



PREMIERE GENERALE

NUMERIQUE ET SCIENCES INFORMATIQUES

FORMULAIRE SUR UNE PAGE WEB

- ▷ Histoire de l'informatique
- ▷ Représentation des données : types et valeurs de base
- ▷ Représentation des données : types construits
- ▷ Traitement de données en tables
- ▷ **Interactions entre l'homme et la machine sur le Web**
- ▷ Architectures matérielles et systèmes d'exploitation
- ▷ Langages et programmation
- ▷ Algorithmique

1 – MISE EN SITUATION

1.1 – Formulaires et JavaScript

Un formulaire est un des principaux moyens **d'interaction entre un utilisateur et un site WEB**. Avec un formulaire HTML, les données recueillies sont envoyées à un serveur qui va ensuite les traiter. Ces formulaires permettent d'échanger avec les utilisateurs ou clients (formulaire de contact), de poster un message sur un forum, de recueillir des informations telles que l'adresse de livraison ou le numéro de carte bancaire lors d'un achat sur internet,...

Les informations peuvent être saisies sous différentes formes : zone de texte, cases à cocher, listes déroulantes, etc... Les données saisies dans un formulaires sont **envoyées via le réseau à un serveur WEB** qui va les traiter et renvoyer une réponse adaptée au navigateur client sous la forme d'une nouvelle page web. Le serveur Web emploie une technologie adaptée, comme par exemple le **langage PHP**.

JavaScript, va nous permettre de **manipuler le formulaire et ses données** directement côté navigateur client, avant d'envoyer ces données vers un serveur externe. Il sera possible, ainsi, de proposer à l'utilisateur, une liste de suggestions au fur et à mesure de sa frappe, ou l'avertir en cas de saisie erronée, ou bien lui proposer une liste de suggestions par exemples.

1.2 – Création de formulaires

Voici ci-contre, un exemple de formulaire :

Formulaire d'inscription

Pseudo :

Mot de passe :

Courriel :

☐ M'envoyer un courriel de confirmation

☐ M'abonner à la newsletter et aux promotions

☐ M'abonner uniquement à la newsletter

☒ Ne pas m'abonner

Nationalité :

Les éléments de saisie des données du formulaires sont contenus entre les balises `<form>` et `</form>`. La balise HTML `<form>` représente une section d'un document qui contient des contrôles interactifs permettant à un utilisateur d'envoyer des données à un serveur web.

Cette balise supporte un certain nombre d'attribut parmi lesquels :

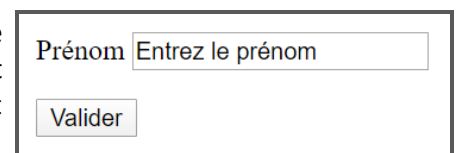
- **method** : définit la **méthode de requête HTTP** qui sera utilisée pour envoyer les données au serveur. C'est un attribut énuméré qui peut prendre les valeurs suivantes : GET et POST.
- **name** : attribut optionnel qui définit le **nom du formulaire**. Il doit être unique parmi tous les formulaires d'un document
- **action** : **définit l'URL** du programme qui traitera les informations soumises par le formulaire.

Notre formulaire sera géré uniquement côté client avec JavaScript, les valeurs des attributs **method** et **action** n'auront pas d'importance dans la suite de l'activité.

2 – PRINCIPAUX ELEMENTS D'UN FORMULAIRE

2.1 – Zones de texte

En HTML, on insère une zone de texte avec la balise `<input>` dans laquelle l'attribut **type** a pour valeur **text**. Cette balise possède également l'attribut **name** qui est le nom de la zone de texte qui va produire la variable et l'attribut facultatif **value** qui définit le contenu au départ de la zone.



La balise `<label>` permet de définir une légende pour un objet d'une interface utilisateur.

Pour valider l'envoi du formulaire, il est nécessaire de créer un bouton de validation à partir de la balise `<input>` dans laquelle l'attribut **type** a pour valeur **submit**.

