

# Intégration Web

MMI 1 – TP#3 S1



Danielo **JEAN-LOUIS**  
Michele **LINARDI**

**Entrons dans le web 2.0**

# Formulaires

- Recueil des données pour les envoyer à un serveur
- Ensemble de balises contenues dans une balise `<form>`
  - **Tous les éléments d'un formulaire doivent s'y trouver dedans**

# Formulaires

- Exemple de fonctionnalités réalisables avec un formulaire :
  - Formulaire de connexion / inscription
  - Messagerie
  - [...]



The image shows a web form titled "Créer un compte". It contains four input fields: "Prénom \*" (first name), "Nom \*" (last name), "Adresse électronique \*" (email address), and "Mot de passe \*" (password). Each field is followed by a red asterisk indicating it is required. The password field has a "Montrer" (Show) button next to it. At the bottom, there is a small blue information icon followed by the text "Les mots de passe doivent comporter au moins 8 caractères".

Cette fonctionnalité est impossible sans un formulaire

```
<form action="traitement.php" method="get">  
  <!-- Contenu du formulaire -->  
</form>
```

**Exemple de squelette de formulaire**

# Balise <form> - Attribut "method"

- Permet de définir la façon dont les données sont envoyées
- Possède la valeur "GET" par défaut
  - Valeur d'attribut insensible à la casse

Get = **GET = get** = gEt

Préférez les casses "GET" et "get" pour l'attribut "method" de la balise <form>

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>

# Balise <form> - Attribut "method"

- Trois valeurs possibles :
  - GET – **Valeur par défaut**
  - POST
  - DIALOG

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>



# Attribut "method" – Valeur "GET"

- Données passent dans l'url / barre d'adresse du navigateur
  - **Données visibles à l'envoi**
- Limitation à 255 caractères historiquement
  - ~2 000 caractères de nos jours

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

# Attribut "method" – Valeur "GET"

- **Ne permet pas d'envoyer des médias**
- Valeur par défaut de l'attribut "method"
- Données envoyées vers le serveur à la soumission du formulaire
- Permet de rejouer la requête au rafraîchissement de la page

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>
- <https://developer.mozilla.org/fr/docs/Glossary/Idempotent>

# Attribut "method" – Valeur "GET"

- Exemples d'utilisation :
  - Test sur ENT
  - Vidéo Youtube
- Prend la forme :
  - mon.url?**val1=toto&val2=3**

Caractère signifiant le début des paramètres d'URL (ajouté par le navigateur à la soumission)

Valeurs de votre formulaire séparée par "&" (ajouté par le navigateur)

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

# Attribut "method" – Valeur "GET"

Caractère signifiant le début des paramètres d'URL. **Indispensable**

`https://cours-prod.cyu.fr/question/edit.php?cmid=205308&cat=42238%2C754843`

Paramètre

Séparateur entre chaque  
paire paramètre/valeur

Valeur

Combien de paramètres avons-nous envoyés avec le formulaire ?

# Attribut "method" – Valeur "POST"

- Données passent dans le corps de la requête
  - **Données invisibles à l'envoi\***
  - **Permet d'envoyer des données sensibles**
  - **Ne garde pas les données envoyées en mémoire**
- Pas de limite de caractères
- Permet d'envoyer des médias (images, vidéos...)
  - Il faudra rajouter un attribut spécial

\* : A condition que le serveur soit https

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

# Attribut "method" – Valeur "POST"

- Données envoyées vers le serveur à la soumission du formulaire
- Exemples d'utilisation :
  - Formulaire d'achat en ligne
  - Upload d'images / vidéos

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

# "POST" ou "GET" ?

- Posez-vous les questions suivantes :
  - Est-ce que les données sont sensibles ?
  - Est-ce que l'utilisateur n'a pas besoin de faire un marque-page ?
  - Est-ce que je dois envoyer des médias ?

**Si une seule des questions a pour réponse "Oui" alors utilisez "POST"**

## Source(s) :

- <https://www.guru99.com/difference-get-post-http.html> – anglais
- [https://www.w3schools.com/tags/ref\\_httpmethods.asp](https://www.w3schools.com/tags/ref_httpmethods.asp) – anglais
- <https://developer.mozilla.org/fr/docs/Glossary/Idempotent>

# "POST" vs "GET" - Récapitulatif

GET	POST
Envoie les données dans l'URL (Les données sont visibles de tous)	Permet d'envoyer des données sensibles
Permet de jouer la requête lorsque la page est rafraîchie	Ne permet pas de rejouer la requête au rafraîchissement
Limité en taille pour la requête	Pas de limite pour la requête
Ne permet pas d'envoyer des médias	Permet d'envoyer des médias
Peut être mis en marque-page	Ne peut pas être mis en marque-page



