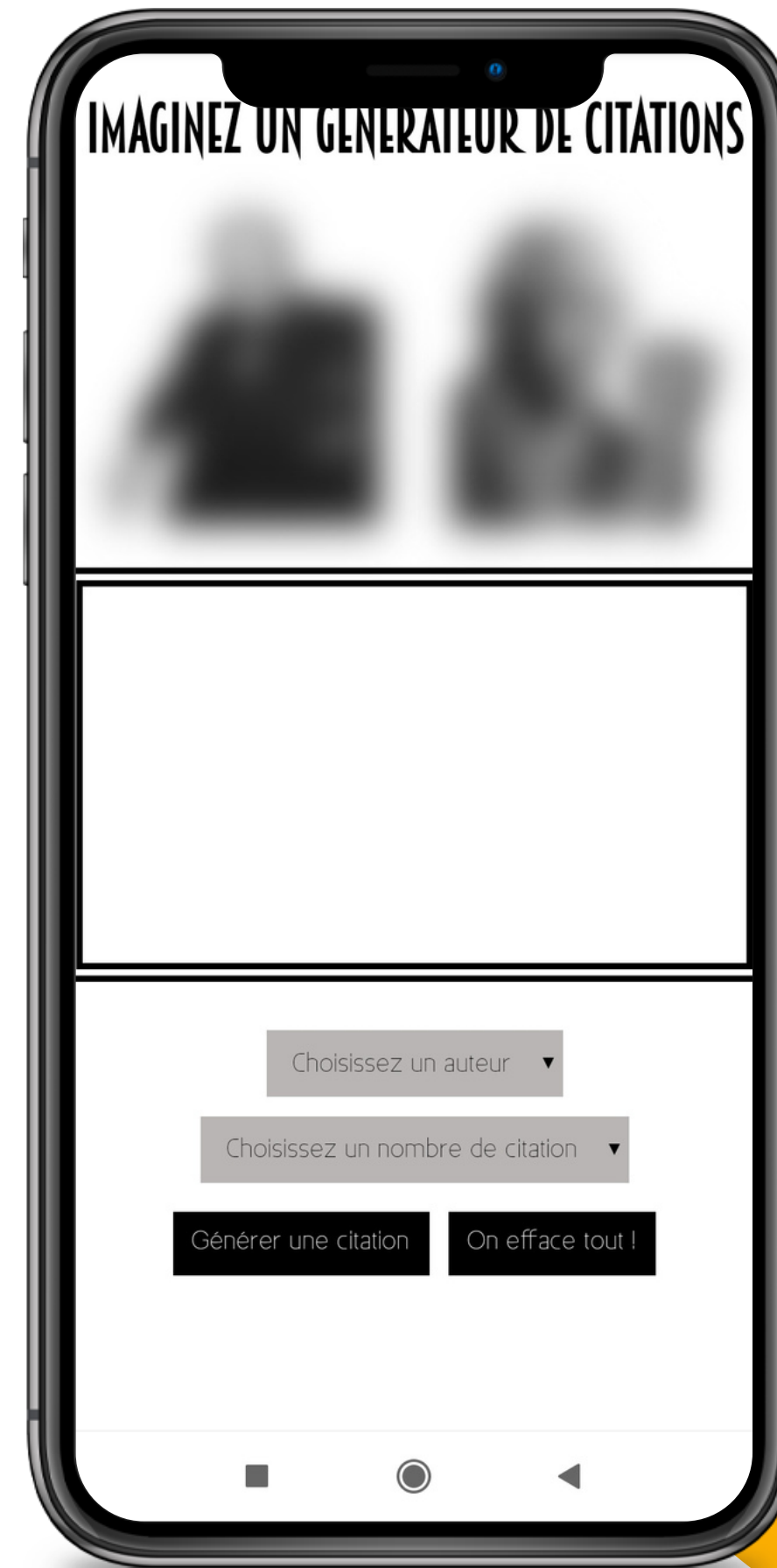


Apprentissage de JavaScript

Imaginez un générateur de citations

Projet 5 OpenClassrooms
Charles-Boris DAUVIER



Dans cette présentation

Voici ce que nous allons
couvrir :

Le cadre du projet :

.....

Les objectifs

.....

Les spécifications exigées

.....

L'application :

.....

Démonstration

.....

La structure HTML / CSS

.....

