

VALIDATION DE FORMULAIRE avec JS

AVEC FORMS[] et Expressions régulières

Note : réalisé avant mon entrée en formation

Objectif : valider les champs d'un formulaires en contrôlant que :

- Les champs obligatoires ne soient pas vides,
- Les données correspondent aux formats voulus,
- Tant que tous les champs ne sont pas validés, le formulaire est bloqué,
- Lorsque les champs sont validés : envoi du formulaire.

1/ récupération du formulaire avec forms[].

```
const formInscr = document.forms['connect'];
```

2/ récupération du éléments du formulaire.

```
const userFirstname = formInscr['firstname'];
const userName = formInscr['name'];
const userMail = formInscr['mail'];
const userPseudo = formInscr['pseudo'];
```

3/ on déclare des const avec des Regex pour le format des données souhaitées

```
const regName = /^[a-zA-ZéèîïÉÊÎÏ]{2,}([-'\s][a-zA-ZéèîïÉÊÎÏ]+)?$/;
const regName2 = /^[a-zA-ZéèîïÉÊÎÏ][a-zéèâçîï]+([-'\s][a-zA-ZéèîïÉÊÎÏ][a-zéèâçîï]+)?$/;
const regMail = /^([\w-\.]*)@((?:[\w]+\.)+)([a-zA-Z]{2,4})/i;
const regPseudo= /^(.){3,10}$/;
```

4/ on déclare une variable pour la gestion des erreurs.

```
let error;
```

5/ mise en place d'un écouteur d'évènement lors de la soumission du formulaire

Lors de la soumission on récupère la valeur du champ et on supprime les éventuels espaces avant/après avec la méthode trim()

```
formInscr.addEventListener('submit',function(e){
  const userFirstnameValue = userFirstname.value.trim(); //suppression des espaces.
  const userNameValue = userName.value.trim();
  const userMailValue = userMail.value.trim();
  const userPseudoValue = userPseudo.value.trim();
```

6/ Les fonctions

On appellera des fonctions de contrôles :

`isEmpty(formInscr)`; fonction qui vérifie que tous les champs sont remplies
`validateRegex(userFirstname,userFirstnameValue,regName)`; la fonction validateRegex permet de contrôler la validation des données par rapport à une regex de référence. Ses 3

paramètres sont le nom de section HTML concerné, la valeur du champ entrée par l'utilisateur (la valeur à contrôler) , la regex de référence pour réaliser le contrôle.

```
isEmpty(formInscr);
validateRegex(userFirstname,userFirstnameValue,regName);
validateRegex(userName,userNameValue,regName);
validateRegex(userMail,userMailValue,regMail);
```

La fonction isEmpty contrôle que tous les champs du formulaire soient remplis

```
function isEmpty(form){
    let NbInputs = form.length-
1; // on compte tous les champs du formulaire moins le bouton
    let nbInputNonVide=0; // pour verifier que tous les champs soient contrôlés
    const optionnalInputs = ""; // si des champs son optionnels on saute la condition qui
demande a remplir le champ et on passe a la validation dans le else

    //---- CONTROLE GENERIQUE sur champs vides----//
    for(i=0; i<NbInputs; i++){
        if(form[i].value == "" && !optionnalInputs){
            controlMsg(form[i],'!! veuillez remplir ce champ !!',false);
            error = true;
        }else{
            nbInputNonVide++
            controlMsg(form[i],'',true);
        }
    }
    //si le nb de champs non vide est = au nb de champs du formulaire alors verif OK
    if(nbInputNonVide == NbInputs){
        error = false; // Les champs sont remplis pas d'erreur : on peut contrôler les don
nées
    }
}
```

On contrôle les données en les comparant au regex établis .

Pour les champs faisant appel au regex, la fonction isEmpty peut sembler en doublon car dans la fonction validateRegex un contrôle des champs vide et de nouveau fait.

Cependant avec isEmpty si un input est rajouté et qu'il ne fait pas de contrôle avec un regex alors le contrôle du champs vide sera quand même fait grâce à isEmpty (si on ne le déclare pas comme optionnel)

De plus isEmpty incrémente a un compteur lorsque celui-ci est égal au nombre de champ du formulaire alors il valide ce contrôle et on passe à la fonction suivante

```
function validateRegex(input,inputValue,regex){
    if(!regex.test(inputValue)){
        error=true;
        controlMsg(input, "Le format n'est pas correct",false);
    }
    if(inputValue == ""){
        controlMsg(input, "Veuillez remplir ce champ",false);
    }
}
```

```

        error=true;
    }
}