**N'utilisez jamais "POST" dans un formulaire qui demande un numéro de carte bancaire. Jamais.**

# Attribut "method" – Valeur "DIALOG"

- Cache le formulaire lors de la soumission
  - Uniquement si le formulaire est contenu dans une balise <dialog>
- Données ne sont pas envoyées vers le serveur à la soumission du formulaire
  - **Nécessite d'envoyer soi-même les données vers le serveur**

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

# Attribut "method" – Valeur "DIALOG"

- Affiche le formulaire dans une modale
  - Le formulaire s'affiche au-dessus de la page
- On ne l'utilisera pas en cours

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>
- <https://developer.mozilla.org/fr/docs/Learn/Forms>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form#attr-method>

# Balise <form> - Traitement des données

- L'attribut "action" permet de définir où les données vont être envoyées
- Données traitées sous forme de pairs clés / valeurs
  - Clé définie par l'attribut "name"
    - **Tous les éléments de formulaire en ont besoin**
  - Valeur définie par l'attribut "value" (sauf pour la balise <textarea></textarea>)

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>

# Balise <form>

Endroit où les données vont  
être envoyées et traitées

```
<form action="traitement.php" method="get">  
  <!-- Contenu du formulaire -->  
</form>
```

Comment les données  
vont être envoyées

# Éléments de formulaire

- Ensemble de balises permettant à l'utilisateur de remplir les formulaires
- Balise la plus courante : `<input>`
  - Très polyvalente
- Fonctionnent de pair avec la balise `<label>`
- **Doivent être dans une balise `<form>`**

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form>

# Balise <input />

- Balise auto-fermante
- **N'accepte que du texte mono-ligne**
  - Ne permet pas de faire un retour à la ligne
- Attribut "value" permet de définir une valeur par défaut
  - Valeur souvent en provenance du serveur

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input/text>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributes/autocomplete>

