# TP n°5 – programmation événementielle en Javascript Les formulaires

Récupérer le fichier tp5.txt sur la plateforme pédagogique et le renommer tp5.html

#### 1- Fonctionnement par défaut

- a) Remplir les 3 champs du formulaire et valider; que se passe-t-il? A quoi servent les attributs action et method?
- b) Ajouter un gestionnaire d'événement lors de la soumission du formulaire (événement "submit") pour ouvrir une popup qui affiche les nom, prénom et mail.

Penser à compléter l'entête :

// nom de l'événement
// cible
// action

### 2- Champs obligatoires

- a) Modifier le code Javascript pour, au moment de la soumission, vérifier que les champs nom et prenom soient remplis (champs obligatoires); si ce n'est pas le cas, empêcher la validation du formulaire grâce à la méthode preventDefault() de l'objet event (paramètre de la fonction de callback) et afficher un message d'erreur dans une popup.
- b) Ajouter directement dans le code HTML un span avec un id après chaque <input>. Modifier le code précédent pour que les messages d'erreur s'affichent dans les span correspondantes (par exemple, le message « nom obligatoire » doit s'afficher à côté de la zone de saisie du nom).

Vous pouvez aussi ajouter un style (dans le document html) pour que les messages s'affichent en rouge.

À chaque nouvelle tentative de validation, il faut penser à faire disparaître tous les messages affichés lors de la précédente tentative.

## 3- Type des champs

a) Dans le code HTML, ajouter une zone de saisie pour l'âge; puis ajouter le code javascript pour vérifier que la valeur saisie est bien un nombre: utiliser pour cela la fonction isNan(valeurATester) qui retourne true si le paramètre valeurATester n'est pas un nombre. NB: le champ n'est pas obligatoire.

Procéder comme à l'étape 2.b pour afficher un message d'erreur.

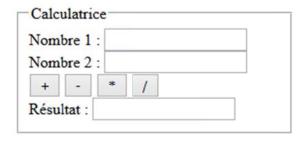
b) vérifier également que le mail comporte un arobase ; sinon afficher un message d'erreur : utiliser pour cela la méthode indexOf(chaineRecherchée) de la classe String.

NB : le champ n'est pas obligatoire.

Procéder comme à l'étape 2.b pour afficher un message d'erreur.

NB: on pourrait améliorer en vérifiant qu'il n'y ait qu'un et un seul arobase.

#### 4- Calculatrice



```
<!DOCTYPE html>
<html lang="fr">
 <head>
      <title>La calculatrice</title>
      <meta charset="utf-8" />
 </head>
 <body>
        <form>
             <fieldset>
                    <legend>Calculatrice</legend>
                    <label>Nombre 1 :</label>
                            <input type="text" id="num1" /><br/>
                    <label>Nombre 2 :</label>
                            <input type="text" id="num2" /><br/>
                    <input type="button" value="+" id="+" />
                    <input type="button" value="-" id="-" />
                    <input type="button" value="*" id="*" />
                    <input type="button" value="/" id="/" /><br/>
                    <label>Résultat :</label>
                            <input type="text" id="res" /><br/>
             </fieldset>
      </form>
      <script>
      </script>
  </body>
</html>
```

Écrire le code javascript permettant de faire fonctionner cette calculatrice : l'utilisateur tape le nombre 1, puis le nombre 2. Quand il appuie sur l'opération (+, -, \* ou /), le résultat s'affiche. NB : on ne doit pas pouvoir diviser par 0.

Version améliorée : vérifier que les données entrées soient des nombres.