



DÉVELOPPEMENT D'APPLICATIONS WEB

COURS 4: MISE EN PAGE CSS AVEC GRID

Enseignante: NGUYEN Thi Minh Tuyen



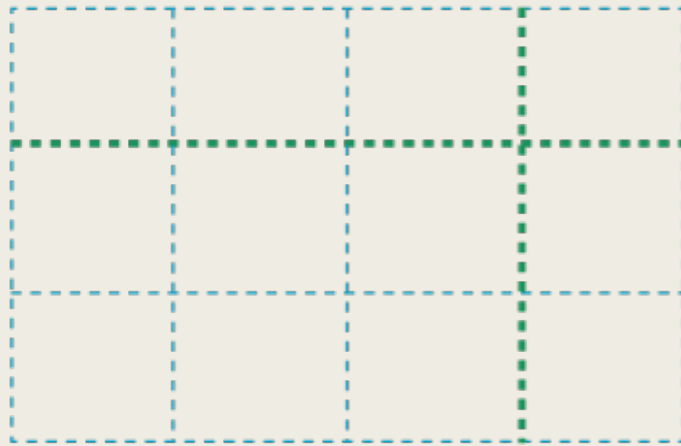
Plan du cours

1. Terminologie de la grille
2. Type d'affichage de grille
3. Création du modèle de grille
4. Nommage des zones de la grille
5. Placement des éléments de la grille
6. Comportement implicite de la grille
7. Espacement et alignement de la grille

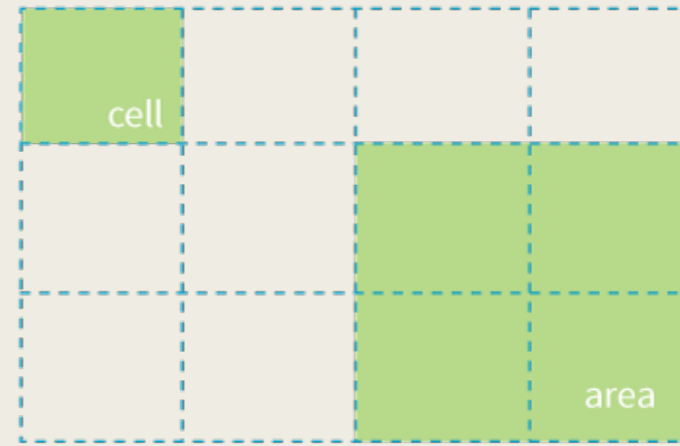
Comment fonctionnent les grilles CSS

1. Définit l'affichage d'un élément au mode `grid` pour établir un conteneur de grille. Ses enfants deviennent des éléments de la grille.
2. Configure les colonnes et les lignes de la grille (explicitement ou avec des règles pour leur création à la volée).
3. Attribue chaque élément de la grille à une zone de la grille (ou laisse-les défiler automatiquement dans un ordre séquentiel).

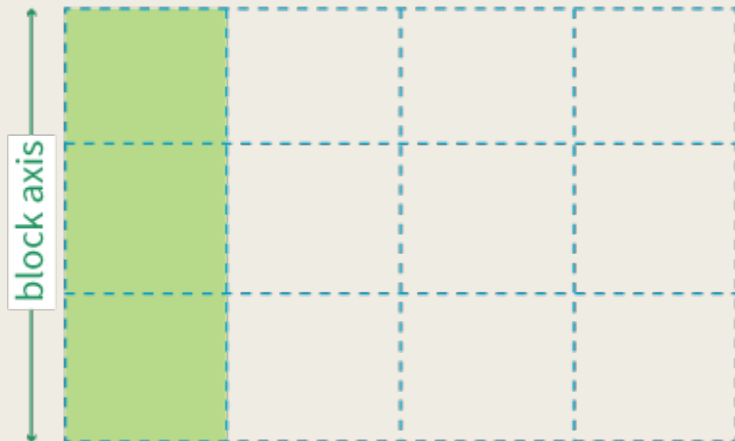
Terminologie de la grille



grid lines



grid cell and grid area



grid track (column)



grid track (row)

Créer un conteneur de grille

- Pour transformer un élément en un conteneur de grille, définissez sa propriété `display` avec la valeur `grid`.
- Tous ses enfants deviennent automatiquement des éléments de la grille.