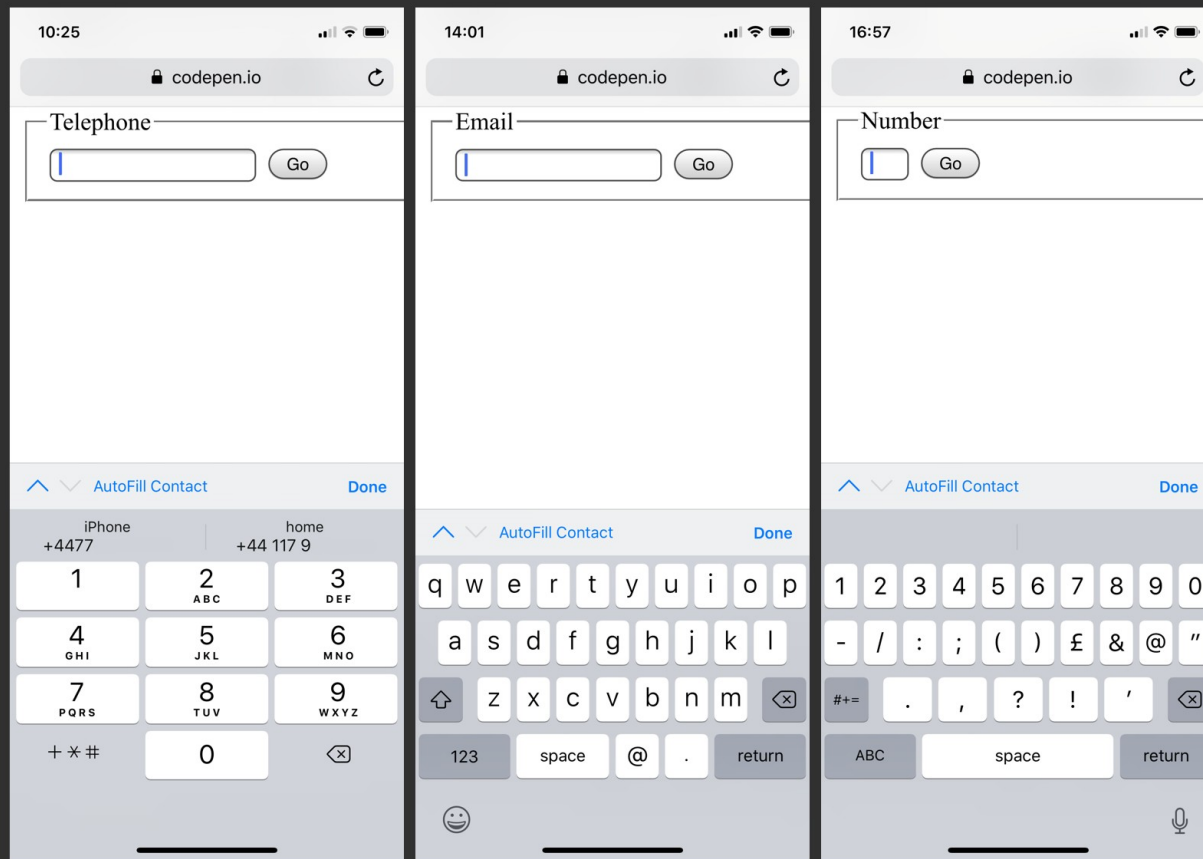
# Balise <input />

- Attribut "type" pour définir le type de données qui peut être entré ou le comportement du champ
  - "text" : texte
  - "email" : courriel (nouveau html5)
  - "tel" : téléphone (nouveau html5)
  - "hidden" : champ caché
  - [...]

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input/text>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input>





Sur mobile (ici iOS) le clavier change en fonction du champ dépendamment de valeur de son attribut "type"

Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input/text>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input>
- <https://www.smashingmagazine.com/2019/01/html5-input-types/?ref=heydesigner> - anglais

# Balise <input />

- Exemple d'utilisation :
  - Champ pour un pseudonyme
  - Champ pour un numéro de téléphone

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input/text>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributs/autocomplete>

# Balise <input />

```
<input  
  type="text"  
  name="prenom"  
  id="firstname"  
>
```

Exemple de balise <input>

**Attribut type.** Permet de définir le type donnée qui peuvent être entrées

**Attribut name.** Permet de définir une variable dont la valeur sera la valeur de l'input

**Attribut id.** Valeur unique sur la page. Permet de lier le champ à une balise <label>

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input/text>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributes/autocomplete>

# Attribut "name"

- Définit une variable côté serveur. Exemple PHP :

```
<input  
    type="text"  
    name="prenom"  
    id="firstname"  
>
```

Exemple de balise <input>



```
<?php  
    // Voilà la valeur de mon champ  
    // avec la valeur "prenom"  
    // pour l'attribut "name" dans le HTML  
    echo $_GET["prenom"];  
?>
```

Avec la valeur de ma page PHP pour l'attribut "action" et l'attribut name, je peux récupérer la valeur dans le code PHP

# Attribut "value"

- Définit la valeur par défaut d'un champ input :

```
<input  
  type="text"  
  name="prenom"  
  id="firstname"  
  value="Je suis là par défaut"  
>
```

Exemple de balise <input>  
avec une valeur par défaut

```
<input  
  type="text"  
  name="prenom"  
  id="firstname"  
  value="<?php echo $_GET['prenom'];?>"  
>
```

Même exemple, mais cette fois-ci, la valeur de  
l'input provient du PHP

# Balise <label></label>

- Décrit le rôle d'un champ pour un être humain
  - **Ne doit en aucun cas être substitué par l'attribut "placeholder" de la balise <input />**

## Source(s) :

- <https://www.anysurfer.be/fr/blog/detail/lelement-placeholder-nest-pas-un-label>
- [https://www.w3schools.com/howto/howto\\_css\\_form\\_icon.asp](https://www.w3schools.com/howto/howto_css_form_icon.asp) - mauvaise pratique

# Point accessibilité – l'attribut placeholder

- Doit uniquement servir à afficher un exemple de données attendues dans un champ
- **Ne doit pas remplacer la balise <label>**

## Source(s) :

- <https://www.anysurfer.be/fr/blog/detail/lelement-placeholder-nest-pas-un-label>
- [https://www.w3schools.com/howto/howto\\_css\\_form\\_icon.asp](https://www.w3schools.com/howto/howto_css_form_icon.asp) - mauvaise pratique

# Point accessibilité – l'attribut placeholder

pseudonyme



Ici, on utilise l'attribut "placeholder" comme `<label>` lorsque le champ est rempli, on ne sait plus sa signification

pseudonyme



Avec la balise `<label>` la valeur attendue dans le champ reste compréhensible même rempli et on peut utiliser l'attribut "placeholder" pour afficher un indice

## Source(s) :

- <https://www.anysurfer.be/fr/blog/detail/lelement-placeholder-nest-pas-un-label>
- [https://www.w3schools.com/howto/howto\\_css\\_form\\_icon.asp](https://www.w3schools.com/howto/howto_css_form_icon.asp) - mauvaise pratique
- <http://www.ohmyband.fr/CANDIDAT.html?anchor=form%20candidat> - mauvaise pratique



