Data binding

### Création d'un projet

En vous aidant du guide suivant: [Quick Start - Without build tools](<https://vuejs.org/guide/quick-start.html#without-build-tools>)

- Chargez la librairie Vue depuis un fichier externe, comme vous le feriez pour jQuery (utilisation d’un CDN)

- Créez directement dans un fichier HTML votre première application Vue.

- Rendez votre application réactive : affichez du texte qui sera renseigné en Javascript, dans l'objet `data` de votre instance de Vue.

Form binding

## Fonctionnalités

Ajoutez à votre page, dans des blocs séparés :

1. Un champ de saisie input dont la valeur est affichée en direct dans une balise H1.

2. De nouveau, un input dont la valeur est affichée dans une balise H1, mais cette fois ci dont la mise à

jour au \*\*clic\*\* sur un bouton et pas en direct.

<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>

<div id="app">

  <h1>{{ message }}</h1>

  <input v-model="message1" placeholder="edit me" />

  <button @click="AssignNewValue">Changer la valeur du header</button>

</div>

<script>

  Vue.createApp({

    data() {

      return {

        message: 'Hello Vue!',

        message1: ''

      }

    },

    methods: {

      AssignNewValue() {

        this.message = this.message1;

      }

    }

  }).mount('#app')

</script>

Le CSS :

<style>

  button {

    width: 300px;

    padding: 10px;

    border: none;

    border-radius: 5px;

    background: #cc7a00;

    font-size: 16px;

    color: #fff;

    cursor: pointer;

  }

  button:hover {

    background: #ff9800;

  }

  @media (min-width: 568px) {

    .name-item,

    .city-item {

      display: flex;

      flex-wrap: wrap;

      justify-content: space-between;

    }

    .name-item input,

    .name-item div {

      width: calc(50% - 20px);

    }

    input {

      box-shadow: inset #abacaf 0 0 0 2px;

      border: 0;

      background: rgba(0, 0, 0, 0);

      appearance: none;

      width: 50%;

      position: relative;

      border-radius: 3px;

      padding: 9px 12px;

      line-height: 1.4;

      color: rgb(0, 0, 0);

      font-size: 16px;

      font-weight: 400;

      height: 40px;

      transition: all .2s ease;

      :hover {

        box-shadow: 0 0 0 0 #fff inset, #1de9b6 0 0 0 2px;

      }

      :focus {

        background: #fff;

        outline: 0;

        box-shadow: 0 0 0 0 #fff inset, #1de9b6 0 0 0 3px;

      }

    }

    h1 {

      padding: 60px;

      text-align: center;

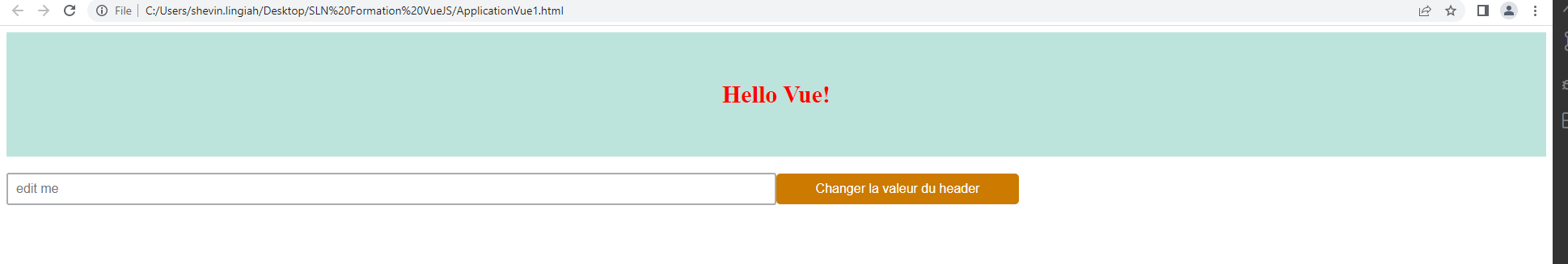
      background: #bde4dc;

      color: rgb(255, 0, 0);

      font-size: 30px;

    }

</style>

Resultat :   
  
Avant :   
  


Graphical user interface, text, application

Description automatically generated

Après :

Text

Description automatically generated with medium confidence