Validation des formulaires

Dr Bakary Diarra
Cité collégiale

Plan de la présentation

- Accès aux champs du formulaire
- Récupération des noms et valeurs
- Validation des valeurs
- Affichage des messages d’erreur
- Communication avec un serveur (base de données)

Accès aux champs

Pour accéder aux champs d’un formulaire en javascript

- En utilisant des attributs (id, name,…), assez long
- En utilisant l’attribut name du formulaire
  - ✓Récupérer les champs par déstructuration
  - ✓Idem pour les valeurs
  - ✓Appliquer les validations nécessaires
  - ✓Envoyer au serveur si tout est valide
  - ✓Sinon afficher les messages adéquats

Formulaires

Manipulation des champs

```
<form action="/destination" name="inscription">
  <div>
      <label for="nom">Nom</label>
      <input type="text" name="nom" id="nom" />
   </div>
  <div>
      <label for="email">Email</label>
      <input type="email" name="email" id="email"/>
   </div>
  <div>
       <select size="4" name="choix" id="choix">
              <option value="1">Math</option>
              <option value="2">Physique</option>
        </select>
   </div>
   <button type="submit">Envoyer</button>
</form>
```

```
//Formulaire

const form = document.forms.inscription

//Récupérer les champs

const {nom, prenom, email, choix = form

const champs = {nom, prenom, email, choix}

//Récupérer les valeurs des champs

const valeurs = { nom:nom.value,
prenom:prenom.value, email:email.value,
choix:choix.value }
```

Formulaires

Validation des valeurs du champ

- Utilisation des expressions rationnelles (regex)

```
//Forme générale de regex

const nomRgex = /^[a-zA-Z]$/   // Uniquement des caractères alpha

//Validation d'un champ avec test de regex

if(!nomRegex.test(nom))       // Retourne un booléen
```

- Utilisation d'autres règles

Formulaires

Erreur de validation des champs

- Un objet avec le même contenu que les valeurs

```
//Objet des erreurs

const erreur = { nom:null, prenom:null, email:null, choix:null }

// Changer l'erreur correspondant au champ courant, utiliser des messages
convenables

if(!nomRegex.test(nom)) erreur.nom = "Le nom doit etre ..."
```

- Réinitialiser l'erreur dès que la valeur du champ est correcte
- Créer une div pour afficher les erreurs

Formulaires

Affichage des messages d’erreur

```html
<form action="/destination" name="inscription">
  <div>
      <label for="nom">Nom</label>
      <input type="text" name="nom" id="nom" />

      <div id="nom-erreur"> </div>

   </div>
  <div>
      <label for="email">Email</label>
      <input type="email" name="email" id="email"/>

      <div id="email-erreur"> </div>

   </div>
   <button type="submit">Envoyer</button>
</form>
```

```javascript
// Affichage de l’erreur

for (let champ in erreur){

   const divErreur = document
           .querySelector(`#${champ}-erreur`)

   if(erreur[champ]{

     // Mettre l’erreur dans la div
     divErreur.innerHTML= erreur[champ]
     // Mettre l’erreur en rouge
     divErreur.style.color= "red"

   } else divErreur.innerHTML= ""

}
```

Communication avec le serveur

- La communication avec le serveur se fait avec fetch ou xhr
- Cela permet d'envoyer des données au serveur
- Lire également des données du serveur pour l'affichage

```
// Une fois tous les champs validés
   const valeurs = {nom, prenom, email, choix}

//Exemple d'envoi

 fetch('https://example.org/post',{
         method: "POST",
         body: JSON.stringify(valeurs),
         ...})
    .then(res=>res.json())
    .then(data=>console.log(data)
    .catch(err=>console.log(err))
```

Communication avec le serveur

- La communication avec le serveur se fait avec fetch ou xhr
- Cela permet d'envoyer des données au serveur
- Lire également des données du serveur pour l'affichage

```
//Exemple de lecture

 fetch('https://api.publicapis.org/entries')
    .then(res=>res.json())

    .then(resultatFinal =>console.log(resultatFinal)

    .catch(err=>console.log(err))
```

- On peut afficher ces données sur une page web avec le

Template littéral

- Utilisation du string littéral
- Facilité d'insertion de contenu
- Tenir des variables dynamiques
- Convenable pour des grandes insertions
- Représentation des données d'un serveur

```
//Présentation du code
 <div id="exo">

     <h1>Welcome, dear learner!</h1>

     <p>Follow correctly this tutorial </p>

 </div>
```

```
const parent = document.querySelector('#exo')
const paraText="Juste un essai!"

//Nouveau contenu
const contenu =
    `<p class="autre">${paraText}</p>
    <h3 class="head-h3">Autre titre</h3>
    <p class="autre">Autre valeur</p>`

parent.innerHTML+=contenu
```

Conclusion

- Apprendre à récupérer les données des formulaires
- Valider les données et afficher des messages d’erreur
- Envoyer des données au serveur
- Lire des données depuis un serveur

Exercices

- Valider , récupérer et envoyer les données du formulaire donné dans la console pour vérification
- Créer une page avec la liste des données provenant du serveur (url) donne en utilisant le template littéral