# Balise <label></label>

- **Attribut "for" indispensable pour l'accessibilité**
  - Associe un label à un champ  
Le champ associé doit avoir l'attribut "id" avec la même valeur
  - Permet au clic de mettre le focus sur le champ associé

## Source(s) :

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/label>
- [https://www.w3schools.com/howto/howto\\_css\\_form\\_icon.asp](https://www.w3schools.com/howto/howto_css_form_icon.asp) - mauvaise pratique

# Bonnes pratiques – Champ de formulaire

- Un champ (input) doit impérativement avoir :
  - Un label qui lui est associé (avec l'attribut "for")
  - Les attributs suivants
    - Un attribut "id" avec une valeur identique à celle de son label associé
    - Un attribut "name" avec une valeur unique
    - Un attribut "type", de préférence en accord avec la donnée attendue

# Balise <label></label>

```
<label for="prenom">Prénom</label>  
<input type="text" name="name" id="prenom" />
```

Associe le **label** à un **champ (input)** possédant la même valeur pour l'attribut **"id"**

Associe le **champ (input)** à un **label** possédant la même valeur pour l'attribut **"for"**

## Source(s) :

- <https://www.anysurfer.be/fr/blog/detail/lelement-placeholder-nest-pas-un-label>
- [https://www.w3schools.com/howto/howto\\_css\\_form\\_icon.asp](https://www.w3schools.com/howto/howto_css_form_icon.asp) - mauvaise pratique

# Pratiquons ! - Notre premier formulaire (Partie 1)

Pré-requis :

- Avoir la ressource **ressources/formulaire-basique**

A télécharger ici :

<https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-2/ressources/formulaire-basique>

# Soumission du formulaire

- Deux choix possibles :
  - Balise `<button>` avec la valeur "submit" pour l'attribut "type"
  - Balise fermante : Permet donc de mettre du texte + image dans le bouton, par exemple

```
<button type="submit">  
  Envoyer !  
</button>
```

Exemple de code de balise `<button>`

# Soumission du formulaire

- Balise <input> avec la valeur "submit" pour l'attribut "type" et l'attribut "value" pour le libellé

```
<input type="submit" value="Envoyer !">
```

Exemple de code de balise <input>

# Soumission du formulaire

- Permet d'envoyer les données vers un serveur
  - Attribut "action" de la balise <form>
- Se place dans le formulaire, à la fin de préférence

## Soumission du formulaire

- POST ou GET : La page sera rechargée à la soumission



# Pratiquons ! - Notre premier formulaire (Partie 2)

Pré-requis :

- Avoir la ressource **ressources/formulaire-basique**

A télécharger ici :

<https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-2/ressources/formulaire-basique>

# Balise `<textarea></textarea>`

- Balise acceptant du texte multi-lignes
  - Permet à l'utilisateur de faire des retours à la ligne
- La valeur par défaut n'est pas dans l'attribut "value" mais entre les deux balises

## Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

# Balise <textarea></textarea>

- Utilisation des propriétés "rows" et "cols" pour définir le nombre de lignes et colonnes visibles
  - Possibilité d'utiliser les attributs "width" et "height" également
- La balise, par défaut, peut se redimensionner
  - **Très mauvaise idée de désactiver cette option**

## Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

# Balise `<textarea></textarea>`

- Exemples d'utilisation :
  - Rédaction de commentaire
  - Rédaction de messages
  - Bref long texte → `<textarea>`
  - [...]

## Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

# Balise <textarea></textarea>

```
<label for="commentaire">Commentaire</label>  
<textarea name="comment" id="commentaire" cols="30" rows="10"></textarea>
```

Exemple de <textarea> avec son <label> associé

**Source :**

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

# Éléments à cocher

- Permettent d'effectuer des choix parmi des choix prédéfinis
- Balise input avec une valeur spécifique pour l'attribut "type" :
  - checkbox : Permet de sélectionner plusieurs éléments en même temps
  - radio : Permet de sélectionner qu'un seul élément

## Source :

- [https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Les\\_blocs\\_de\\_formulaires\\_natifs#%C3%89l%C3%A9ments\\_%C3%A0\\_cocher](https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Les_blocs_de_formulaires_natifs#%C3%89l%C3%A9ments_%C3%A0_cocher)

# Éléments à cocher

- Doivent être associés à un label respectif pour qu'ils soient compréhensibles pour l'être humain
- Attribut "checked" permet de pré-sélectionner une valeur (ou plus)
- Nécessitent l'attribut "name" pour savoir quel est le nom de la valeur envoyée