- HTML:

```
<div id="layout">  
  <div id="one">One</div>  
  <div id="two">Two</div>  
  <div id="three">Three</div>  
  <div id="four">Four</div>  
  <div id="five">Five</div>  
</div>
```

- CSS:

```
#layout {  
  display: grid;  
}
```

Définition des pistes de lignes et de colonnes

`grid-template-rows, grid-template-columns`

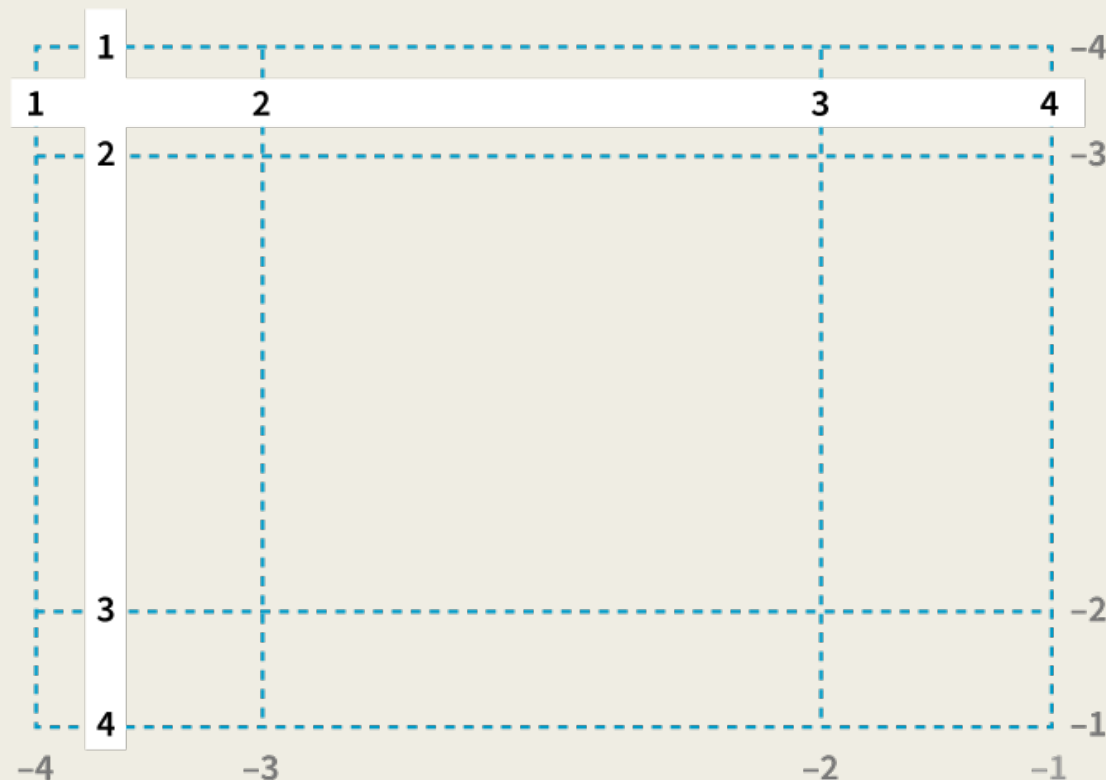
- Valeurs: none, liste des tailles de piste et noms de lignes optionnel
- La valeur de `grid-template-rows` (`grid-template-columns`) est une liste des hauteurs (largeur) de chaque piste de ligne (de colonnes) dans la grille.

```
#layout {  
    display: grid;  
    grid-template-rows: 100px 400px 100px;  
    grid-template-columns: 200px 500px 200px;  
}
```

- Le nombre de tailles fournies détermine le nombre de lignes / colonnes dans la grille. Dans l'exemple ci-dessus, cette grille comporte 3 lignes et 3 colonnes.

