

TP – QR Codes & Codes-barres en HTML / JavaScript

Objectifs

- Distinguer QR Code (2D) et code-barres (1D)
- Générer des QR Codes pour différents usages normalisés
- Générer des codes-barres standards (EAN-13, Code 128)
- Comprendre les **formats de données** (vCard, iCalendar, EAN)
- Manipuler le DOM et des librairies JavaScript

Rappel :

- Un QR Code est une image générée à partir d'une chaîne de caractères
- Scanner ≠ interpréter (le téléphone interprète le contenu)

Prérequis

- HTML / CSS de base
- Bases JavaScript
- Navigateur web moderne
- Connexion Internet (CDN)

Organisation du projet

Le projet devra respecter l'arborescence suivante :

```
qr_event_generator_web/
├── index.html          # Page principale du TP
└── assets/
    ├── css/
    │   └── style.css    # Styles de la page
    ├── js/
    │   └── script.js    # JavaScript du TP
    └── img/              # (optionnel) images ou exports
```

Le fichier **index.html** contiendra la structure HTML. Le CSS sera placé dans le dossier **css**. Le JavaScript sera placé dans le dossier **js**.

Dans le rendu final, tout le JavaScript devra être placé dans **assets/js/script.js**.

Code HTML de base

- A compléter / comprendre la logique, pas copier-coller

index.html

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>TP QR Codes & Codes-barres</title>

    <!-- Librairie QR Code -->
    <script src="https://cdn.jsdelivr.net/npm/qrcodejs/qrcode.min.js">
    </script>

    <!-- Librairie Code-barres -->
    <script
src="https://cdn.jsdelivr.net/npm/jsbarcode@3.11.5/dist/JsBarcode.all.min.js"></script>

    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 30px;
        }
        .box {
            margin-bottom: 30px;
        }
    </style>
</head>
<body>

    <h1>TP – QR Codes & Codes-barres</h1>

    <div class="box">
        <h2>Zone de génération</h2>
        <div id="qrcode"></div>
        <svg id="barcode"></svg>
    </div>

    <script>
        // Le JavaScript sera ajouté ici
    </script>

</body>
</html>
```

Code JavaScript

```
/*
=====
 FONCTION DE GÉNÉRATION QR CODE
=====
 */
```

```
function generateQRCode(text) {
    document.getElementById("qrcode").innerHTML = "";
    new QRCode(document.getElementById("qrcode"), {
        text: text,
        width: 200,
        height: 200
    });
}

// Appel de la fonction
generateQRCode("BTS CIEL – Carnus");
```

Livrable attendu pour chaque partie :

- Un QR Code visible dans la page
 - Le code JavaScript commenté
-

Partie 1 – QR Code simple

Créer une page HTML qui génère un QR Code contenant le texte :

```
"TP CIEL-1 : Web – QR Code"
```

- Le QR Code devra mesurer **200 × 200 px**.

Partie 2 – QR Code lien Internet

Modifier le QR Code pour qu'il ouvre le site suivant après scan :

```
https://www.carnus.fr
```

Partie 3 – QR Code dynamique

Créer un formulaire permettant :

- la saisie d'un texte
- la génération dynamique du QR Code (un bouton **Générer**)

Le QR Code devra afficher le contenu saisi par l'utilisateur.

Partie 4 – QR Code Contact (vCard)

Créer un QR Code permettant d'ajouter automatiquement le contact suivant :

- Prénom : Charles

- Nom : Carnus
- Téléphone : 0565733700
- Email : lycee@carnus.fr
- Société : Carnus Enseignement Supérieur

Le format vCard doit respecter strictement la norme :