## Source :

- [https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Les\\_blocs\\_de\\_formulaires\\_natifs#%C3%89l%C3%A9ments\\_%C3%A0\\_cocher](https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Les_blocs_de_formulaires_natifs#%C3%89l%C3%A9ments_%C3%A0_cocher)

## Element à cocher – input radio

- Nécessite la valeur "radio" pour l'attribut "type" de la balise `<input />`
- Ne permet de sélectionner qu'un seul choix
- **Doivent être de forme circulaire**

### Source :

- [https://developer.mozilla.org/fr/docs/Learn/Forms/Basic\\_native\\_form\\_controls#bouton\\_radio](https://developer.mozilla.org/fr/docs/Learn/Forms/Basic_native_form_controls#bouton_radio)



# Element à cocher – input radio

- Ne permet pas de désélectionner un choix
- Les choix liés doivent avoir la même valeur pour l'attribut "name" et une valeur unique pour l'attribut "value"

## Source :

- [https://developer.mozilla.org/fr/docs/Learn/Forms/Basic\\_native\\_form\\_controls#bouton\\_radio](https://developer.mozilla.org/fr/docs/Learn/Forms/Basic_native_form_controls#bouton_radio)

# Element à cocher – input radio

- Peut remplacer la balise `<select>` quand les choix sont limités
- Exemples d'utilisation :
  - Choix exclusif
    - VISA ou MasterCard
    - Entrée ou dessert

## Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

# Element à cocher – input radio

## Exemple d'input type "radio"

```
<input type="radio" id="gaucher" name="habilite" value="gaucher">  
<label for="gaucher">Gaucher</label><br>  
<input type="radio" id="droitier" name="habilite" value="droitier">  
<label for="droitier">Droitier</label><br>  
<input type="radio" id="ambidextre" name="habilite" value="ambidextre">  
<label for="ambidextre">Ambidextre</label><br>
```

Valeur "radio" pour avoir  
le comportement attendu

Les trois input **ont la même valeur pour  
l'attribut "name"** conséquemment seul un  
choix parmi les trois peut être sélectionné

Source :

- [https://developer.mozilla.org/fr/docs/Learn/Forms/Basic\\_native\\_form\\_controls#bouton\\_radio](https://developer.mozilla.org/fr/docs/Learn/Forms/Basic_native_form_controls#bouton_radio)

# Element à cocher – input radio

## Exemple d'input type "radio"

```
<input type="radio" id="gaucher" name="habilite" value="gaucher">  
<label for="gaucher">Gaucher</label><br>  
<input type="radio" id="droitier" name="habilite" value="droitier">  
<label for="droitier">Droitier</label><br>  
<input type="radio" id="ambidextre" name="habilite" value="ambidextre">  
<label for="ambidextre">Ambidextre</label><br>
```

Chaque input possède une valeur unique  
pour l'attribut "value"

Source :

- [https://developer.mozilla.org/fr/docs/Learn/Forms/Basic\\_native\\_form\\_controls#bouton\\_radio](https://developer.mozilla.org/fr/docs/Learn/Forms/Basic_native_form_controls#bouton_radio)

## Element à cocher – Case à cocher

- Nécessite la valeur "checkbox" pour l'attribut "type" de la balise `<input />`
- Permet de sélectionner plusieurs choix
- Permet de désélectionner un choix
- **Doivent être de forme carrée**

### Source :

- [https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Les\\_blocs\\_de\\_formulaires\\_natifs#%C3%89l%C3%A9ments\\_%C3%A0\\_cocher](https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Les_blocs_de_formulaires_natifs#%C3%89l%C3%A9ments_%C3%A0_cocher)

# Case à cocher – Envoi des données

- Valeurs uniques

```
<input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">  
<label for="vehicle1"> I have a bike</label><br>  
<input type="checkbox" id="vehicle2" name="vehicle2" value="Car">  
<label for="vehicle2"> I have a car</label><br>
```

La valeur de l'attribut "name" est unique à chaque checkbox.