Exercice 1

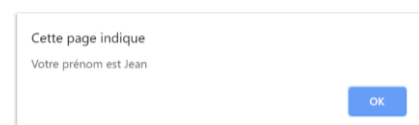
1. **Editer** le code HTML précédent. **Enregistrer** le fichier sous le nom **exercice1.html**.

exercice1.html :

```
<html>
<head>
  <script src="scriptEx1.js"></script>
</head>
<body>
  <form method="get" action="">
    <p> <label for="prenom">Prénom</label>
      <input type="text" name="prenom" id="prenom" value="Entrez le prénom"> </p>
    <p> <input type="submit" value="Valider" onclick="afficheEntree();"> </p>
  </form>
</body>
</html>
```

2. **Créer** un nouveau document nommé **scriptEx1.js** et **compléter** la fonction **afficheEntree()** qui récupère et affiche le prénom qui sera saisi par l'utilisateur. **Vérifier** le fonctionnement obtenu.

```
function afficheEntree() {
  var prenom = document.querySelector('#prenom').value;
  .....
}
```





Il est possible de modifier la page Web afin d'ajouter une **aide lors de la saisie de donnée** dans une zone de texte. Lorsque qu'un champ de saisie est activé avec la souris, ou le clavier il reçoit le focus. En détectant cet événement, il est possible de rajouter un texte d'aide à la saisie.

Lorsqu'une zone de saisie est sélectionnée, elle reçoit le focus. Il est possible de détecter le focus grâce à l'événement **onfocus**. Il est également possible de détecter lorsque la zone n'a plus le focus grâce à l'événement **onblur**.

Exercice 2

On veut afficher un texte d'aide sur la page HTML lorsque l'utilisateur active le champ de saisie du texte. Pour cela il faut créer une fonction **afficheAide()** qui sera exécutée lorsque le champ de saisie recevra le focus. Le message affiché est le suivant : « Veuillez entrer prénom avec un nombre maximal de 20 caractères ». Ce texte sera affiché entre les balises **** et **** du fichier HTML.

Prénom Veuillez entrer votre prénom en majuscules

1. **Modifier** le code HTML précédent avec le code ci-dessous. **Enregistrer** le fichier sous le nom **exercice2.html**.

exercice2.html :

```
<html>
<head>
  <script src="scriptEx2.js"></script>
</head>
<body>
  <form method="get" action="">
    <p> <label for="prenom">Prénom</label>
      <input type="text" name="prenom" id="prenom" onfocus="afficheAide();">
      <span id="aidePrenom"></span> </p>
    <p> <input type="submit" value="Valider" onclick="afficheEntree();"></p>
  </form>
</body>
</html>
```

2. **Modifier** le code JavaScript précédent afin de répondre au cahier des charges. **Enregistrer** le fichier sous le nom **scriptEx2.js**.
3. **Vérifier** le fonctionnement obtenu.
4. **Modifier** le code JavaScript précédent afin que le message d'aide ne soit plus affiché lorsque la zone de saisie n'a plus le focus. **Enregistrer** le fichier. **Vérifier** le fonctionnement obtenu.

Il est possible de rajouter des attributs à cette zone de texte afin de valider la saisie :

- L'attribut **minlength** permet de définir le nombre minimal de caractères qui peuvent être saisis.

```
<input type="text" name="donnee" id="donnee" minlength="5">
```

Prénom Jq

 Veuillez respecter le format requis.

- L'attribut **maxlength** permet de définir le nombre maximal de caractères qui peuvent être saisis.

```
<input type="text" name="donnee" id="donnee" maxlength="10">
```



- L'attribut **required** est un attribut booléen qui indique que le champ doit être renseigné avant de pouvoir envoyer le formulaire.

```
<input type="text" name="donnee" id="donnee" required>
```

- L'attribut **required** associé à l'attribut **pattern** permet d'afficher un message d'erreur si le format des données saisies n'est pas correct (Pour une description plus précise de l'attribut **pattern** : Site « developer.mozilla.org »). Par exemple pour imposer une saisie en majuscules :

```
<input type="text" name="donnee" id="donnee" required  
pattern="[A-Z]">
```

Exercice 3

1. **Modifier** le code HTML précédent afin que le prénom soit constitué de 2 à 15 caractères et autoriser uniquement les lettres majuscules et minuscules. **Enregistrer** le fichier sous le nom **exercice3.html**.
2. **Vérifier** le fonctionnement obtenu.

