



كلية العلوم
والتقنيات - مراكش
FACULTE DES SCIENCES
ET TECHNIQUES - MARRAKECH

Faculté des Sciences et Techniques
Marrakech

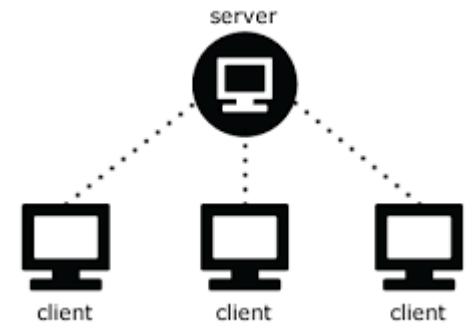


Module Programmation web

Filière LPU SIR

Prof. Sara Qassimi

18 novembre 2022



Chapitre 2: Les langages Côté Client (2)

- Objectifs:
 - CSS
 - Javascript
 - Bootstrap

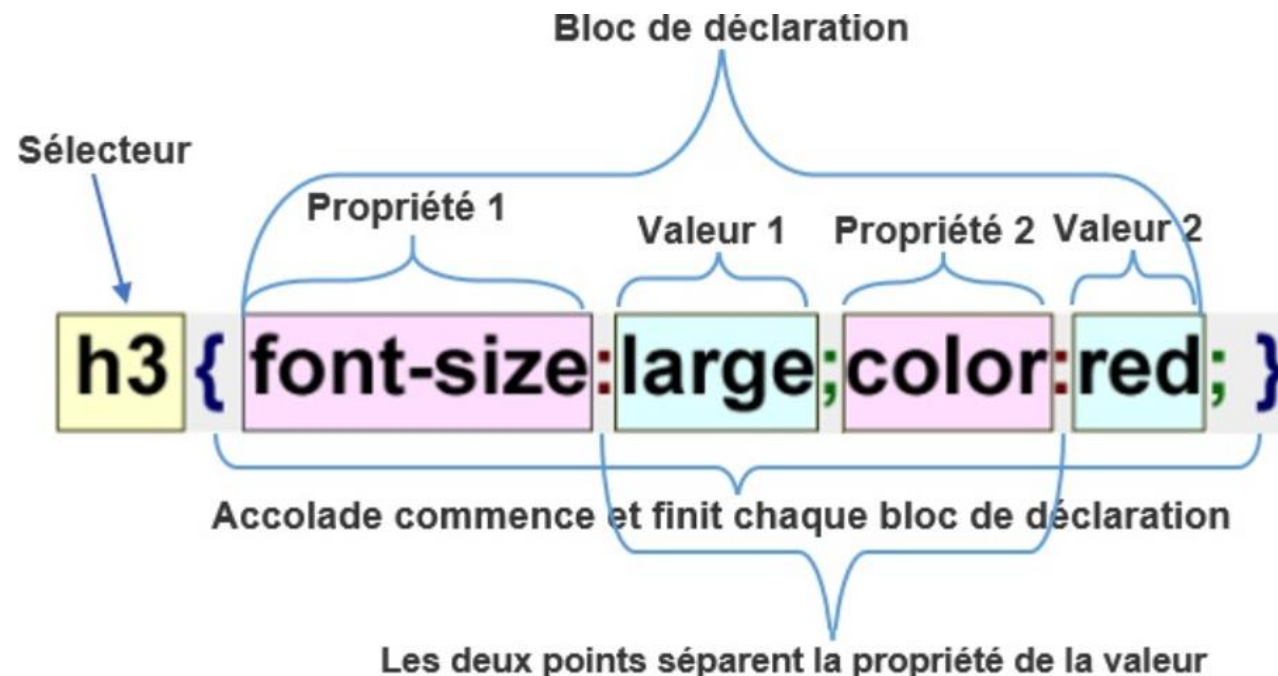
CSS (Cascading Style Sheets)



- Complète le HTML, il s'occupe de la mise en forme d'une page web.
- le HTML est brut et sans saveurs.
- Grâce au CSS, nous allons pouvoir choisir la couleur du texte, la position des différents blocs et gérer l'affichage suivant la taille des écrans.
- Un fichier indépendant se termine par l'extension **.css**.

LES SÉLECTEURS ET PROPRIÉTÉS

- Une feuille de style de CSS est composée d'un ensemble de règles. Une règle est composée de trois notions différentes :
 - **Sélecteur** : Indique le ou les éléments sur lesquels la mise en forme va s'appliquer.
 - **Propriété** : Précise la propriété CSS, nous allons les découvrir au fur et à mesure.
 - **Valeur** : Détermine la valeur de la propriété à appliquer sur le sélecteur.
- Une règle débute par le sélecteur, puis le reste du contenu est englobé entre deux accolades : { et }.



Sélecteur de type

- «sélecteur de type » (parfois appelé « sélecteur de balise » ou Sélecteur d'élément)
- Un sélecteur de type est simplement un sélecteur qui prend une seule balise HTML pour en faire de mise en forme
- Si par exemple vous devez cibler toutes les listes non ordonnées, utilisez simplement `ul {}`.
- Nous inscrivants par exemple:

`p {color : yellow ; background-color : blue;}`

- pour que le texte de tous les paragraphes figure en jaune sur fond bleue;
- Nous pouvons ainsi définir un style propre à chaque élément comme il en existe un par défaut dans les navigateurs

Sélecteurs regroupés

- Afin de réduire les déclarations répétitives dans les feuilles de style, le regroupement de sélecteurs et de déclarations est autorisé.
- On peut appliquer le même style à plusieurs éléments par la sélection de plusieurs éléments différents en les énumérant et en les séparant par une virgule dans le sélecteur. plutôt que de manipuler les définitions:

```
h1 {color : black ; background-color : red;}  
div {color : black ; background-color : red;}  
p {color : black ; background-color : red;}
```

- Nous pouvons écrire le style suivant :

```
h1,div,p {color : black ; background-color : red;}
```

Sélecteurs regroupés

- Cette possibilité de regroupement peut être utile pour définir des styles communs à un ensemble d'éléments en écrivant ce type de sélecteur pour cet ensemble, puis en ajoutant d'autres propriétés spécifiques à un des éléments de la liste.
- On définit ainsi une sorte de tronc commun à un groupe, puis on affine chacun de ses composants.
- Si nous écrivons par exemple le code suivant :

```
h1,div,p {color : black ; background-color : red;}
```

```
div {margin : 20px;}
```

- l'élément **<div>** va avoir à la fois un texte noir, un fond rouge et une marge de 20 pixels, car la propriété margin, définie uniquement pour l'élément **<div>**, s'ajoute à celles déjà définies pour le sélecteur d'éléments **h1,div,p**

Sélecteur universel

- Pour appliquer un style à tous les éléments, nous utiliserons le sélecteur universel * avant la définition d'une ou plusieurs propriétés.
- Par exemple, pour que la couleur de fond de tous les éléments soit le jaune, nous écrirons :

```
*{background-color : yellow;}
```

- Cela n'empêche pas de modifier cette couleur de fond pour un élément particulier, en la redéfinissant pour celui-ci, par exemple:

```
*{background-color : yellow;}
```

```
p{background-color : gray;}
```

- Dans ce cas, tous les éléments ont un fond jaune, sauf qui a un fond gris redéfini spécialement.

