

Les classes Bootstrap  
La structuration  
Le regroupement  
La validation

FORMULAIRE



Les classes

- Le composant Reboot de Bootstrap contient des modificateurs de style CSS associés aux balises HTML ➔ modifications de base pour plus d’élégance, de consistance et de simplicité.
  - Un style CSS est appliqué par défaut aux balises input de HTML
  - On peut ne pas spécifier de classes CSS aux éléments d’un formulaire
- Bootstrap propose différentes solutions d’affichage des formulaires

Email address

We'll never share your email with anyone else.

Password

☐ Check me out

Submit

Standard

Email address

Password

Label flottant

Email

Password

Radios

☒ First radio

☐ Second radio

☐ Third disabled radio

☐ Example checkbox

Sign in

Gestionnaire de placement



# Les classes

Bootstrap définit plusieurs classes spécifiques aux éléments d'un formulaire HTML :

- champ texte, fichier, zone d'écriture, date, etc.
  - La balise div contenant les constituants d'un champ (label, input)
    - Rien → comportement standard
    - `.form-floating` → label flottant (attention à ne pas spécifier de classe pour le label de l'input)
  - La balise label contenant le nom du champ → `.form-label`
  - La balise input
    - `.form-control` → affichage standard
    - `.form-control-plaintext` → affichage du contenu sans bordure en lecture seulement
  - La balise div contenant une aide à la saisie → `.form-text`
- Curseur numérique (Range) → `.form-range`
- Les listes
  - La balise select → `.form-select`

# Les classes

- Case à cocher et Bouton radio
  - La balise div contenant la case à cocher ou le bouton radio et son label
    - `.form-check` → mise en forme d'une div pour case à cocher ou bouton radio
    - `.form-switch` → utiliser la mise en forme d'un bouton switch
    - `.form-check-inline` → grouper la case à cocher ou bouton radio sur la même ligne
    - `.form-check-reverse` → placer le contenu du coté opposé
  - La balise label contenant le texte
    - `.form-check-label` → affichage standard
    - `.btn btn-primary` → affichage sous forme de bouton à état (toggle button)
  - La balise input
    - `.form-check-input` → affichage standard
    - `.btn-check` → affichage sous forme de bouton à état

# La structuration avec le gestionnaire de placement

- Pour personnaliser le rendu d'un formulaire, on utilise :
  - Le système de grille (row, col)
  - Les classes Gutters → padding entre les colonnes du système de grille (\* ∈ [0,5])
    - `gx-*` ⇔ Horizontal
    - `gy-*` ⇔ Vertical
    - `g-*` ⇔ Horizontal et Vertical
  - Pour bien aligner le label en face de son input, il faut utiliser la classe `.col-form-label`
  - Exemple :

```
<form>
  <div class="row mb-3">
    <label for="inputEmail" class="col-sm-2 col-form-label">Email</label>
    <div class="col-sm-10">
      <input type="email" class="form-control" id="inputEmail">
    </div>
  </div>
  <div class="row mb-3">
    <label for="inputPassword" class="col-sm-2 col-form-label">Password</label>
    <div class="col-sm-10">
      <input type="password" class="form-control" id="inputPassword">
    </div>
  </div>
  <button type="submit" class="btn btn-primary">Sign in</button>
</form>
```

# Le regroupement

- C'est un moyen d'étendre les champs d'un formulaire en ajoutant du texte, des boutons, des groupes de boutons, des sélections, etc.
- Objectif : simplifier la saisie, améliorer l'ergonomie, mieux informer l'utilisateur
- Procédure :
  - Créer une balise div de regroupement ayant la classe `.input-group`
  - Placer le label du champs en dehors de cette div de regroupement
  - Créer dans la div de regroupement :
    - Des balises span pour le texte ayant la classe `.input-group-text`
    - Des boutons avec la balise button ayant la classe `.btn`
    - Des boutons ayant une liste déroulante avec la balise button ayant les classes `.btn` et `.dropdown-toggle` ainsi qu'un attribut `data-bs-toggle="dropdown"` et une liste d'éléments du menu ayant la classe `.dropdown-menu` avec des items ayant la classe `.dropdown-item`
    - Une ou plusieurs balises inputs pour la saisie ayant la classe adéquate à la valeur de son type
- Exemples :