Lorsque la zone de texte recevoir un **mot de passe**, il est possible de cacher le texte entré par le visiteur en utilisant **type="password"** au lieu de **type="text"**. Le texte saisi est remplacé par des astérisques * ou des points •.

Exercice 4

1. **Modifier** le code HTML précédent afin que le texte saisi ne soit directement pas visible. **Enregistrer** le fichier sous le nom **exercice4.html**.
2. **Vérifier** le fonctionnement obtenu.

2.2 – Zones de texte multilignes

Une zone de texte multilignes est une grande zone dans laquelle, le visiteur peut écrire un **message avec autant de lignes qu'il veut**. La création de cette zone est réalisée avec l'aide de la balise **<textarea>**.

Les attributs **rows** et **cols** permettent de définir la taille (nombre de lignes et de colonnes) de la zone visible. Le contenu facultatif au départ de la zone de texte est indiqué entre les balises **<textarea>** et **</textarea>**.

Exercice 5

1. **Editer** les codes ci-dessous et **enregistrer** les fichiers.

exercice5.html :

```
<html>  
  <head>  
    <script src="scriptEx5.js"></script>  
  </head>  
  <body>  
    <p> Veuillez poster un commentaire </p>  
    <form method="get" action="">  
      <p> <textarea name="message" id="zone1" rows="8" cols="40">  
        Tapez votre commentaire ici... </textarea> <br/> </p>  
      <p> <input type="submit" value="Valider" onclick="afficheEntree();" </p>  
    </form>  
  </body>  
</html>
```

Exercice 5**scriptEx5.js :**

```
function afficheEntree() {  
    const texte = document.querySelector('#zone1').value ;  
    alert('Message : ' + texte) ;  
}
```

2. Vérifier le fonctionnement obtenu.**2.3 – Liste déroulante**

La création d'une liste déroulante est réalisée avec l'aide de la balise `<select>`.

L'attribut `selected` (attribut booléen qui ne possède pas de valeur) de la balise `<option>` permet de définir le choix de la liste affiché par défaut. Si aucune des options ne possède cet attribut, le premier choix de la liste est affiché par défaut.

Exercice 6**1. Editer les codes ci-dessous et enregistrer les fichiers.****exercice6.html :**

```
<html>  
  <head>  
    <meta charset="utf-8">  
    <script src="scriptEx6.js"></script>  
  </head>  
  <body>  
    <p> Vous habitez dans quel pays ? </p>  
    <form method="GET" action="">  
      <p> <select name="pays" id="liste1">  
        <option value="Allemagne"> Allemagne </option>  
        <option value="Canada"> Canada </option>  
        <option value="Espagne"> Espagne </option>  
        <option value="Etats-Unis"> Etats-Unis </option>  
        <option value="France"> France </option>  
        <option value="Japon"> Japon </option>  
        <option value="Royaume-Uni"> Royaume-Uni </option>  
      </select><br/> </p>  
      <p> <input type="submit" value="Valider" onclick="afficheEntree();" > </p>  
    </form>  
  </body>  
</html>
```

scriptEx6.js :

```
function afficheEntree() {  
    const texte = document.querySelector('#liste1').value;  
    alert('Pays de résidence : ' + texte);  
}
```

2. Vérifier le fonctionnement obtenu.**3. Modifier le code HTML afin que le choix « France » soit affiché par défaut Vérifier le fonctionnement obtenu.**

2.4 – Cases à cocher

La création d'une liste déroulante est réalisée avec l'aide de la balise `<select>`.

Quelle langue étudiez-vous ?

