

COMPETENCES TRAVAILLEES

ANA	Décomposer un problème en sous-problèmes
REA	Faire preuve d'autonomie, d'initiative et de créativité
APP	Comprendre et réutiliser des codes sources existants

1. Etape 1 : un formulaire « type »

On dispose d'un exemple de formulaire type.

Ouvrir le fichier **formulaire.html** et la feuille de style **style.css** dans un éditeur de texte.

TRAVAIL DEMANDE

- 1) Analyser les fonctionnalités présentes et les associer au code HTML et CSS . Noter les observations et compléter le tableau.
- 2) Modifier l'attribut **method** de la balise **<form>** : passer de « *GET* » à « *POST* » et noter les changements observés. Compléter le tableau.
- 3) Que peut-on dire par rapport à la confidentialité des données saisies ?

<input> Champs de saisie				
Attribut type	<i>text</i>	<i>radio</i>		<i>date</i>
Décrire l'effet :			Case à cocher	
Attribut type	<i>submit</i>	<i>reset</i>		
Décrire l'effet :				
<label> </label>				
Décrire l'effet :				
<select> </select>				
Décrire l'effet :				
<form> </form> Zone de formulaire				
Attribut method	<i>GET</i>		<i>POST</i>	
Décrire l'effet :				

2. Concevoir son formulaire web

On va maintenant concevoir le formulaire adapté au projet de l'anniversaire de Pearl.
L'attribut **method** du formulaire sera réglé à « *POST* ».

TRAVAIL DEMANDE

A partir de l'exemple, concevoir le formulaire de renseignement en ligne en respectant ce qui est demandé dans le projet.

AIDES et RAPPELS :

- **class et id**

Les attributs *class* et *id* utilisables dans n'importe quelle balise sont importants pour le comportement de la balise et de son contenu, notamment les règles CSS.

→ Une classe particulière **class = "maclasse"** est utilisable dans plusieurs balises : elle est multiple et permet d'identifier un ensemble de balises. Les règles CSS lui sont données en utilisant la syntaxe :

`.maclasse{ /* mes règles CSS */ }`

→ Un identifiant particulier **id = "monid"** n'est utilisable que dans une seule balise : il est unique et permet d'identifier une et une seule balise. Les règles CSS lui sont données en utilisant la syntaxe :

`#monid{ /* Règles CSS */ }`

- **Sources utiles** www.w3schools.org/

POUR ALLER PLUS LOIN :

- **Framework**

Un framework est un ensemble complet de règles CSS standardisées et mises à jour. Cela permet une unité graphique et visuelle de la mise en forme rapide et facile à mettre en place. Un des plus utilisés est **Bootstrap** (<https://getbootstrap.com/>), et pour l'intégrer à la page, il faut ajouter au code le lien de style suivant :

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous"/>
```

- **Mise en forme par grilles (flex boxes)**

Pour un comportement *responsive* (ie qui s'adapte aux écrans variés tels que smartphones et tablettes), on utilisera plutôt des *flex boxes* plutôt que des tableaux, plus rigides.

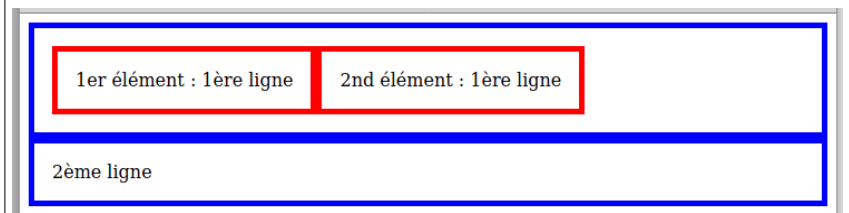
Exemple de code HTML :

```
<body>
  <div class="row">
    <div class="col">
      1er élément : 1ère ligne
    </div>
    <div class="col">
      2nd élément : 1ère ligne
    </div>
  </div>
  <div class="row">
    2ème ligne
  </div>
</body>
```

Exemple de feuille de style associée :

```
.row {
  display: flex;
  flex-wrap: row;
  padding: 1em;
  border: solid 5px blue;
}
.col {
  display: flex;
  flex-wrap: column;
  padding: 1em;
  border: solid 5px red;
}
```

Rendu web : (Firefox)



Voir https://www.w3schools.com/csS/css3_flexbox.asp