Sélecteur contextuels ou sélecteur descendants

- Les sélecteurs contextuels sont des chaînes de deux sélecteurs simples ou plus, séparés par des espaces.
- Exemple

```
<ul id="liste1">  
  <li>abc</li>  
  <li>def</li>  
</ul>
```

```
<ul id="liste2">  
  <li>xyz</li>  
  <li>rst</li>  
</ul>
```

Sélecteur contextuels ou sélecteur descendants

- Supposons qu'on veuille changer uniquement la couleur des “li” de la deuxième liste.

```
li {color: red;}
```

- ne marche pas, car elle change la couleur des “li” dans les deux listes.
- On peut utiliser un sélecteur composé de deux sélecteurs simples:

```
#liste2 li {color: red;}
```

- L'espace entre “#liste2” et “li” est très important: il veut dire “descendant de”.
- Donc: “#liste2 li” veut dire : tous les li qui sont descendants de #liste2

Sélecteur d'identifiant (id)

- Ce sélecteur simple permet de cibler un élément d'un document en fonction de la valeur de son **attribut id**.
- L'élément d'une page qui possède l'ID fourni (pour une page HTML donné, on ne peut avoir qu'un seul élément pour un ID donné).
- Un sélecteur d'identifiant (ID selector) permet, pour un document HTML, de cibler un élément grâce à la valeur de son attribut id.

```
/* L'élément avec l'identifiant id="demo" */
```

```
#demo {
```

```
  border: red 2px solid;
```

```
}
```

```
<div id="demo"> .... </div>
```

Sélecteur de classe

- Un sélecteur de classe est un sélecteur simple permet de cibler les éléments en fonction de la valeur de leur attribut class.
- Avec le sélecteur de classe il est possible d'appliquer des styles différents pour des éléments HTML d'un même type.
- **Classe qui s'applique à un seul élément HTML**
- Par exemple, si nous souhaitons que certains paragraphes (<p>) soient affichés en rouge (disons, pour mettre en évidence des paragraphes importants), et que d'autres paragraphes aient un alignement centré, nous pouvons définir deux classes pour chacun de types de paragraphe :

```
p.introduction{text-align:center}  
p.important{color:red}
```
- Dans le document HTML, l'attribut class permet de choisir la classe qui sera utilisée :

```
<p class="introduction"> Un paragraphe d'introduction </p>  
<p class="important">Un paragraphe important</p>
```

Sélecteur de classe

- **Classe qui s'applique à tous les éléments HTML**
- Pour définir une classe qui s'applique à tous les éléments HTML, et pas uniquement à un seul élément HTML, on utilise la syntaxe suivante : l'élément HTML n'est pas écrit ; on commence directement par un point, suivi du nom de la classe.

- Exemple :

`.important{color:red}`

- Ainsi, la classe important pourra être utilisée pour tous les éléments HTML :

`<h1 class="important">Un titre important </h1>`

`<p class="important">Un paragraphe important </p>`

TP1 - CSS

- Soit le code HTML suivant:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tp1 CSS</title>
</head>
<body>
  <div id="layout">
    <h2>introduction au CSS</h2>
    <p class="gauche">
      Introduit au milieu des annees 1990,
      CSS devient couramment utilise dans la conception de sites web et bien pris
      en charge par les navigateurs web dans les annees 2000.
    </p>
    <p class="droite">
      Le CSS est un langage de style qui difinit
      la presentation des documents HTML.
      Par exemple, CSS couvre les polices,
      les couleurs, les marges, les lignes, la hauteur, la largeur,
      les images d'arriere-plan, les positionnements evolues et bien d'autres chases.
    </p>
```

TP1 - CSS

- La suite du code HTML :

```
<p class="justifiee">
  Le but d'avoir le css dans vos sites
  c'est de compenser les manques du langage HTML
  en ce qui concerne la mise en page et la presentation
</p>
</div>
<button>read more</button>
</body>
</html>
```

- **Appliquer ce qui suit:**
- Un background de couleur bleu ciel ,une largeur de 500px ,un retrait de 5px , et un alignement centré pour l'élément identifié par "layout".
- Aligner le texte à gauche pour la classe "gauche" , à droite pour la classe "droite" et justifié pour la classe justifié.
- Colorer le texte en bleu pour les éléments h2 et button, et les éléments de p en rouge.
- Changer la taille du titre à "1.5em" en affichant la première lettre de chaque mot en capitale.
- Affecter une couleur jaune au background du bouton, et une largeur de 100px.
- **em** : elle **est** proportionnelle à la taille de la police de l'élément parent ou du document. Par défaut, 1 **em** = 16 px si aucune taille de police n'**est** définie.

TP1 – CSS - Correction

```
<style>
  #layout{background-color: skyblue;
width: 500px;
padding: 5px;
text-align: center;}
  .gauche{text-align: left;}
  .droite{text-align: right;}
  .justifiee{text-align: justify;}
  h2,button{color:blue}
  p{color:red}
  h2{font-size: 1.5em;
text-transform: capitalize;}
  button{background-color: yellow;}
</style>
```


Javascript



- Un langage de scripts qui, incorporé aux balises HTML, permet d'améliorer la présentation et l'interactivité des pages Web.
- Ces scripts sont gérés et exécutés par le navigateur lui-même sans faire appel aux ressources du serveur. Ces instructions seront donc traitées en direct et surtout sans retard par le navigateur.
- Permet de rendre dynamique un site internet développé en HTML.
- Un langage de script simplifié orienté objet dont la syntaxe est basée sur celle du Java.
- Un langage interprété , c'est-à-dire que le code source s'intègre directement dans la page web. Les scripts sont introduits dans le document HTML à l'aide d'une balise `<script>`.

L'accès aux éléments HTML en Javascript

- Une page HTML est ensemble d'éléments ou d'objets. Ces éléments peuvent être par exemple une balise de titre, un paragraphe ou un champ de formulaires.
- Cet ensemble d'élément constitue ce qu'on l'appelle le **DOM** de la page, qui est le **Document Object**
- L'objet **document** est un objet important, qui représente l'ensemble de l'arborescence du document analysé.
- Il possède de nombreuses **propriétés** et **méthodes** . Cinq de ces dernières permettent de « pointer » directement un ou plusieurs éléments dans le document.

Accès aux éléments à partir de l'ensemble du document

- La méthode `getElementById`

- Fonction: permet de sélectionner un élément d'identifiant donné dans une page.
- Exemple, si on a dans la page
`<p id="intro">(…)</p>`, `document.getElementById("intro")`
- permettra de sélectionner précisément l'élément p en question.

- La méthode `getElementByName`

- Fonction: permet de sélectionner les éléments portant un nom donné dans une page ; mais cette méthode n'est pas supportée par Internet Explorer ;
- Exemple: Page suivante

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Test getElementByName </title>
</head>
<body>
  <form name="repas">
    <input type="checkbox" name="dessert" value="Fraise"> Fraise <BR>
    <input type="checkbox" name="dessert" value="Banane"> Banane <BR>
    <input type="checkbox" name="dessert" value="Pomme"> Pomme <BR>
  </form>
</body>
<script type="text/javascript">
  console.log(typeof document.getElementById("dessert"));
  document.getElementById("dessert")[0].checked = true;
  document.getElementById("dessert")[1].checked = false;
  document.getElementById("dessert")[2].checked = true;
</script>
</html>
```

Test getElementByName

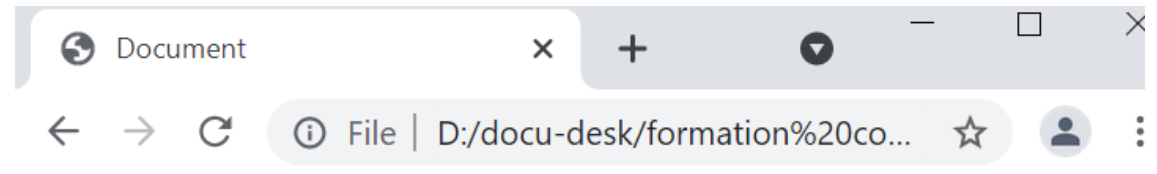
← → ↻ ⓘ File | D:/docu-desk/formation%20con... ☆ 👤 ⋮

☒ Fraise
☐ Banane
☒ Pomme

Accès aux éléments à partir de l'ensemble du document

- La méthode `getElementsByName`

- Fonction: permet de sélectionner les éléments portant un nom de balise donné dans une page.
- Exemple:



Lorem ipsum dolor sit amet consectetur adipisicing elit. Maxime hic est modi dolores facilis sed asperiores aperiam, delectus rem dolor et architecto omnis quas a ducimus, minus accusantium suscipit quisquam?