Numéros de lignes de grille

- Les navigateurs attribuent automatiquement un numéro à chaque ligne de la grille, en commençant par 1 au début de chaque piste et par chaque piste de colonne et en commençant par -1 à la fin.



Noms de lignes de grille

- Vous pouvez également attribuer des noms aux lignes pour les rendre plus intuitives à référencer ultérieurement.
- Les noms de lignes de grille sont ajoutés entre crochets dans la position où ils apparaissent par rapport aux pistes.
- Pour attribuer plusieurs noms à une ligne, inclue tous les noms entre parenthèses, séparés par des espaces.

```
#layout {  
  display: grid;  
  grid-template-rows: [header-start] 100px [header-end  
content-start] 400px [content-end footer-start] 100px;  
  grid-template-columns: [ads] 200px [main] 500px [links]  
200px;  
}
```


Suivre les valeurs de taille

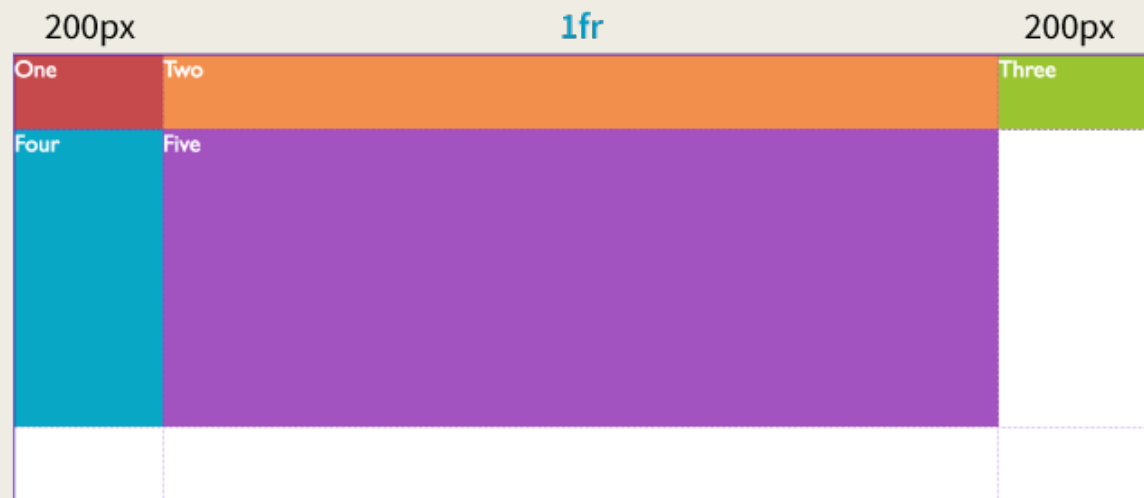
La spécification CSS Grid fournit de nombreuses façons pour spécifier la largeur et la hauteur d'une piste. Certaines de ces méthodes permettent aux pistes de s'adapter à l'espace disponible et/ou au contenu qu'elles contiennent:

- Longueurs (telles que px ou em)
- Valeurs en pourcentage (%)
- Unités fractionnelles
- `minmax()`
- `min-content`, `max-content`
- `auto`
- `fit-content()`

Unités fractionnées (fr)

- Représente une fraction de l'espace restant dans le conteneur en grille

```
#layout {  
    display: grid;  
    grid-template-rows: 100px 400px 100px;  
    grid-template-columns: 200px 1fr 200px;  
}
```



Gamme de taille avec `minmax()`

- Limite la taille de la piste en définissant une dimension minimale et maximale.
- Il est utilisé à la place d'une taille de piste spécifique.
- Cette règle définit la colonne du milieu à au moins 15em mais jamais à plus de 45em:

```
grid-template-columns: 200px minmax(15em, 45em)  
200px;
```

min-content et max-content

- min-content est le plus petit qu'une piste puisse être.
- max-content alloue la quantité maximale d'espace nécessaire.
- auto laisse le navigateur s'en charger. Commence avec auto pour le dimensionnement basé sur le contenu.