☐ Allemand ☐ Anglais ☐ Espagnol ☐ Italien

Exercice 7

1. **Editer** les codes ci-dessous et **enregistrer** les fichiers.

exercice7.html :

```
<html>
<head>
  <meta charset="utf-8">
  <script src="scriptEx7.js"></script>
</head>
<body>
  <form method="get" action="">
    <p> Quelle langue étudiez-vous ? </p>
    <p> <input type="checkbox" name="langue" id="1" value="Allemand">
      <label for="1">Allemand</label>
      <input type="checkbox" name="langue" id="2" value="Anglais">
      <label for="2">Anglais</label>
      <input type="checkbox" name="langue" id="3" value="Espagnol">
      <label for="3">Espagnol</label>
      <input type="checkbox" name="langue" id="4" value="Italien">
      <label for="4">Italien</label> </p>
    <p> <input type="submit" value="Valider" onclick="afficheEntree();" </p>
  </form>
</body>
</html>
```

scriptEx7.js :

```
function afficheEntree() {
  if (document.getElementsByName("langue")[0].checked == true) {
    alert("Vous étudiez l'allemand");
  }
  if (document.getElementsByName("langue")[1].checked == true) {
    alert("Vous étudiez l'anglais");
  }
  if (document.getElementsByName("langue")[2].checked == true) {
    alert("Vous étudiez l'espagnol");
  }
  if (document.getElementsByName("langue")[3].checked == true) {
    alert("Vous étudiez l'italien") ;
  }
}
```

2. **Vérifier** le fonctionnement obtenu.

La balise `label` est facultative. Elle associe le libellé à la case à cocher qui a le même « id » que son attribut « for » et ainsi permettre de sélectionner la case également avec un clic sur le libellé.

L'attribut `checked` (attribut booléen qui ne possède pas de valeur) de la balise `<input>` permet de cocher une case par défaut.

**Exercice 8**

1. **Modifier** le code HTML précédent afin que la case « Espagnol » soit cochée par défaut. **Enregistrer** le fichier sous le nom **exercice8.html**.
2. **Vérifier** le fonctionnement obtenu.

2.4 – Boutons d'options

Les boutons d'options fonctionnent par groupes de deux au minimum. Tous les boutons du groupe d'avoir le même nom afin de désactiver les boutons non cochés quand un des boutons est activé.

De quel sexe êtes vous ?

☐ Féminin ☐ Masculin

Exercice 9

1. **Editer** le code HTML ci-dessous. **Enregistrer** le fichier sous le nom **exercice9.html**.

exercice9.html :

```
<html>
  <head>
    <meta charset="utf-8">
    <script src="scriptEx9.js"></script>
  </head>
  <body>
    <form method="GET" action="">
      <p> De quel sexe êtes vous ? </p>
      <p> <input type="radio" name="sexe" id="1" value="F">
        <label for="1">Féminin</label>
        <input type="radio" name="sexe" id="2" value="M">
        <label for="2">Masculin</label> </p>
      <p><input type="submit" value="Valider" onclick="afficheEntree();"></p>
    </form>
  </body>
</html>
```

2. **Créer** un nouveau document nommé **scriptEx9.js** et **éditer** la fonction **afficheEntree()** qui affiche, après l'appui sur le bouton « Valider » :
 - « Bonjour Madame » si bouton « Féminin » est coché
 - « Bonjour Monsieur » si bouton « Masculin » est coché
3. **Vérifier** le fonctionnement obtenu.

2.5 – Zone d'entrée d'adresse mail

La balise **<input>** dont l'attribut **type** a pour valeur **email** permettent à un utilisateur de saisir et d'éditer une adresse mail.

Veuillez entrer votre adresse email

L'attribut **size** permet fixer le nombre de caractères affichés par le champ. L'attribut **placeholder** qui permet d'afficher une adresse d'exemple lorsqu'aucune valeur n'est saisie.