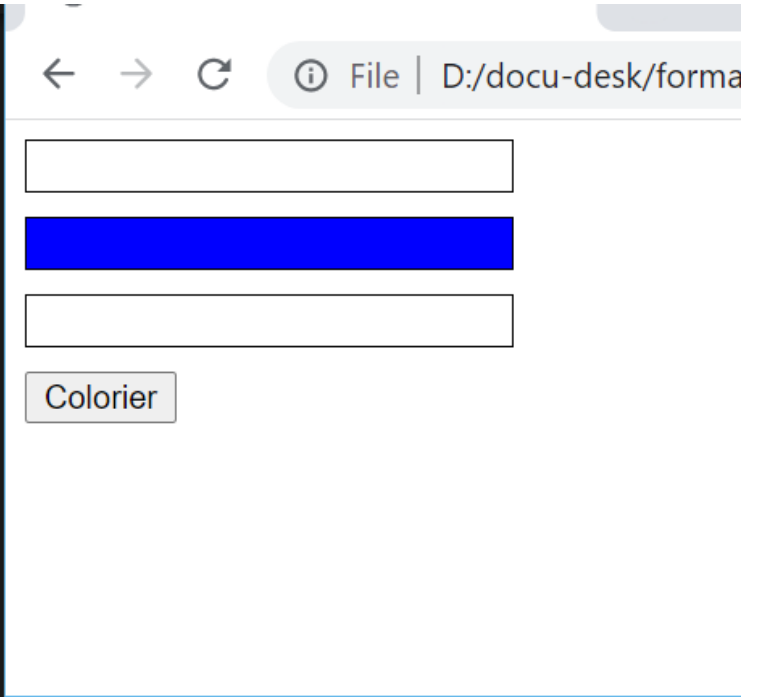
Il y a 1 éléments div HTML dans cette page

```
<body>
  <div>
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Maxime hic e
  </div>
</body>
<script type="text/javascript">
  var divs=document.getElementsByTagName("div");
  document.write("Il y a "+divs.length+" éléments div HTML dans cette page");
</script>
```

Accès aux éléments à partir de l'ensemble du document

- l'exemple suivant montre comment colorer un div en cliquant sur un bouton

```
<title>Document</title>
</head>
<style>
  div{width:200px; height:20px;
    border: 1px solid black;
    margin-bottom:10px;
  }
</style>
</head>
<body>
  <div></div>
  <div id="couleur"></div>
  <div></div>
  <button type="button" onclick="colorier()"> Colorier</button>
</body>
<script>
  function colorier(){
    document.getElementById("couleur").style.backgroundColor="blue";
  }
</script>
</html>
```



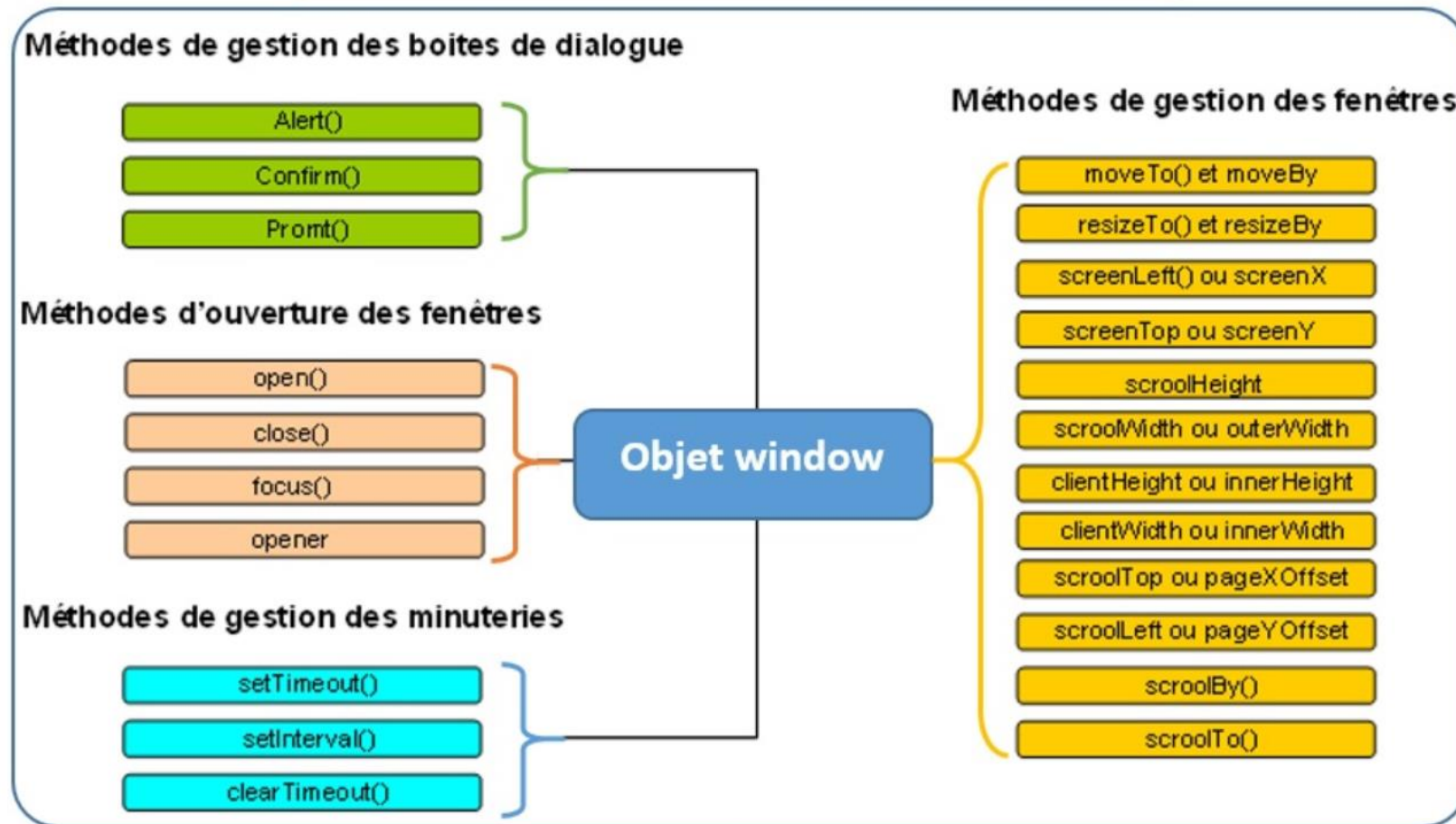
The screenshot shows a web browser window with the address bar displaying 'D:/docu-desk/forma'. The page content includes three horizontal rectangular boxes. The middle box is colored blue, while the top and bottom boxes are white. Below the boxes is a button labeled 'Colorier'.

Les objets en Javascript

- L'objet Window et ses méthodes

- L'objet window est un objet qui correspond à la fenêtre dans laquelle s'affiche une page Web. Cet objet est l'objet le plus haut de la hiérarchie javascript.