```

7/ Contrôle de données sans Regex

Si un champs n'a pas de de regex , alors on peut rajouter une vérification « classique » par exemple On rajoute le champ pseudo qui accepte n'importe quel caractères mais qui doit avoir une longueur comprise entre 3 et 10 caractères,

La vérification peut se faire ainsi :

A la suite des contrôle regex.

```

if(userPseudoValue && (userPseudoValue.length <3 || userPseudoValue.length >10)){
    controlMsg(userPseudo,'entre 3 et 10 caractères attendus',false);
    error=true;
}

```

Si on veut se passer totalement du isEmpty nous pouvons également contrôler le champ vide

```

if(userPseudoValue && (userPseudoValue.length <3 || userPseudoValue.length >10)){
    controlMsg(userPseudo,'entre 3 et 10 caractères attendus',false);
    error=true;
}else if(userPseudoValue == ''){
    controlMsg(userPseudo,'Merci de remplir ce champ',false);
    error=true;
}

```

Enfin on bloque le formulaire en cas de défaut ou on l'envoie

```

// validation finale envoi du formulaire
if(error){
    e.preventDefault();
    alert('Formulaire non envoyé : Merci de vérifier les données renseignés')
}else{
    alert('Formulaire envoyé');
}

```

Version 1 avec isEmpty

```
const formInscr = document.forms['connect'];
const userFirstname = formInscr['firstname'];
const userName = formInscr['name'];
const userMail = formInscr['mail'];
const userPseudo = formInscr['pseudo'];
const regName = /^[a-zA-ZéèïîÉÊÎÏ]{2,}([-'\s][a-zA-ZéèïîÉÊÎÏ]+)?$/;
const regName2 = /^[a-zA-ZéèïîÉÊÎÏ][a-zéèêàçîï]+([-'\s][a-zA-ZéèïîÉÊÎÏ][a-zéèêàçîï]+)?$/;
const regMail = /^([\w-\.\+])@((?:(\w+\.\.)))([a-zA-Z]{2,4})/i;
const regPseudo= /^(.){3,10}$/;
let error;

function isEmpty(form){
    let NbInputs = form.length-
1; // on compte tous les champs du formulaire moins le bouton
    let nbInputNonVide=0; // pour verifier que tous les champs soient contrôlés
    const optionnalInputs = ""; // si des champs son optionnels on saute la condition qui
demande a remplir le champ et on passe a la validation dans le else

    //---- CONTROLE GENERIQUE sur champs vides----//
    for(i=0; i<NbInputs; i++){
        if(form[i].value == "" && !optionnalInputs){
            controlMsg(form[i],'!! veuillez remplir ce champ !!',false);
            error = true;
        }else{
            nbInputNonVide++
            controlMsg(form[i],'',true);
        }
    }
    //si le nb de champs non vide est = au nb de champs du formulaire alors verif OK
    if(nbInputNonVide == NbInputs){
        error = false; // les champs sont remplis pas d'erreur : on peut contrôler les don
nées
    }
}

function validateRegex(input,inputValue,regex){
    // Contrôle sur le prénom
    if(!regex.test(inputValue)){
        error=true;
        controlMsg(input, "Le format n'est pas correct",false);
    }
    if(inputValue == ""){
        controlMsg(input, "Veuillez remplir ce champ",false);
        error=true;
    }
}

//fonction de gestion des erreurs et styles erreur
```

```

function controlMsg(input, message, valid){
    const formControl = input.parentElement;
    const small = formControl.querySelector('small');

    small.innerText = message;

    if(!valid){
        formControl.classList.add('error');
        formControl.classList.remove('success');
    }else{
        formControl.classList.add('success');
        formControl.classList.remove('error');
    }
};

formInscr.addEventListener('submit',function(e){
    const userFirstnameValue = userFirstname.value.trim(); //suppression des espaces.
    const userNameValue = userName.value.trim();
    const userMailValue = userMail.value.trim();
    const userPseudoValue = userPseudo.value.trim();

    isEmpty(formInscr);
    validateRegex(userFirstname,userFirstnameValue,regName);
    validateRegex(userName,userNameValue,regName);
    validateRegex(userMail,userMailValue,regMail);

    if(userPseudoValue && (userPseudoValue.length <3 || userPseudoValue.length >10)){
        controlMsg(userPseudo,'entre 3 et 10 caractères attendus',false);
        error=true;
    }

    // validation finale envoi du formulaire
    if(error){
        e.preventDefault();
        alert('Formulaire non envoyé : Merci de vérifier les données renseignées')
    }else{
        alert('Formulaire envoyé');
    }
})
})

```