Chaque donnée est envoyée indépendamment

# Élément à cocher – input checkbox

- Exemples d'utilisation - Choix multiples
  - framadate / doodle → sélection de dates

## Source :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/input>

# Pratiquons ! - Notre premier formulaire (Partie 3)

Pré-requis :

- Avoir la ressource **ressources/formulaire-basique**

A télécharger ici :

<https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-2/ressources/formulaire-basique>



# Liste déroulante – Balise `<select>`

- Permet de sélectionner un choix parmi une liste de choix définis
- Peut se substituer à la balise `<input type="radio">` quand il y a trop de choix
- Permet de sélectionner plusieurs choix en même temps
  - A éviter car ce n'est pas naturel pour l'utilisateur

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/select>

# Liste déroulante – Balise <select>

- Ne peut contenir que des balises <option>
  - Contient (balise option) l'attribut "value"
- Possibilité de grouper les options via la balise <optgroup>

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/select>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/option>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Optgroup>

# Liste déroulante – Balise `<select>`

- Attribut "selected" pour présélectionner un élément
- Possibilité de personnalisation très limitée
  - **Ne pas faire un `<select>` custom** → Crée des problèmes pour l'accessibilité

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/select>

# Liste déroulante – Balise <select>

- Exemple d'utilisation :
  - Liste de choix longue :
    - Sélection d'un pays parmi la liste de tous les pays du monde

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/select>

# Liste déroulante – Balise <select>

```
<label for="pet-select">Choose a pet:</label>

<select name="pets" id="pet-select">
  <option value="">--Please choose an option--</option>
  <option value="dog">Dog</option>
  <option value="cat">Cat</option>
  <option value="hamster">Hamster</option>
  <option value="parrot">Parrot</option>
  <option value="spider">Spider</option>
  <option value="goldfish">Goldfish</option>
</select>
```

L'attribut "name" est défini  
au niveau de la balise  
<select>

La valeur est définie au  
niveau de la balise "option"

# Point accessibilité – Personnalisation

- **Beau n'est pas égal à fonctionnel**
- Personnaliser des éléments de formulaire pour qu'ils soient beaux est une très mauvaise idée si ça ne les rend pas clairs à l'utilisation
- Peut supprimer / endommager le côté accessible des éléments

# Pratiquons ! - Notre premier formulaire (Partie 4)

Pré-requis :

- Avoir la ressource **ressources/formulaire-basique**

A télécharger ici :

<https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-2/ressources/formulaire-basique>

# Validation des formulaires – Champs requis

- Attribut "required"
  - Empêche la validation d'un formulaire si la valeur du champ est absente

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributes/required>
- [https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Validation\\_donnees\\_formulaire](https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Validation_donnees_formulaire)



# Validation des formulaires

- Do not trust user input / Ne pas faire confiance aux entrées utilisateur
- Validation automatique mais on peut tricher... avec la console du navigateur
  - **Une validation côté serveur est indispensable**

## Source(s) :

- [https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Validation\\_donnees\\_formulaire](https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Validation_donnees_formulaire)
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Form#attr-novalidate>

# Pratiquons ! - Notre premier formulaire (Partie 5)

Pré-requis :

- Avoir la ressource **ressources/formulaire-basique**

A télécharger ici :

<https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-2/ressources/formulaire-basique>

# Console du navigateur

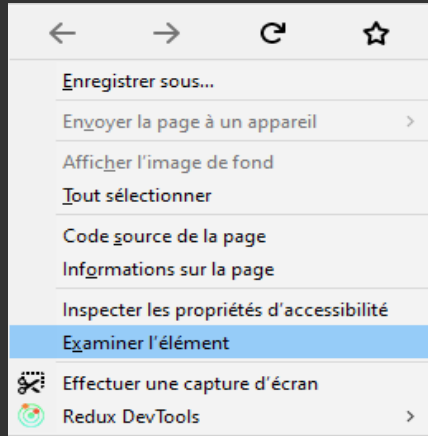
- Outil natif des navigateurs modernes
  - Même sous mobile avec un ordinateur
- Permet d'effectuer rapidement des modifications / tests sur une page web
  - ça fonctionne sur **tous** les sites web