Les méthodes de l'objet window

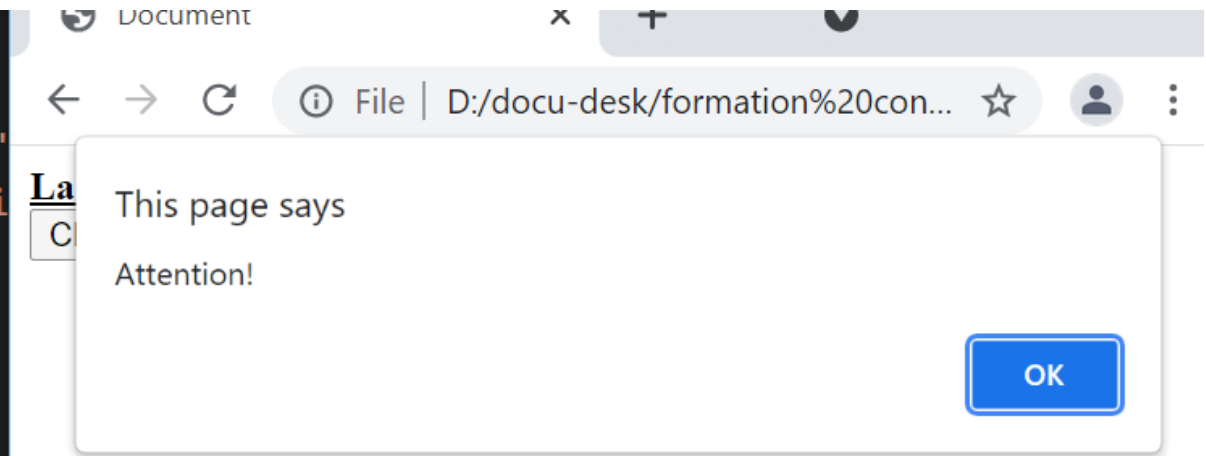


Les boîtes de dialogue en Javascript

- Une boîte de dialogue est une fenêtre qui s'affiche au premier plan suite à un événement, et qui permet :
 - Soit d'avertir l'utilisateur
 - Soit le confronter à un choix
 - Soit lui demander de compléter un champ pour récupérer une information
- JavaScript propose des boîtes de dialogue par défaut qui permettent d'interagir avec l'utilisateur.
- Il en existe 3 types:
 - Les **alert()** qui permettent d'afficher un message.
 - Les **confirm()** qui permettent de récupérer une valeur booléenne.
 - Les **prompt()** qui permettent de récupérer une valeur textuelle.

La méthode alert()

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Document</title>
  <script language="javascript">
    function msg(){
      alert("Attention!"); }
  </script>
</head>
<body>
  <b><u>La méthode Alert()</u></b><br>
  <form name= "f1">
    <input type="button" value="Cliquer ici !!! " onclick="msg()" >
  </form>
</body>
</html>
```



La méthode Confirm()

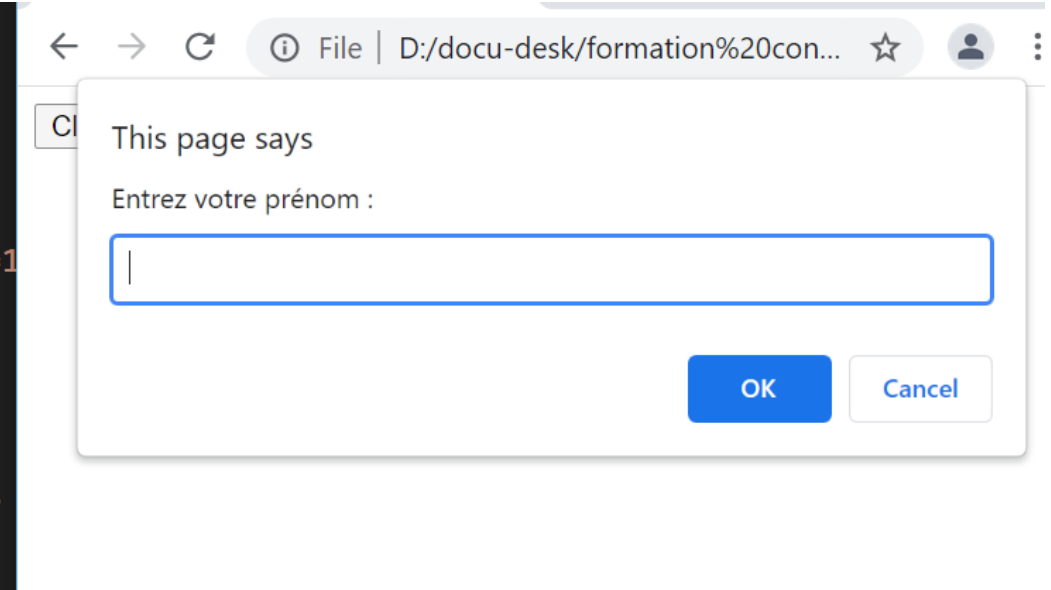
- La boîte de dialogue **confirm()** permet d'afficher un message et deux boutons :
- un bouton « OK » et un bouton « Annuler »
- Cette fonction retourne une valeur booléenne qui vaut « true » si c'est le bouton OK qui est cliqué et retourne « false » si c'est le bouton « Annuler » qui est cliqué.
- <Lorsque l'utilisateur appuie sur OK" la méthode renvoie la valeur true. Elle renvoie false dans le cas contraire

```
if ( confirm( "Message à afficher" ) ) {  
    // Code à exécuter si le l'utilisateur clique sur "OK"  
} else {  
    // Code à exécuter si l'utilisateur clique sur "Annuler"  
}
```

La méthode prompt()

- La boîte de dialogue **prompt()** permet d'afficher un message, un champ à remplir (input de type texte) et un bouton « OK ». Cette fonction retourne la valeur qui a été entré dans le champ par l'utilisateur.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Document</title>
  <script>
    function msg(){
      var userName = prompt('Entrez votre prénom :');
      alert(userName); // Affiche le prénom entré par l'utilisateur
    }
  </script>
</head>
<body>
  <form name= "f1">
    <input type="button" value="Cliquer ici !!! " onclick="msg()" >
  </form>
</body>
</html>
```



L'objet Date en Javascript

- L'objet **Date** en JavaScript permet de gérer les informations relatives au temps, soit la date et l'heure.
- Apporte une gestion complète des dates en proposant des accès aux différentes parties la constituant ainsi que des méthodes de manipulation pour effectuer des calculs ou des modifications.
- Se basent sur une valeur de temps qui est le nombre de millisecondes depuis 1er janvier 1970 minuit.
- On utilise cet objet par exemple pour créer un calendrier pour pouvoir sélectionner la date de naissance sur un formulaire d'inscription plus facilement, ou afficher l'horloge sur une page Web, ou encore calculer la durée de séjour d'un client qui compte réserver une chambre d'hôtel sur un site d'hébergement.

L'objet Date en Javascript

- Syntaxe et utilisation

- L'objet Date en JavaScript présente une particularité dans la mesure où il faut d'abord le créer, avant de pouvoir l'utiliser

```
variable=new Date();
```

```
variable=new Date(année, indexMois[, jour[, heures[, minutes[, secondes[,  
millisecondes]]]]]);
```

- Si les parenthèses de Date() sont vides, cela signifie qu'on veut créer l'objet date à partir de la date courante du système. Par contre on peut toujours faire référence à une date différente en la précisant entre les parenthèses comme ceci:

```
variable=new Date(2021,11,28,00,00,00)
```

L'objet Date en Javascript

- Quelques méthodes
- **getFullYear()**: permet de retourner l'année sur deux chiffres.
- **getFullYear()** : permet de retourner l'année sur 4 chiffres.
- **getDate()**: retourne la date du mois comprise entre 1 et 31.
- **getDay()**: retourne un indice numérique qui représente le jour de la semaine. 0 pour dimanche, 1 pour lundi, ... , 6 pour samedi.
- **getMonth()**: retourne un indice numérique qui représente le mois. 0 pour janvier, 1 pour février, ... , 11 pour décembre.
- **getHours()**: retourne l'heure. Si la date crée fait référence à 7h du matin, la méthode retournera 7 et non 07.
- **getMinutes()**: retourne les minutes.
- **getSeconds()**: retourne les secondes.