Text content in cell

Look for the good in
others and they'll see
the good in you.

Column width set to
max-content

Look for the good in others and they'll see the good in you.

Column width set to
min-content

Look
for
the
good
in
others
and
they'll
see
the
good
in
you.

Répétition des tailles de piste [1]

- La fonction de raccourci `repeat()` vous permet de répéter des motifs dans les tailles de piste:

`repeat(#, track pattern)`

- Le premier nombre est le nombre de répétitions. Les tailles de piste après la virgule fournissent le motif:
- AVANT: `grid-template-columns: 200px 20px 1fr 20px 1fr 20px 1fr 20px 1fr 20px 1fr 200px;`
- AFTER: `grid-template-columns: 200px repeat(5, 20px 1fr) 200px;`

Répétition des tailles de piste [2]

- Vous pouvez laisser le navigateur déterminer le nombre de fois qu'un motif répété s'adapte avec `auto-fill` et `auto-fit` au lieu d'un nombre:

`grid-template-rows: repeat(auto-fill, 15em);`

- `auto-fill` crée autant de pistes qu'elle tiendra dans l'espace disponible, même s'il n'y a pas assez de contenu pour remplir toutes les pistes.
- `auto-fit` crée autant de pistes que nécessaire, en supprimant les pistes vides de la mise en page.

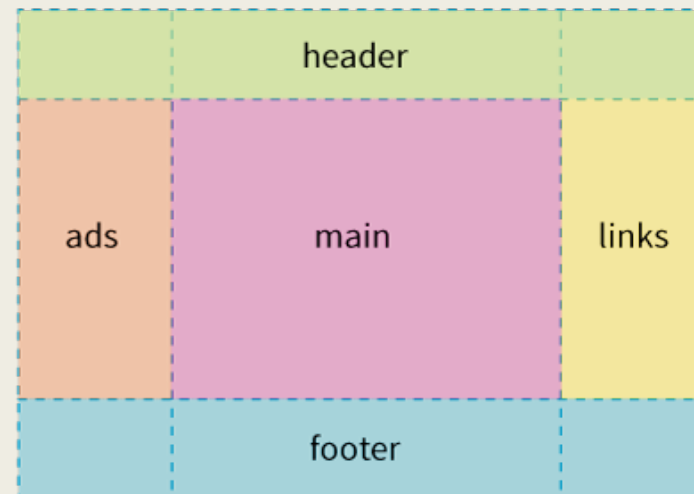
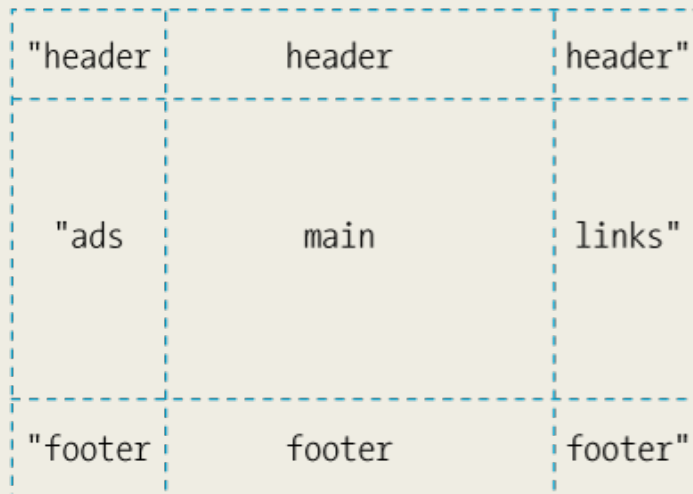
Donner des noms aux zones de la grille [1]

