

Doc

Responivness

1. Architecture Générale

1.1 Structure HTML Standard

Structure hiérarchique :

Form

- └ presentation-grid (container grid)
- └ grid-field
 - └ grid-label
 - └ grid-input

1.2 Exemple HTML – Formulaire 4 lignes

```
<div class="presentation-grid" grid style="--rows: 4">
```

```
<div class="grid-field">  
  <div class="grid-label">  
    <Tsi-Label class="col-4"  
      [labelValue]="valeursGerees_opcvm_libelle">  
    </Tsi-Label>  
  </div>  
  <div class="grid-input">  
    <Tsi-Text-Box [(inputField)]="opcvm.libelle"></Tsi-Text-Box>  
  </div>  
</div>
```

```
<div class="grid-field">  
  <div class="grid-label">  
    <Tsi-Label class="col-4"  
      [labelValue]="valeursGerees_opcvm_capitalInitial">  
    </Tsi-Label>  
  </div>  
  <div class="grid-input">  
    <Tsi-Decimal-Input [(inputField)]="opcvm.capitalInitial"></Tsi-Decimal-Input>  
  </div>  
</div>
```

```
<div class="grid-field">
  <div class="grid-label">
    <Tsi-Label class="col-4"
      [labelValue]="valeursGerees_opcvm_dateAgreement">
    </Tsi-Label>
  </div>
  <div class="grid-input">
    <Tsi-Date-Picker [(inputField)]="opcvm.dateAgreement"></Tsi-Date-Picker>
  </div>
</div>

<div class="grid-field">
  <div class="grid-label">
    <Tsi-Label class="col-4"
      [labelValue]="valeursGerees_opcvm_deviser">
    </Tsi-Label>
  </div>
  <div class="grid-input">
    <Tsi-Search-Combo
      [(bind)]="opcvm.deviser"
      id-field="uid"
      label-field="libelle">
    </Tsi-Search-Combo>
  </div>
</div>

</div>
```

2. Système Grid

2.1 Container principal

Classe utilisée :

```
.simple-form
```

Rôle :

- Définit le conteneur principal du formulaire
- Gère l'espacement général
- Applique le style global du bloc

2.2 Configuration du Grid

Classe utilisée :

```
.presentation-grid
```

Cette classe active le système de mise en page basé sur **CSS Grid**.

Rôle :

- Une organisation automatique des champs
 - Une gestion dynamique du nombre de lignes
 - Une adaptation selon la taille d'écran
 - Une optimisation intelligente de l'espace
-

2.3 Explication des propriétés utilisées

2.3.a `display: grid`

Active le système CSS Grid pour organiser les champs sous forme de colonnes et lignes.

2.3.b Gestion dynamique des lignes

Le nombre de lignes est défini dynamiquement via une variable CSS :

```
style="-- rows: 4"
```

Rôle :

- D'adapter facilement la structure
 - D'éviter de modifier le CSS principal
 - De rendre le système flexible et réutilisable
-

2.3.c Flux automatique optimisé

Le système :

- Remplit verticalement les lignes
- Passe automatiquement à la colonne suivante
- Optimise les espaces vides

Cela garantit une distribution intelligente des champs.

2.3.d Colonnes adaptatives

Le système :

- Définit une largeur minimale pour éviter l'écrasement

- Permet aux colonnes de s'étendre selon l'espace disponible
 - Maintient une mise en page équilibrée
-

2.3.e Espacement uniforme

Un espacement homogène est appliqué :

- Horizontalement
- Verticalement

Cela améliore la lisibilité et l'esthétique.

3. Structure d'un Champ

Classes utilisées :

```
.grid-field : Conteneur vertical d'un champ (Label + Input)
.grid-label : Affichage du titre du champ
.grid-input : Contient le composant (TextBox, DatePicker, Combo, etc.)
```

Rôle :

- Une cohérence visuelle
 - Une maintenabilité
 - Une standardisation des formulaires
-

4. Responsiveness

Le comportement change automatiquement selon la largeur d'écran.

4.1 Mobile

Comportement :

- Une seule colonne
 - Champs empilés verticalement
 - Lecture optimisée pour petit écran
-

4.2 Tablet

Comportement :

- Deux colonnes
 - Distribution équilibrée
 - Bon compromis entre lisibilité et densité
-

4.3 Desktop

Comportement :

- Trois colonnes
 - Optimisation de l'espace large écran
 - Mise en page professionnelle
-

5. Extension Standard

Champ pleine largeur

Classe utilisée :

```
.grid-field.full-width
```

Rôle :

- Le champ occupe toutes les colonnes disponibles
 - Idéal pour :
 - Textarea
 - Notes
 - Champs importants
-

6. Résumé Technique

- ✓ Nombre de lignes dynamique via variable CSS
- ✓ Colonnes adaptatives selon largeur écran

7. Exemple idéal pour le suivi dans le code HTML

7.1 Module « Valeurs Gérées » – Écran OPCVM (OPCVMWeb)

- Cas d'un écran contenant un nombre important de champs (environ 29 à 30).
-

7.2 Module « Paramétrages » – Écran Année (OPCVMWeb)

- Cas d'un écran contenant un nombre réduit de champs (environ 3 à 4).
-

Revision #6

Created 2026-02-11 08:12:05 UTC by Oussema Mhamdi

Updated 2026-02-17 09:36:53 UTC by Oussema Mhamdi