Exercice : Date en Javascript

- Utilisez la méthode `Get` pour afficher la date actuelle, au format JJ-MM-AAAA, en JavaScript
- Si vous préférez un format différent, changez simplement l'ordre des commandes.
- Affichez les heures, les minutes et les secondes après la date affichée précédemment

Correction-exercice : Date en Javascript

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Exercice Date en javascript</title>
</head>
<body onload="afficherDateActuelle()">
  <div id='zoneSaisie'></div>
</body>
<script>
  function afficherDateActuelle()
  {
    var today = new Date();
    var date = today.getDate()+'-'+(today.getMonth()+1)+'-'+today.getFullYear();
    var time = today.getHours() + ':' + today.getMinutes() + ':' + today.getSeconds();
    alert(date+ ' ' + time);
    document.getElementById('zoneSaisie').innerHTML = date+ ' ' +time;
  }
</script>
</html>
```

← → × ⓘ File | D:/docu-desk/formation%20con... ☆ 👤 ⋮

This page says

27-11-2021 20:38:40

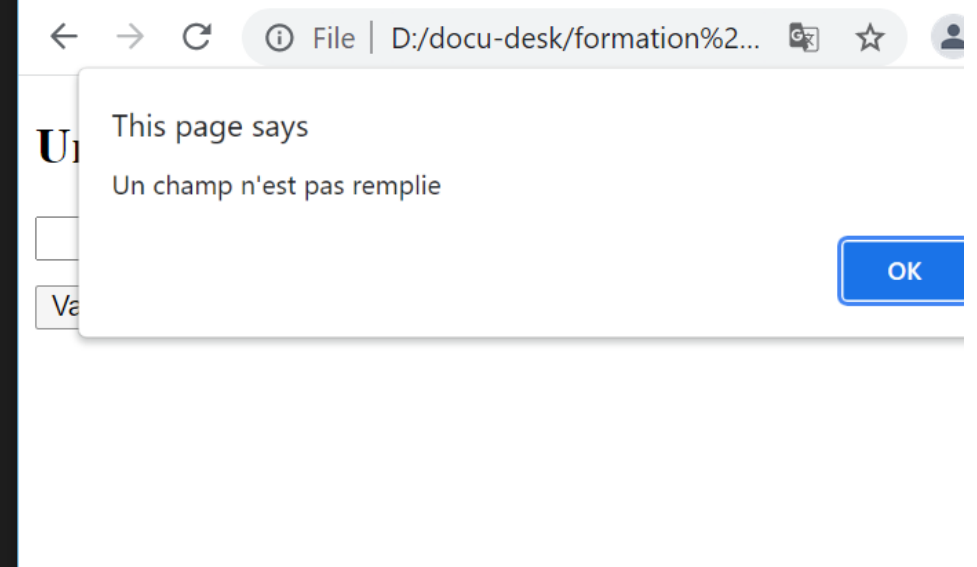
OK

Formulaire et javascript

Dans un formulaire, vous voulez que certains champs soient obligatoirement remplis.

```
<style>
#bouton {margin-top: 12px;}
</style>
<script>
function verif_champ()
{
var mots_cles =document.getElementById("input").value;
    if (mots_cles == "")
    {
        alert("Un champ n'est pas remplie");
        return false;
    }
    return true;
}
</script>
<body>

<h2> Un champs de texte vide (Testez)</h2>
<form>
<input type="text" id="input" name="mots_cles" value=""><br>
<input type="button" id="bouton" value="Valider" onclick="verif_champ()">
</form>
</body>
</html>
```



Validation d'un formulaire n HTML5

La validation à l'aide d'attributs dans les balises HTML, elle aura toujours lieu et ce, même si le JavaScript est désactivé.

- La syntaxe HTML5 permette la validation des données saisies dans un formulaire grâce aux éléments:
 - **pattern**: Cet attribut spécifie une expression régulière sur laquelle la valeur de l'élément `<input>` est vérifiée lors de la soumission du formulaire.
 - **placeholder**: Cet attribut spécifie un indice court décrivant la valeur attendue d'un champ de saisie (par exemple, une valeur exemple ou une brève description du format attendu).
 - **required**: Cet attribut spécifie qu'un champ de saisie doit être rempli avant de soumettre le formulaire.
 - **disabled**: Cet attribut que l'élément `<input>` doit être désactivé, il peut être défini pour empêcher un utilisateur d'utiliser l'élément `<input>` jusqu'à ce qu'une autre condition soit remplie (par exemple, sélectionner une case à cocher, etc.).
 - **max**: Cet attribut spécifie la valeur maximale pour un élément `<input>`
 - **min**: Cet attribut spécifie la valeur minimale pour un élément `<input>`
 - **type**: Cet attribut spécifie le type de média Internet (anciennement appelé type MIME) du document lié, il n'est utilisé que si l'attribut href est défini.


Validation d'un formulaire en HTML5

required

- L'attribut "`required`" permet de rendre obligatoire le remplissage d'un champ et bloquer la validation du formulaire si l'un des champs (concernés par cet attribut) n'a pas été renseigné.
- Lorsqu'il est présent, cet attribut spécifie qu'un champ de saisie doit être rempli avant de soumettre le formulaire.
- L'attribut `required` assure que le champs de saisie ne peut pas être laissée vide.

Inscription

Nom:

 Veuillez renseigner ce champ.

```
<input type="text" required="required">
```

Si l'utilisateur a cliqué sur le bouton de soumission de formulaire alors qu'un champ requis est vide, la soumission du formulaire n'aura pas lieu et un message sera automatiquement affiché

- Cet attribut n'a pas besoin de valeur car sa seule présence suffit, cependant les syntaxes suivantes sont admises :

```
<input type="text" required>
```

Validation d'un formulaire en HTML5

placeholder

- L'attribut `placeholder` peut être placé sur les éléments : `<input>` : de type `text`, `search`, `password`, `url`, `tel`, `email` `<textarea>`
- `placeholder` est un attribut qui permet de renseigner un texte indicatif par défaut dans un champ de formulaire.
- C'est une valeur qui s'efface dès que l'utilisateur active le champ de formulaire.

Inscription

Nom

Ici on a utilisé l'attribut `placeholder` pour ajouter un texte indicatif par défaut dans un champ de formulaire.

- L'attribut `placeholder` peut être placé sur les éléments :
 - `<input>` : de type `text`, `search`, `password`, `url`, `tel`, `email`
 - `<textarea>`

Validation d'un formulaire en HTML5

pattern

- L'attribut `pattern` permet de valider le format attendu à l'aide d'une expression régulière. Un message sera affiché si l'information entrée est erronée.
- À l'aide de l'attribut `pattern`, vous pouvez déclarer vos propres exigences de validation à l'aide d'expressions régulières.
- Cet attribut vous permet d'ajouter une validation de base sans recourir à JavaScript.
- Des patterns par défaut existent pour certains types de champ, comme les champs de type email ou url par exemple, qui attendent des formats spécifiques.
- Une `expression régulière` est une chaîne formalisée de caractères définissant un motif. Par exemple, `[a-zA-Z0-9]+` un motif qui correspond à une chaîne de n'importe quelle longueur, à condition que la chaîne ne contienne que des lettres minuscules (`a-z`), des lettres majuscules (`A-Z`) ou des nombres (`0-9`).
- Les expressions régulières, ou plus communément `regex` (contraction de `regular expression`) permettent de représenter des modèles de chaînes de caractère.

Inscription

Nom:

Code:

Ce champ attend une valeur numérique d'au moins 3 chiffres précédés d'une lettre majuscule comprise entre A et F. Si le format n'est pas respecté, le navigateur en informe l'utilisateur.

Inscription

Nom:

Code:

! Veuillez respecter le format requis.

Validation d'un formulaire en HTML5

1. Créer un champ qui attend une valeur numérique d'au moins 5 chiffres précédés d'une lettre majuscule comprise entre A et F. Si le format n'est pas respecté, le navigateur doit informer l'utilisateur.

```
<input type="text" pattern="[A-F][0-9]{5}">
```

2. Créer un champ qui attend une valeur numérique d'au moins 5 chiffres précédés de deux lettres majuscules comprises entre A et F. Si le format n'est pas respecté, le navigateur doit informer l'utilisateur.