Le JavaScript

Les objectifs :

- Apprendre la programmation et utiliser des algorithmes avec JavaScript
- Créer un générateur de citations
- S'inspirer du Pipotron et du Vandammotron
- Afficher les phrases générées

Les spécificités :

- Combiner au minimum 3 morceaux de phrase
- Choix du nombre de citations générées (de 1 à 5)
- Choix entre deux types de générateur
- Proposer une nouvelle génération de citation
- Effacer les citations générées

Démonstration

de l'application "Générateur de citations".



La structure HTML

```
<body>
  <div class="container">
    <h1>Imaginez un générateur de citations</h1>
    <div>
      <div class="containerDisplayQuoteWithImg">
        <div class="displayQuoteWithImg--img">
          
          
        </div>
        <div class="displayQuoteWithImg--quote" id="containerQuoteGenerate"></div>
      </div>
    </div>
  </div>
```

- Le 1er conteneur de la page
- Contient le H1 de la page
- Contient la div ou seront injectées les citations
- Contient les images des auteurs

La structure HTML

```
<div class="containerQuoteOptionsWithBtn">

  <div class="quoteOptions">
    <select class="quoteOptions--author selectCustom" name="quoteAuthor" id="quoteAuthorSelect">
      <option value="empty">Choisissez un auteur</option>
      <option value="winstonC">Winston Churchill</option>
      <option value="albertE">Albert Einstein</option>
    </select>

    <select class="quoteOptions--number selectCustom" name="quoteNumber" id="quoteNumberSelect">
      <option value="empty">Choisissez un nombre de citation</option>
      <option value="1">1</option>
      <option value="2">2</option>
      <option value="3">3</option>
      <option value="4">4</option>
      <option value="5">5</option>
    </select>
  </div>

  <div class="buttonQuote">
    <button class="buttonQuote--generate btnCustom" id="btnGenerate">Générer une citation</button>
    <button class="buttonQuote--delete btnCustom" id="btnDelete">On efface tout !</button>
  </div>
</div>
```

- Le 1er élément Select pour choisir l'auteur
- Le 2ème élément Select pour choisir le nombre de citations
- Le 1er bouton pour générer une citation
- Le 2ème bouton pour effacer les citations affichées

La structure CSS

- Morcellement du CSS en 2 fichiers

style.css

- Contient le style générale de la page
- Contient les media queries pour que l'application soit responsive

components.css

- Contient le code des composants

```
index.html x style.css x components.css x main.js x
1 @font-face {
2   font-family: 'retroTypoTitle';
3   src: url(../font/breamcatcher.ttf);
4 }
5
6 @font-face {
7   font-family: 'retroTypo';
8   src: url(../font/bakersfield_light.ttf);
9 }
10
11 body {
12   margin: 0;
13   padding: 0;
14 }
15
16 body, select, button {
17   font-family: 'retroTypo', Arial, Serif;
18 }
19
20 .container {
21   padding-right: 15px;
22   padding-left: 15px;
23   margin-right: auto;
24   margin-left: auto;
25 }
26
```

```
x.html x style.css x components.css x main.js x Quote
Cc W .* 0 results
.btnCustom {
  padding: 10px;
  border: none;
  background-color: black;
  color: white;
  cursor: pointer;
  transition: background-color 300ms ease-in-out;
}
.btnCustom:hover {
  background-color: white;
  color: black;
  border: 1px solid black;
}
.selectCustom {
  background-color: #b9b4b4fc;
  padding: 10px;
  border: none;
}
```


Le JavaScript

Utilisation de la programmation orientée objet (POO) :

- Utiliser des objets pour modéliser les objets du monde réel que l'on souhaite représenter dans nos programmes
- Fournir un moyen simple d'accéder à une fonctionnalité qu'il serait difficile d'utiliser autrement.
- À l'intérieur de l'objet, appelé aussi classe, est défini un ensemble de données et d'actions