**Exercice 10**

1. **Editer** les codes ci-dessous et **enregistrer** les fichiers sous les noms **exercice10.html** et **scriptEx10.js**.

exercice10.html :

```
<html>
<head>
  <meta charset="utf-8">
  <script src="scriptEx10.js"></script>
</head>
<body>
  <form method="get" action="">
    <p> <label for="mail">Veuillez votre adresse email</label>
      <input type="email" name="mail" id="mail">
    <p> <input type="submit" value="Valider" onclick="afficheEntree();"> </p>
  </form>
</body>
</html>
```

scriptEx10.js :

```
function afficheEntree() {
  const texte = document.querySelector('#mail').value;
  alert('Votre adresse mail est ' + texte); }
```

2. **Vérifier** le fonctionnement obtenu.
3. **Indiquer** ce qui se passe si on clique sur le bouton valider alors que l'adresse email ne contient pas le symbole « @ » ou bien si rien n'est saisi après le symbole « @ ».
4. **Modifier** le code HTML afin de fixer d'imposer la saisie de l'adresse email, de fixer le nombre de caractères de la zone de saisies à 30 et d'afficher dans la zone de saisie le message suivant « prenom.nom@etu.u-bordeaux.fr » lorsqu'aucune adresse n'a été saisie.
5. **Vérifier** le fonctionnement obtenu.

3 – FORMULAIRE D'INSCRIPTION

Le but de cette activité est de réaliser le formulaire d'inscription suivant :

Formulaire d'inscription

Pseudo :

Mot de passe :

Courriel :

☐ M'envoyer un courriel de confirmation

☐ M'abonner à la newsletter et aux promotions

☐ M'abonner uniquement à la newsletter

☐ Ne pas m'abonner

Nationalité :



Le cahier des charges est le suivant :

- Il est obligatoire de remplir toutes les zones de saisie hormis la case à cocher « M'envoyer un email de confirmation ».
- **Champ de saisie « Pseudo »** : la valeur doit avoir une taille comprise entre 4 et 8 caractères. Lorsque le focus est sur la zone de saisie, le message d'aide suivant doit être affiché :

Pseudo : Saisir un pseudo. Il doit être compris entre 4 et 8 caractères

- **Champ de saisie « mot de passe »** : la valeur doit être cachée et sa taille doit être de 8 caractères au minimum. Lorsque le focus est sur la zone de saisie « mot de passe », le message d'aide suivant doit être affiché :

Mot de passe : Saisir un mot de passe. Il doit être constitué d'au moins 8 caractères

- **Champ de saisie « Courriel »** : La taille de la zone de saisie doit être de 30 caractères. Lorsque le focus est sur cette zone de saisie, le message d'aide suivant doit être affiché :

Courriel : Saisir une adresse mail valide

- **Boutons d'options « Abonnement »** : Le bouton « Ne pas m'abonner » doit être sélectionné par défaut.
- **Liste déroulante « Nationalité »** : Cette liste est constituée de quatre options : Française, Belge, Suisse et Autre. La nationalité Française doit être sélectionnée par défaut.
- **Bouton « Annuler »** : Ce bouton doit permettre l'affichage suivant :

Cette page indique
Inscription annulée

OK

- **Bouton « Envoyer »** : Ce bouton doit permettre l'affichage suivant :

Formulaire d'inscription

Pseudo :

Mot de passe :

Courriel :

- ☒ M'envoyer un courriel de confirmation
- ☐ M'abonner à la newsletter et aux promotions
- ☒ M'abonner uniquement à la newsletter
- ☐ Ne pas m'abonner

Nationalité :

Cette page indique

Pseudo : Jojo33
Mot de passe : mdpmdp01
Adresse mail : jojo33@hotmail.com
Email de confirmation : Oui
Abonnement : Abonnement newsletters
Nationalité : FR

OK

Pour l'affichage il faut créer un message global qui va contenir tout le message à afficher. Pour sauter à la ligne il faut placer les caractères « \n ». Ce message étant variable il est nécessaire de le déclarer comme une variable par l'intermédiaire de l'instruction `var`.



Par exemple :

```
const texte_prenom = "Jean"
const texte_nom = "Dupont"
var message = 'Prénom : ' + texte_prenom + '\nNom : ' + texte_nom ;
alert(message)
```

Cette page indique

Prénom : Jean

Nom : Dupont

OK

Exercice 11

1. **Créer** les fichiers **inscription.html** et **script.js**.
2. **Editer** les codes HTML et Javascript permettant de répondre au cahier des charges.
3. **Vérifier** le fonctionnement obtenu.