```
<input type="text" pattern="[A-F]{2}[0-9]{5}">
```

3. Créer un champ de saisie qui ne peut contenir que trois lettres (pas de chiffres ni de caractères spéciaux)

```
<input type="text" pattern="[A-Za-z]{3}">
```

4. Créer un champ qui attend une valeur composé que de lettres minuscules; pas de majuscules, de chiffres ou d'autres caractères spéciaux autorisés. De plus, la longueur de la valeur ne doit pas dépasser 15 caractères.

```
<input type="text" pattern="[a-z]{1,15}">
```

5. Créer un champ de type = "mot de passe" qui doit contenir au moins 8 caractères.

```
<input type="text" pattern=".{8,}">
```

6. Créer un champ de type = "mot de passe" qui doit contenir au moins 8 caractères d'au moins un chiffre et une lettre majuscule et minuscule.

```
<input type="text" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}">
```

TP: Reproduire en HTML 5 le formulaire, ci-dessous.

Vos coordonnées

Nom :

vous nom

Prénom :

vous prénom

Date de naissance :

jj/mm/aaaa

Mail :

vous mail

Votre site :

Code :

Genre :

☐ Homme
☐ Femme

Pays :

Votre pays ▾

Vos goûts

☐ Tarte aux fraises à la crème d'amandes

☐ Tarte aux prunes

☐ Tarte rustique aux mirabelles de Lorraine, noisettes et miel

Décrivez vos goûts en détail

Envoyez nous votre photo

Choisir un fichier

Aucun fichier choisi

Effacer

Envoyer



- est un framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Ce framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les smartphones.
- C'est une boîte à outils open source pour le développement avec HTML, CSS et JS. C'est une boîte à outils puissante permettant de créer et de créer des pages Web et des applications Web. C'est un projet libre et open source, hébergé sur GitHub , et créé à l'origine par (et pour) Twitter .
- les développeurs Web peuvent se concentrer sur le travail de développement, sans se soucier de la conception, et obtenir rapidement un site Web de belle apparence. Inversement, il offre aux concepteurs Web une base solide pour la création de thèmes

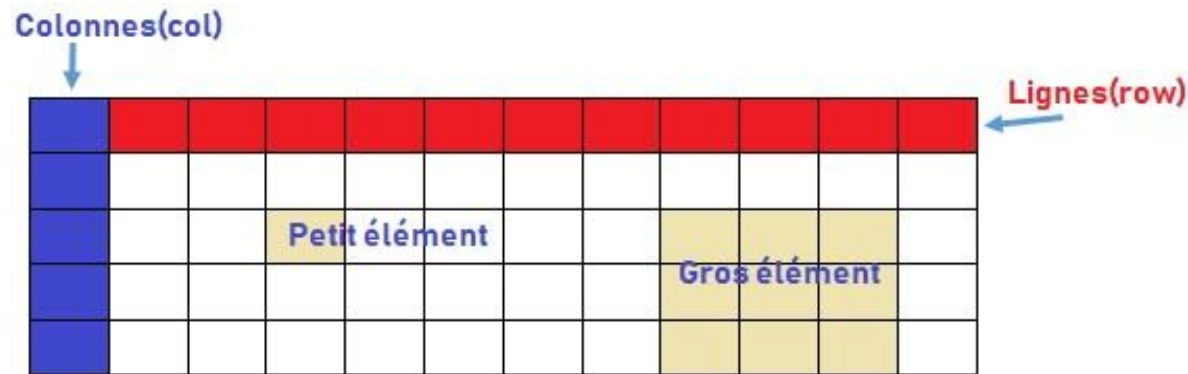
Squelette d'une page web avec Bootstrap

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after* these tags -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="bootstrap/dist/js/bootstrap.min.js"></script>
  </body>
</html>
```

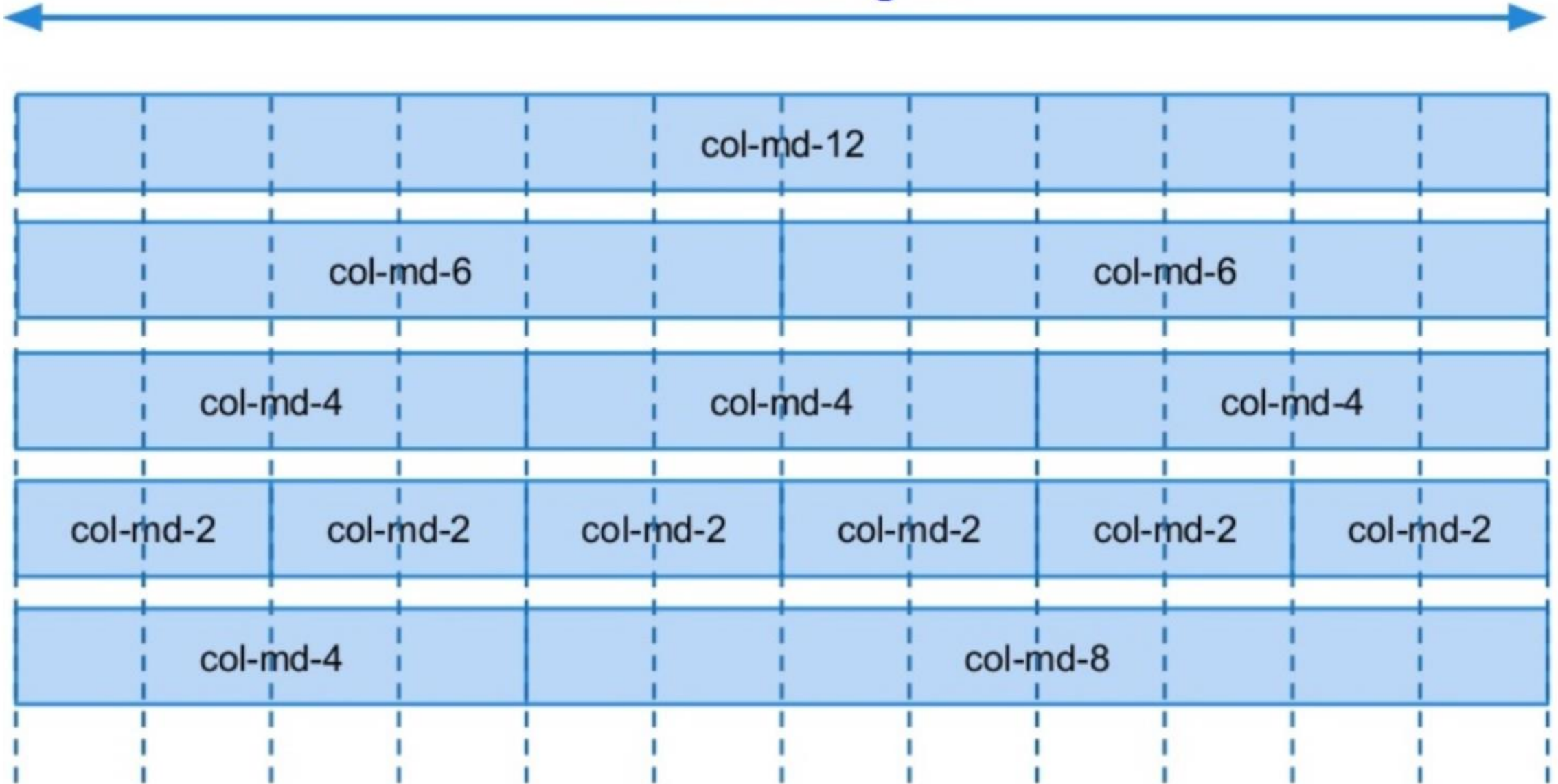
Le système de grille Bootstrap

- En conception web, comme en typographie et en publication assistée par ordinateur, une grille est un quadrillage invisible sur lequel sont calés les contenus. Elle contient donc des colonnes (col) et parfois des lignes (row).
- Une grille est découpée en rangées (appelées « **row** », parce que tout est en anglais) et colonnes « **col** »
- C'est un système de grille fluide de 12 colonnes qui se réarrangent selon la taille de l'écran de l'utilisateur en fonction des critères que vous définissez.
- Chaque **row** (ce qui signifie un groupe de contenu) de votre code sera divisée en groupes horizontaux de colonnes, et Bootstrap vous fournit des classes prédéfinies pour définir la largeur de chaque colonne.



