programme terminé. Rafraîchissez la page pour redémarrer.", "warn");
document.querySelectorAll("button").forEach(btn => btn.disabled = true);
}

// ----- Événements -----
document.getElementById("btn-solde").onclick = consulterSolde;
document.getElementById("btn-etat").onclick = consulterEtatBillets;
document.getElementById("btn-retirer").onclick = retirerArgent;
document.getElementById("btn-confirmer").onclick

```

