Fiche de soutien - Bases de JavaScript et API REST

1. Les concepts fondamentaux

Variables et constantes

javascript

```javascript
// Variable (peut changer)
let compteur = 0;

// Constante (ne peut pas changer)
const apiUrl = "https://restcountries.com/v3.1/all";
```

Fonctions

javascript

```javascript
// Fonction classique
function afficherMessage(message) {
  console.log(message);
}

// Fonction fléchée (arrow function)
const afficherMessage = (message) => {
  console.log(message);
};
```

Sélection d'éléments du DOM

javascript

```javascript
// Par ID
const element = document.getElementById('monId');

// Par classe
const elements = document.getElementsByClassName('maClasse');

// Par sélecteur CSS
const element = document.querySelector('.maClasse');
const elements = document.querySelectorAll('.maClasse');
```

Manipulation du DOM

javascript

```javascript
// Créer un élément
const div = document.createElement('div');

// Définir des attributs
div.className = 'carte';
div.id = 'carte1';

// Ajouter du contenu
div.textContent = 'Titre de la carte';
// OU
div.innerHTML = '<h2>Titre de la carte</h2><p>Description</p>';

// Ajouter à un parent
document.body.appendChild(div);
```

2. Utilisation des API avec fetch

Structure de base

javascript

```javascript
fetch('https://api-url.com/data')
  .then(response => {
    if (!response.ok) {
      throw new Error('La requête a échoué');
    }
    return response.json();
  })
  .then(data => {
    console.log(data); // Traiter les données ici
  })
  .catch(error => {
    console.error('Erreur:', error);
  });
```

Avec async/await (plus moderne)

javascript

```javascript
async function fetchData() {
  try {
    const response = await fetch('https://api-url.com/data');

    if (!response.ok) {
      throw new Error('La requête a échoué');
    }

    const data = await response.json();
    console.log(data); // Traiter les données ici
  } catch (error) {
    console.error('Erreur:', error);
  }
}

fetchData();
```

3. Manipulation des tableaux

Méthodes utiles

javascript

```javascript
// filter - Crée un nouveau tableau avec les éléments qui passent le test
const nombresPairs = nombres.filter(n => n % 2 === 0);

// map - Crée un nouveau tableau en transformant chaque élément
const doubles = nombres.map(n => n * 2);

// sort - Trie les éléments du tableau
nombres.sort((a, b) => a - b); // Tri croissant
nombres.sort((a, b) => b - a); // Tri décroissant

// slice - Extrait une portion du tableau
const premiersCinq = tableau.slice(0, 5); // Du premier au cinquième élément
```

4. Événements

Ajouter un écouteur d'événement

javascript

```javascript
// Syntaxe de base
element.addEventListener('click', function() {
  console.log('Element cliqué!');
});

// Avec une fonction nommée
function handleClick() {
  console.log('Element cliqué!');
}
element.addEventListener('click', handleClick);

// Événements courants:
// click, input, change, submit, keyup, keydown, mouseover, mouseout
```

5. Exemples spécifiques pour le projet REST Countries

Recherche dans un tableau

javascript

```javascript
function filterByName(countries, searchTerm) {
  return countries.filter(country =>
    country.name.common.toLowerCase().includes(searchTerm.toLowerCase())
  );
}
```

Tri des pays par nom

javascript

```javascript
function sortCountriesByName(countries, ascending = true) {
  return [...countries].sort((a, b) => {
    if (ascending) {
      return a.name.common.localeCompare(b.name.common);
    } else {
      return b.name.common.localeCompare(a.name.common);
    }
  });
}
```

Formater un nombre avec des séparateurs

javascript

```javascript
function formatNumber(num) {
  return num.toString().replace(/\B(?=(\d{3})+(?!\d))/g, " ");
}

// Exemple: 1234567 devient "1 234 567"
```