Le système de grille Bootstrap

12 colonnes/ligne



Le système de grille Bootstrap

- Les systèmes de grille sont utilisés pour créer des mises en page à travers une série de lignes et de colonnes hébergeant votre contenu.

Comment écrire des pages à l'aide d'une grille

```
<div class="container">
  <!-- chaque ligne doit avoir 12 colonnes -->
  <div class="row">
    <div class="col-md-4">
      <!-- content -->
    </div>
    <!-- besoin de remplir 8 colonnes supplémentaires -->
  </div>
</div>
```



- Les lignes doivent être placées dans une classe **.container** pour un alignement et un remplissage corrects.
- Des classes de grille prédéfinies telles que **.row** et **col-md-4** sont disponibles pour créer rapidement des présentations de grille.

Le système de grille Bootstrap

- La classe `container` (`.container`)
- La classe conteneur est utilisée pour créer un contenu encadré. Il existe deux classes de conteneur dans Bootstrap: `.container` et `.container-fluid`
- Les conteneurs sont les composants les plus basiques de Bootstrap. Ils permettent de structurer la grille.
- En utilisant la classe `.container`, on centre la grille sur une largeur maximale fixe. Celle-ci est variable suivant la largeur de la page.
- En utilisant `.container-fluid`, on permet à la grille d'occuper 100 % de la largeur de la page.
- Testez le code suivant et faites agrandir la largeur de votre page pour voir la différence.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1
    <!-- The above 3 meta tags *must* come first in the head; any other
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="bootstrap/dist/css/bootstrap.min.css" rel="stylesheet"
  </head>
  <body>
    <div class="container-fluid" style="background: #cdd;">
      .container-fluid
    </div>
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual f
    <script src="bootstrap/dist/js/bootstrap.min.js"></script>
  </body>
</html>

```

← → ↻ ⓘ File | D:/docu-desk/formation%20conti

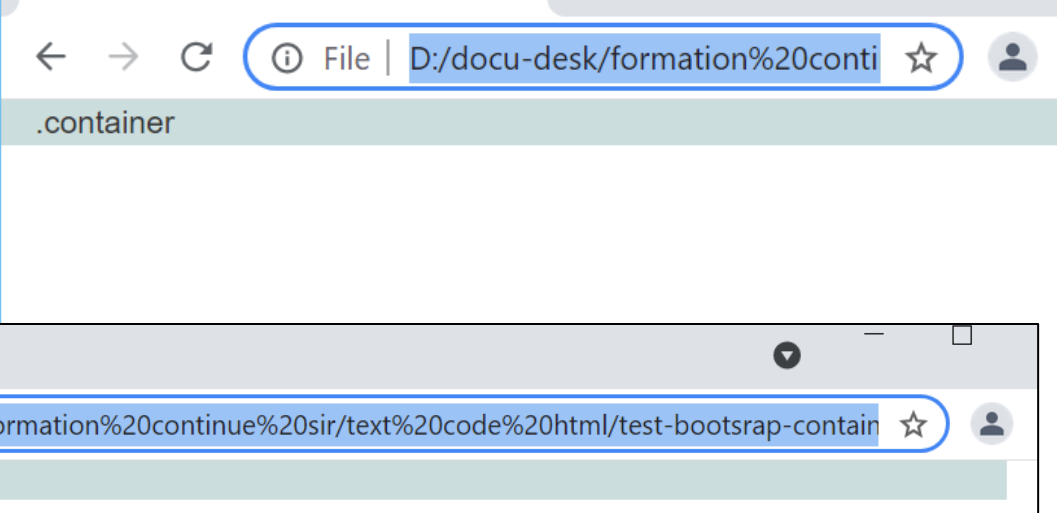
.container-fluid

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head, any other
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <div class="container" style="background-color: #cdd;">
      .container
    </div>
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files
    <script src="bootstrap/dist/js/bootstrap.min.js"></script>
  </body>
</html>

```



La classe `.container`

- possède une largeur maximale fixe suivant la largeur de la page web. En effet, Bootstrap réagit suivant plusieurs formats. Le tableau ci-après récapitule les points clés.

	Extra small <576px	Small ≥ 576px	Medium ≥ 768px	Large ≥ 992px	Extra large ≥ 1200px
Largeur maximale du conteneur	auto	540px	720px	960px	1140px
Préfixe des classes	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
Type d'écran	Smartphone	Smartphone & tablette	Tablette	Écran d'ordinateur de bureau	Grand écran et télévision

- `.col-` (très petits appareils – largeur d'écran inférieure à 576 pixels)
- `.col-sm-` (petits appareils – largeur de l'écran égale ou supérieure à 576 pixels)
- `.col-md-` (périphériques de taille moyenne – largeur d'écran égale ou supérieure à 768 pixels)
- `.col-lg-` (gros périphériques – largeur d'écran égale ou supérieure à 992px)
- `.col-xl-` (xlarge devices – largeur d'écran égale ou supérieure à 1200px)


```

<!-- Bootstrap -->
<link href="bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <div class="container" >
    <h4>Exemple de liste de définition horizontale</h4>
    <dl class="row">
      <dt class="col-sm-2">Description 1</dt>
      <dd class="col-sm-2">Elément 1</dd>
    </dl>
    <dl class="row">
      <dt class="col-sm-2">Description 2</dt>
      <dd class="col-sm-2">Elément 2</dd>
    </dl>
  </div>
  <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
  <script src="jquery.min.js"></script>
  <!-- Include all compiled plugins (below), or include individual files as needed -->
  <script src="bootstrap/dist/js/bootstrap.min.js"></script>
</body>
</html>

```

← → ↻ ⓘ File | D:/docu-desk/formation%20conti ☆ ⓘ

Exemple de liste de définition horizontale

Description 1

Elément 1

Description 2

Elément 2

← → ↻ ⓘ File | D:/docu-desk/formation%20continue%20s ⓘ

Exemple de liste de définition horizontale

Description 1

Elément 1

Description 2

Elément 2

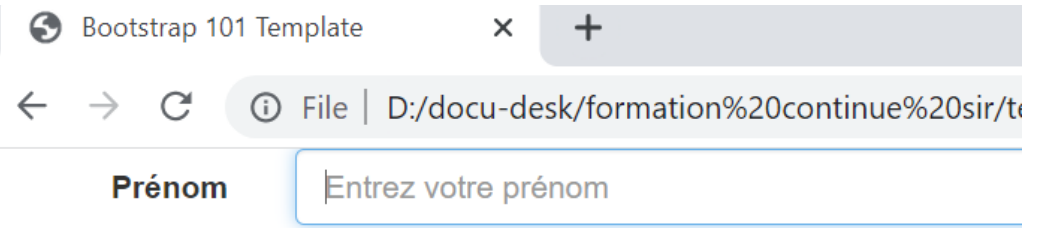
Les Contrôles de formulaire pris en charge en Bootstrap

- Les `<input>`

La classe `.form-control` pour mettre en forme la majorité des champs de formulaire et notamment les éléments `input`, `select` et `textarea`.