# Console du navigateur

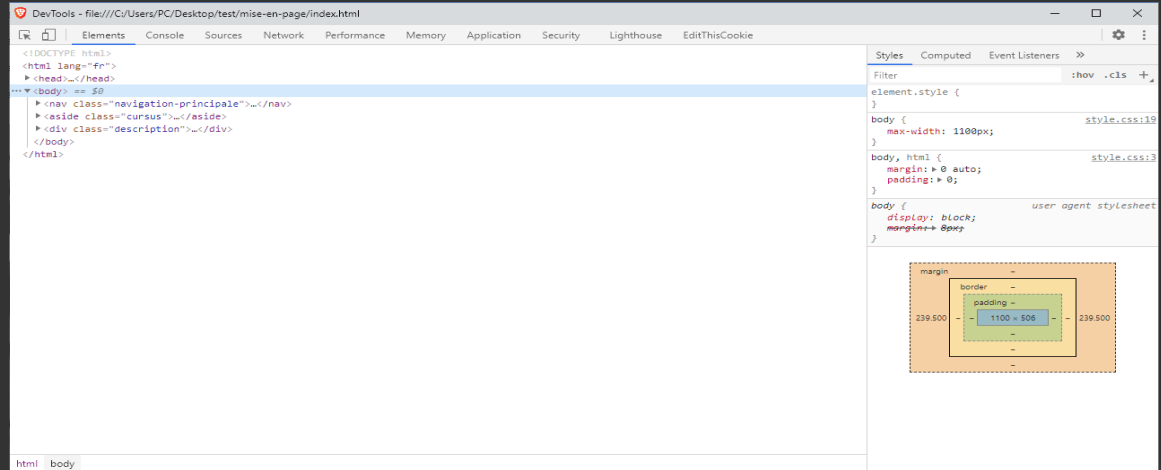
- Les changements sont locaux
- **Indispensable pour débbugger son site**
  - **Ou exploiter des failles de sécurité**
- Permet "d'émuler" des appareils mobiles

# Console du navigateur

- Accessible via la touche F12 (Win) ou clic droit > Inspecter / examiner l'élément n'importe où sur la page



Menu contextuel de  
Firefox



console du navigateur Brave

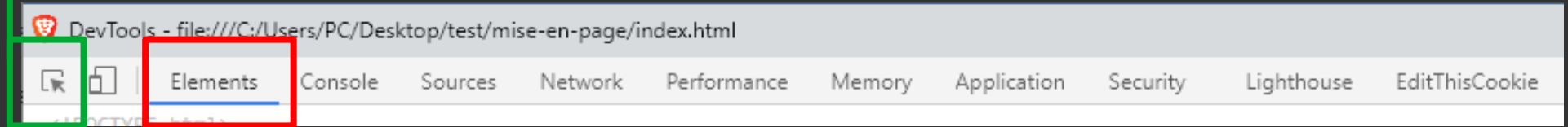
## Source(s) :

- [https://developer.mozilla.org/en-US/docs/Tools/Web\\_Console/Keyboard\\_shortcuts](https://developer.mozilla.org/en-US/docs/Tools/Web_Console/Keyboard_shortcuts)
- <https://developers.google.com/web/tools/chrome-devtools/shortcuts>

# Console du navigateur

## Outil "Sélection"

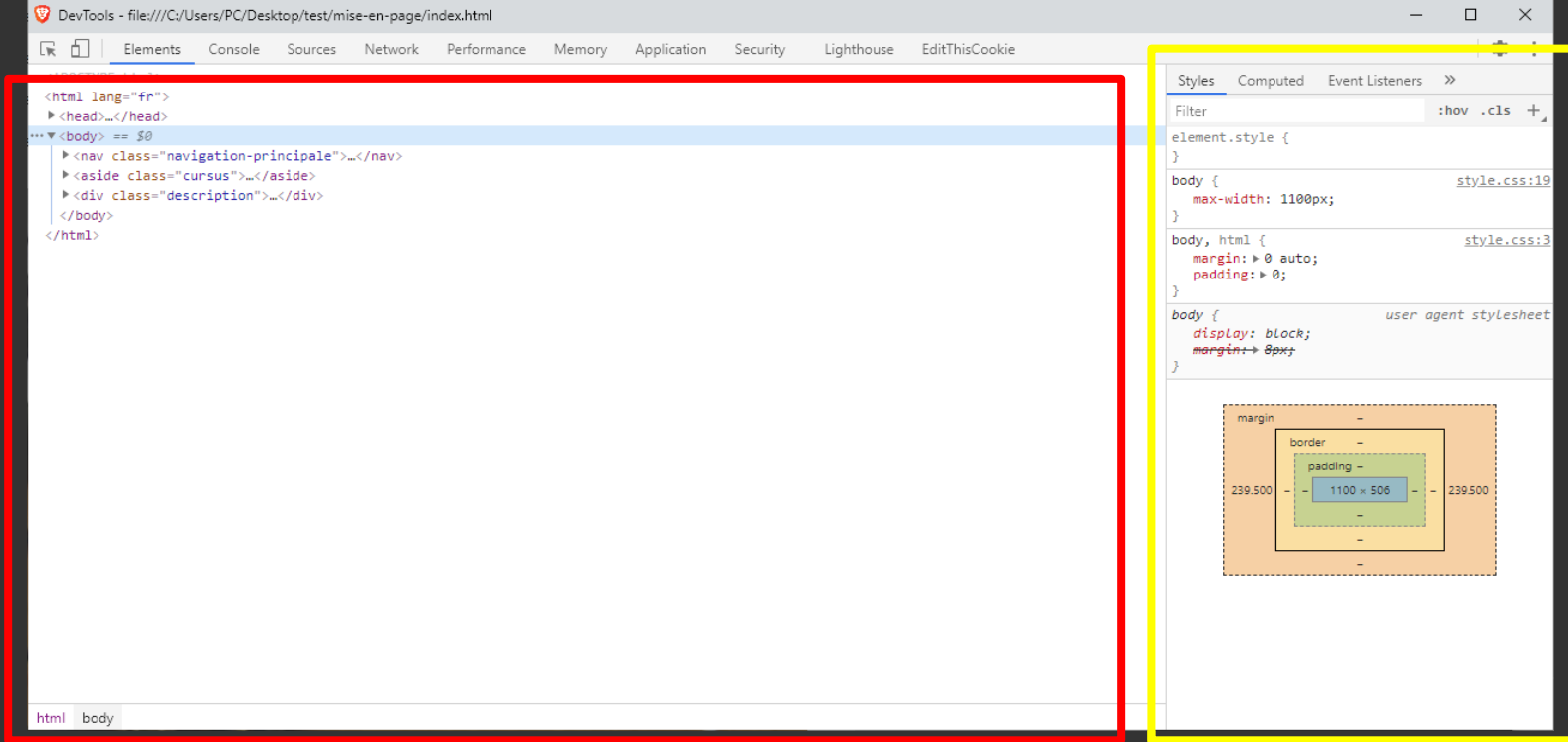
Permet de sélectionner un élément directement sur la page