```
BEGIN:VCARD  
VERSION:3.0  
...  
END:VCARD
```

Partie 5 – QR Code Événement (Agenda)

Créer un QR Code permettant d'ajouter un événement à l'agenda :

- Titre : Portes ouvertes
- Lieu : Salle 215
- Date : 31 janvier 2026
- Heure : 09h00 → 16h30

Les dates sont au format AAAAMMJJTHHMMSS (ex : 20260119T153000) Le format utilisé est iCalendar (.ics) Si une ligne est mal écrite, l'événement ne sera pas reconnu.

Bonus

- Ajouter un bouton pour télécharger le QR Code

Indice : utiliser `canvas.toDataURL()` ou clic droit → enregistrer.

Partie 6 – Découverte des codes-barres

Introduction

Les **codes-barres** sont différents des QR Codes :

- ils sont **unidimensionnels (1D)**
- ils contiennent des données **très structurées**
- chaque format impose **des règles strictes**

Contrairement aux QR Codes, on **ne peut pas mettre n'importe quel texte** dans un code-barres.

Objectifs :

- générer un code-barres avec JavaScript
- choisir le **format adapté** (Code 128, EAN-13)
- comprendre pourquoi certaines valeurs sont refusées
- manipuler une librairie JavaScript externe

Librairie utilisée : JsBarcode

- Librairie JavaScript professionnelle très utilisée
- Génère des codes-barres en :
 - SVG
 - Canvas
 - Image

Formats supportés (extraits) :

- Code 128
- EAN-13 / EAN-8
- UPC
- Code 39
- ITF
- MSI

Intégration de la librairie

Dans le fichier `index.html` :

```
<script  
src="https://cdn.jsdelivr.net/npm/jsbarcode@3.11.5/dist/JsBarcode.all.min.  
js"></script>
```

Rappel théorique

QR Code	Code-barres
2D	1D
Texte libre	Format strict
URL, vCard, événement	Numérique / alphanumérique
Forte capacité	Capacité limitée

Un QR Code peut contenir un contact ou un événement, un code-barres non.

Exemple 1 – Code-barres Code 128

Le format **Code 128** accepte :

- lettres
- chiffres
- symboles

HTML

```
<svg id="barcode128"></svg>
```

JavaScript

```
JsBarcode("#barcode128", "TPWEB2026", {  
    format: "CODE128",  
    width: 2,  
    height: 80,  
    displayValue: true  
});
```

Le texte **TPWEB2026** est affiché sous le code-barres.

Exemple 2 – Code-barres EAN-13

Le format **EAN-13** impose :

- **13 chiffres obligatoires**
- aucun caractère alphabétique

HTML

```
<svg id="barcodeEAN"></svg>
```

JavaScript

```
JsBarcode("#barcodeEAN", "978020137962", {  
    format: "EAN13",  
    width: 2,  
    height: 80,  
    displayValue: true  
});
```

Si la valeur ne contient pas exactement **13 chiffres**, le code-barres ne s'affiche pas.

Remarque

Dans les exemples précédents, le code JavaScript est très similaire.

- seul le **format** et la **valeur** changent
- cela permet de comprendre le rôle de chaque paramètre

Dans la suite, nous allons **regrouper ce code dans une fonction**.

Fonction de génération

Dans le fichier `assets/js/script.js` :

```
/*
=====
 FONCTION DE GÉNÉRATION CODE-BARRES
=====
 */

function generateBarcode(value, format) {
  JsBarcode("#barcode", value, {
    format: format,
    width: 2,
    height: 80,
    displayValue: true
  });
}
```

Exemple 3 – Code-barres dynamique

HTML

```
<input type="text" id="code" placeholder="Valeur du code">
<select id="format">
  <option value="CODE128">Code 128</option>
  <option value="EAN13">EAN-13</option>
</select>
<button onclick="generate()">Générer</button>

<svg id="barcode"></svg>
```

JavaScript

```
function generate() {
  const value = document.getElementById("code").value;
  const format = document.getElementById("format").value;

  generateBarcode(value, format);
}
```

Règles importantes à respecter

Format	Contraintes
Code 128	Texte libre

Format	Contraintes
EAN-13	13 chiffres uniquement

- Si les règles ne sont pas respectées :
 - le code-barres ne s'affiche pas
 - ou la librairie génère une erreur

Manipulations

1. Code 128

Générer un code-barres **Code 128** contenant :

TP2026CIEL

2. EAN-13

Générer un code-barres **EAN-13** avec la valeur :

978020137962

3. Interface utilisateur

Créer une interface permettant de :

- saisir une valeur
- choisir le type de code-barres
- générer le code correspondant

Tester avec :

- 978020137962 (EAN-13)
- TPWEB2026 (Code 128)

Partie 7 – Générateur universel (QR + Barres)

Créer une interface permettant de :

- choisir le type (QR / Code-barres)
- choisir le format
- saisir une valeur
- générer le code correspondant

Partie 8 - Questions

- Pourquoi un QR Code contact fonctionne sans application spécifique ?
 - Quelle est la différence entre un QR Code URL et texte ?
 - Que se passe-t-il si le format vCard est incorrect ?
-