# Le regroupement

```
<form class="my-3">
  <label for="impotInput">Montant de l'impot</label>
  <div class="input-group">
    <input id="impotInput" type="number" min="0" class="form-control" value="0">
    <span class="input-group-text">.00</span>
    <span class="input-group-text">€</span>
  </div>
</form>
```

# Le regroupement

```
<div class="input-group">
  <span class="input-group-text">Prénom et nom</span>
  <input type="text" aria-label="prénom" class="form-control">
  <input type="text" aria-label="nom" class="form-control">
</div>
<div class="input-group">
  <button class="btn btn-outline-secondary dropdown-toggle" type="button"
data-bs-toggle="dropdown" aria-expanded="false">Type de taxe</button>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Impôt sur le revenu</a></li>
    <li><hr class="dropdown-divider"></li>
    <li><a class="dropdown-item" href="#">Taxe foncière</a></li>
    <li><a class="dropdown-item" href="#">Taxe d'habitation</a></li>
  </ul>
  <input id="impotInput" type="text" min="0" class="form-control"
value="0">
  <span class="input-group-text">.00</span>
  <span class="input-group-text">€</span>
</div>
```

# La validation

- Bootstrap se base sur la validation intégrée au formulaire HTML5 assurée par le navigateur.
- Par défaut, il affiche dans des info-bulles les messages de validation du navigateur.
- Remarque : la validation HTML5 opère quand l'utilisateur appuie sur le bouton submit pour envoyer les données au serveur.
- Deux stratégies de fonctionnement pour fournir des informations sur la validité des champs
  - Avant l'envoi du formulaire → Déclarer la classe `.was-validated` dans la balise form
  - Après l'envoi du formulaire → Déclarer la classe `.needs-validation` dans la balise form
 Pour ce dernier cas, il faut aussi :
  1. Désactiver la validation par défaut du navigateur en ajoutant l'attribut `novalidate` dans la balise form
  2. Ajouter un script js d'interception de l'événement submit du formulaire et de désactivation de l'envoi s'il y a des erreurs de saisie

# La validation

- Bootstrap permet d'afficher des messages de validation personnalisés avec des styles personnalisés dans des info-bulles ou incrustés dans le formulaire lui-même
  - Incrusté au formulaire, dans une div ayant la classe
    - `.valid-feedback` → message affiché quand le champ est valide
    - `.invalid-feedback` → message affiché quand le champ est invalide
  - Infobulle, dans une div ayant la classe `.position-relative` contenant l'input et une div ayant la classe
    - `.valid-tooltip` → message affiché quand le champ est valide
    - `.invalid-tooltip` → message affiché quand le champ est invalide

## La validation ► exemple ► HTML

```
<form class="mt-3 needs-validation" novalidate>
  <div class="row mb-3">
    <label for="inputEmail3" class="col-sm-2 col-form-label">Email</label>
    <div class="col-sm-10">
      <input type="email" class="form-control" id="inputEmail3" required>
      <div class="valid-feedback">information valide</div>
      <div class="invalid-feedback">Le mail est obligatoire et doit être
valide</div>
    </div>
  </div>
  <div class="row mb-3">
    <label for="inputPassword3" class="col-sm-2 col-form-label">Password</label>
    <div class="col-sm-10 position-relative">
      <input type="password" class="form-control" id="inputPassword3" required>
      <div class="valid-tooltip">information valide</div>
      <div class="invalid-tooltip">Le mot de passe est obligatoire</div>
    </div>
  </div>
  <button type="submit" class="btn btn-primary">Sign in</button>
</form>
```

## La validation ► exemple ► script JS

```
// Example starter JavaScript for disabling form submissions if there are
// invalid fields
(() => {
  'use strict'
  // Fetch all the forms we want to apply custom Bootstrap validation
  styles to
  const forms = document.querySelectorAll('.needs-validation')
  // Loop over them and prevent submission
  Array.from(forms).forEach(form => {
    form.addEventListener('submit', event => {
      if (!form.checkValidity()) {
        event.preventDefault()
        event.stopPropagation()
      }
      form.classList.add('was-validated')
    }, false)
  })
})()
```

# La validation ► résultat

Email

Le mail est obligatoire et doit être valide

Password

Le mot de passe est obligatoire

Sign in

Infobulle

Incrusté

Exercices de l'activité B2 sur l'espace Moodle

## PAUSE ACTIVITÉ PRATIQUE