## Onglet Elements (Chrome) / Inspecteur (Firefox)

Affiche la page web courante du point de vue code HTML/CSS

# Console du navigateur



Code HTML de la page courante  
L'élément en bleu est l'élément sélectionné

Déclarations CSS et  
sélecteurs CSS de l'élément  
sélectionné

# Console du navigateur

- En double cliquant sur une balise, il est possible d'en éditer
  - Le nom
  - Les Attributs
  - Le contenu (ajouter / supprimer) des balises



# Console du navigateur

- Possibilité d'éditer le CSS appliqué sur le panneau de droite
  - Ajout / retrait / modification sélecteurs CSS
  - Ajout / retrait / modification propriétés CSS
  - Modification valeur de propriétés CSS

## Pratiquons ! - Console web

- Ouvrez la console du navigateur sur le site que vous voulez
  - F12 ou clic droit > Examiner / Inspecter
  - Sélectionnez l'onglet Éléments / Inspecteur
  - Sélectionnez des balises
    - Éditez le code HTML
    - Éditez le code CSS (panneau droit)

# Console du navigateur - Fonctionnalités\*

- Modifier le code de la page
  - HTML/CSS/javascript
- Vérifier le contenu des requêtes serveurs
- Auditer les performances du site web
- Tester le mode responsive du site web
- Outrepasser la désactivation du clic droit sur les sites web

# Console du navigateur – Attention : Self-XSS

- Technique de hack visant à tromper la victime via la console du navigateur
  - **Ne copier-collez jamais du code que vous ne comprenez pas dans la console du navigateur**

## Sources :

- <https://en.wikipedia.org/wiki/Self-XSS> – anglais
- [https://www.asafety.fr/vuln-exploit-poc/poc-xss-elever-et-exploiter-une-self-xss-via-wysinwyc/#Payload\\_Self-XSS\\_de\\_vol\\_de\\_credential](https://www.asafety.fr/vuln-exploit-poc/poc-xss-elever-et-exploiter-une-self-xss-via-wysinwyc/#Payload_Self-XSS_de_vol_de_credential) – Exemples de failles liées à la console

**Soyez malin/maligne : Utilisez  
la console du navigateur pour  
développer**

# Console du navigateur



Vidéo de présentation en anglais  
sur des fonctionnalités de la console  
(cliquez sur l'image pour y accéder)

## Source(s) :

- [https://developer.mozilla.org/en-US/docs/Tools/Web\\_Console/Keyboard\\_shortcuts](https://developer.mozilla.org/en-US/docs/Tools/Web_Console/Keyboard_shortcuts)
- <https://developers.google.com/web/tools/chrome-devtools/shortcuts>
- <https://www.youtube.com/watch?v=TcTSqhpm80Y> – anglais – présentation de fonctionnalités de la console

# Pratiquons ! - Notre premier formulaire

Appliquez ce que vous venez d'apprendre en ajoutant de nouvelles zones à votre projet avec ces nouvelles notions





**Questions ?**