main.js

Création de 3 variables qui contiennent des tableaux et bouts de citations

À partir de la classe, création d'un objet qui contient les données et attributs définis dans la classe.

```
main.js x Quote.js x Generate.js x
1 // ===== Tableau citations Churchill =====...
2
3 let startQuoteChurchill = [
4     "La critique peut être désagréable mais elle est nécessaire.",
5     "On ne devrait jamais tourner le dos à un danger pour tenter de le fuir.",
6     "Les chiens vous regardent tous avec vénération."
7 ];
8
9 let middleQuoteChurchill = [
10     "Elle est comme la douleur pour le corps humain.",
11     "Si vous le faites, vous le multipliez par deux.",
12     "Les chats vous toisent tous avec dédain."
13 ];
14
15 let endQuoteChurchill = [
16     "Elle attire l'attention sur ce qui ne va pas.",
17     "Mais si vous l'affrontez rapidement et sans vous dérober, vous le réduirez de moitié.",
18     "Il n'y a que les cochons qui vous considèrent comme leurs égaux."
19 ];
20
```

```
// Instanciation des deux classes
let churchillQuote = new Quote(startQuoteChurchill, middleQuoteChurchill, endQuoteChurchill);
let einsteinQuote = new Quote(startQuoteEinstein, middleQuoteEinstein, endQuoteEinstein);
let generate = new Generate(churchillQuote, einsteinQuote);
```

Quote.js

- Création de la classe Quote
- Constructor appelé lors de l'instanciation
- Méthode qui génère aléatoirement une citation

```
n.js x Quote.js x Generate.js x
class Quote {
  constructor(startQuote, middleQuote, endQuote) {
    this.startQuote = startQuote;
    this.middleQuote = middleQuote;
    this.endQuote = endQuote;
  }

  // Méthode qui génère une citation aléatoire
  randomQuote(generateAuthor, quoteNumber) {
    for (let i = 0; i < quoteNumber; i++) {
      this.quoteGenerate = generateAuthor.startQuote[Math.floor(Math.random()*generateAuthor.startQuote.length)]
        + ' ' +
        generateAuthor.middleQuote[Math.floor(Math.random()*generateAuthor.middleQuote.length)]
        + ' ' +
        generateAuthor.endQuote[Math.floor(Math.random()*generateAuthor.endQuote.length)];

      this.paraQuoteGenerate = document.createElement('p');

      this.paraQuoteGenerate.innerHTML = this.quoteGenerate;

      document.getElementById('containerQuoteGenerate').appendChild(this.paraQuoteGenerate);
    }
  }
}
```

- Insertion de la citation dans l'élément #containerQuoteGenerate

Generate.js

- Méthode pour récupérer le nom de l'auteur
- Récupération de la valeur de l'élément Select
- Conditions et actions en fonction de la valeur

```
class Generate {
  constructor(generateChurchill, generateEinstein) {
    this.generateChurchill = generateChurchill;
    this.generateEinstein = generateEinstein;
    this.events()
  }

  // Méthode pour récupérer la valeur de l'auteur
  getQuoteAuthor() {
    this.authorValue = document.getElementById("quoteAuthorSelect").value;

    if (this.authorValue === "empty") {
      alert("Vous n'auriez pas oublié de choisir un auteur par hasard ?! ;-)");
    } else {
      return this.authorValue;
    }
  }
}
```

Generate.js

- Méthode events lancée au moment de l'instanciation
- Condition pour générer la citation en fonction de l'auteur choisi

```
// Méthode qui intègre tous les événements
events() {
  document.getElementById( elementId: "btnGenerate").addEventListener( type: 'click', listener: () => {

    this.deleteQuote()
    this.getQuoteAuthor();
    this.getQuoteNumber();

    if (this.authorValue === "winstonC") {
      console.log(this.authorValue);
      churchillQuote.randomQuote(this.generateChurchill, this.quoteNumber);
    } else if (this.authorValue === "albertE") {
      einsteinQuote.randomQuote(this.generateEinstein, this.quoteNumber);
    }
  });

  document.getElementById( elementId: "btnDelete").addEventListener( type: 'click', listener: () => {
    this.deleteQuote();
  });
}
```

Generate.js

```
// Modifie le style de l'image en fonction du choix de l'auteur
let authorName = document.getElementById( elementId: "quoteAuthorSelect");
const einsteinImg = document.getElementById( elementId: "einsteinImg");
const churchillImg = document.getElementById( elementId: "churchillImg");

authorName.addEventListener( type: 'change', listener: () => {

    switch (authorName.value) {
        case "empty":
            einsteinImg.style.filter = "blur(10px)";
            churchillImg.style.filter = "blur(10px)";
            break;
        case "winstonC":
            einsteinImg.style.filter = "blur(10px)";
            churchillImg.style.filter = "blur(0)";
            break;
        case "albertE":
            einsteinImg.style.filter = "blur(0)";
            churchillImg.style.filter = "blur(10px)";
            break;
    }
});
```

- Animation des images en fonction du choix de l'auteur
- Ajout et modification du style, la fonction CSS filter est utilisée dans ce cas.



Merci pour votre attention.

Avez-vous des questions ?