La classe `.form-group` pour grouper des label avec le champ de formulaire correspondant.

```
</head>
<body>
  <form class="form-horizontal" role="form">
    <div class="form-group">
      <label for="prenom"
        class="col-sm-2 control-label">Prénom</label>
      <div class="col-sm-8">
        <input type="text"
          class="form-control" id="prenom"
          placeholder="Entrez votre prénom">
      </div>
    </div>
  </form>
</body>
```



Bootstrap 101 Template

← → ↻ File | D:/docu-desk/formation%20continue%20sir/t

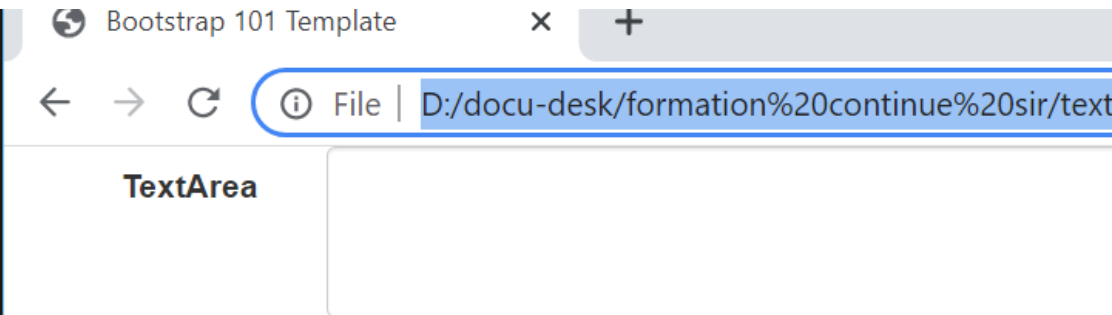
Prénom

Entrez votre prénom

Les Contrôles de formulaire pris en charge en Bootstrap

- Les `<textarea>`

```
</head>
<body>
  <form class="form-horizontal" role="form">
    <div class="form-group">
      <label for="prenom"
        class="col-sm-2 control-label">TextArea</label>
      <div class="col-sm-8">
        <textarea class="form-control" rows="3"></textarea>
      </div>
    </div>
  </form>
</body>
```



Les Contrôles de formulaire pris en charge en Bootstrap

- Les `<checkbox>`

```
<body>
  <form class="form-horizontal" role="form">
    <label for="name">Exemple de case à cocher par défaut </label>
    <div class="checkbox"> <label>
      <input type="checkbox" value="">Option 1</label>
    </div>
    <div class="checkbox">
      <label><input type="checkbox" value="">Option 2</label>
    </div>
    <div class="checkbox"> <label>
      <input type="checkbox" value="">Option 3</label>
    </div>
    <div class="checkbox"> <label>
      <input type="checkbox" value="">Option 4</label>
    </div>
```

Les Contrôles de formulaire pris en charge en Bootstrap

- Les `<checkbox>` (la suite)

`.checkbox-inline` : pour une série de cases à cocher afin que les contrôles apparaissent sur la même ligne.

```
<label for="name">Exemple de case à cocher en ligne </label>
  <div>
    <label class="checkbox-inline">
      <input type="checkbox" id="inlineCheckbox1"
        value="option1"> Option 1
    </label>
    <label class="checkbox-inline">
      <input type="checkbox" id="inlineCheckbox2"
        value="option2"> Option 2
    </label>
    <label class="checkbox-inline">
      <input type="checkbox" id="inlineCheckbox1"
        value="option1"> Option 3
    </label>
    <label class="checkbox-inline">
      <input type="checkbox" id="inlineCheckbox2"
        value="option2"> Option 4
    </label>
  </div>
</form>
```

← → ↺ (i) File | D:/docu-desk/formation%

Exemple de case à cocher par défaut

☐ Option 1

☐ Option 2

☐ Option 3

☐ Option 4

Exemple de case à cocher en ligne

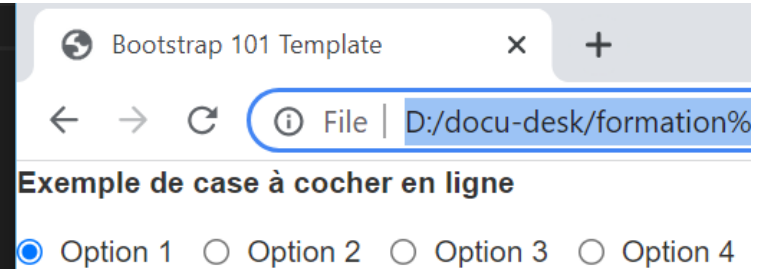
☐ Option 1 ☐ Option 2 ☐ Option 3 ☐ Option 4

Les Contrôles de formulaire pris en charge en Bootstrap

- Les `<radio>`

`.radio-inline` pour une série de radios afin que les contrôles apparaissent sur la même ligne.

```
<form class="form-horizontal" role="form">
  <label for="name">Exemple de case à cocher en ligne </label>
  <div>
    <label class="radio-inline" >
      <input type="radio" name="optionsRadiosinline"
        value="option1" checked> Option 1
    </label>
    <label class="radio-inline" name="optionsRadiosinline">
      <input type="radio" name="optionsRadiosinline"
        value="option2"> Option 2
    </label>
    <label class="radio-inline" name="optionsRadiosinline">
      <input type="radio" name="optionsRadiosinline"
        value="option3"> Option 3
    </label>
    <label class="radio-inline" name="optionsRadiosinline">
      <input type="radio" name="optionsRadiosinline"
        value="option4"> Option 4
    </label>
  </div>
</form>
```



Les Contrôles de formulaire pris en charge en Bootstrap

- Les `<select>`

```
<body>
  <form role="form">
    <div class="form-group">
      <label for="name">Liste de sélection</label>
      <select class="form-control">
        <option>1</option>
        <option>2</option>
        <option>3</option>
        <option>4</option>
        <option>5</option>
      </select>
      <label for="name">Liste de sélection multiple</label>
      <select multiple class="form-control">
        <option>1</option>
        <option>2</option>
        <option>3</option>
        <option>4</option>
        <option>5</option>
      </select>
    </div>
  </form>
```

← → ↻ ⓘ File | D:/docu-desk/formation%20contin

Liste de sélection

5

Liste de sélection multiple

1
2
3
4
5

TP: Créer une page HTML5 et utiliser la méthode local pour appeler bootstrap
Soit le contenu <body>de votre page (à recopier)

```
<body>
  <div class="container">
    <div class="row">
      <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">
        <div class="card card-signin my-5">
          <div class="card-body">
            <h5 class="card-title text-center"><b>Se connecter</b></h5>
            <form class="form-signin" action="LoginServlet"
method="post" autocomplete="off">
              <div class="form-label-group">
                <input type="email" id="inputEmail" name="username" class="form-control"
placeholder="Adresse e-mail" required autofocus
value="" autocomplete="off">
                <label for="inputEmail">Adresse e-mail</label>
              </div>
              <div class="form-label-group">
                <input type="password" id="inputPassword" name="password" class="form-
control" placeholder="Mot de passe" required value="" autocomplete="off">
                <label for="inputPassword">Mot de passe</label>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
```


Utilisez les balises HTML5 appropriées pour indiquer les en-têtes, les pieds de pages. Assurez-vous qu'ils s'affichent correctement dans votre navigateur.

```
<div class="custom-control custom-checkbox mb-3">
    <input type="checkbox" class="custom-control-input"
id="customCheck1">
    <label class="custom-control-label" for="customCheck1">Se souvenir de
moi</label>
    <label class="motpasse" for="customCheck1">Mot de passe
oublié</label>
    </div>
    <button class="btn btn-lg btn-primary btn-block text-uppercase"
type="submit">Se connecter</button>
    </form>
</div>
</div>
</div>
</div>
</div>
</body>
```

Bootstrap 101 Template

←

→

↻

File |

D:/docu-desk/formation%20continue%20sir/text%20code%20h

☆

Se connecter

Adresse e-mail

Adresse e-mail

Mot de passe

Mot de passe

☐ Se souvenir de moi

Mot de passe oublié

SE CONNECTER