`grid-template-areas`

- Valeurs: none, série de noms de zone par ligne
- Permet d'attribuer des noms aux zones de la grille pour faciliter le placement d'éléments dans cette zone ultérieurement.
- La valeur est une liste de noms pour chaque cellule de la grille, listée par ligne.
- Lorsque les cellules voisines partagent un nom, elles forment une zone de grille portant ce nom.

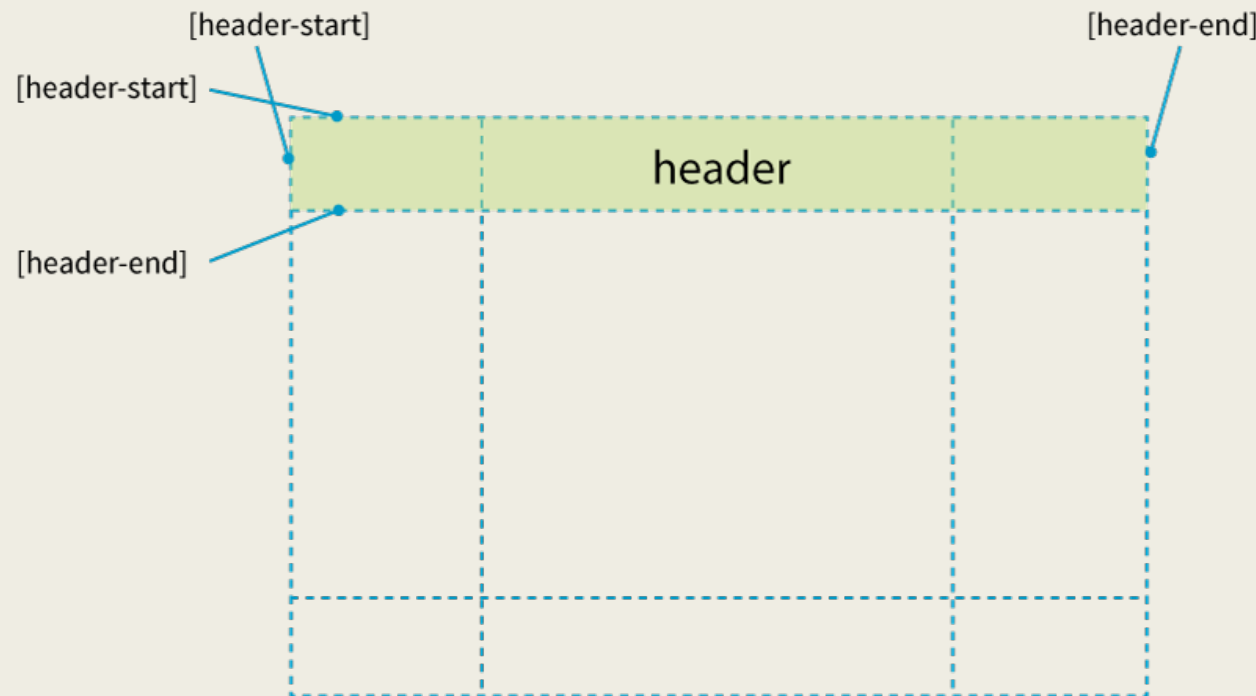
Donner des noms aux zones de la grille [2]

```
#layout {  
  display: grid;  
  grid-template-rows: [header-start] 100px [content-start] 400px  
  [footer-start] 100px;  
  grid-template-columns: [ads] 200px [main] 1fr [links] 200px;  
  grid-template-areas:  
    "header header header"  
    "ads main links"  
    "footer footer footer"  
}
```



Donner des noms aux zones de la grille [3]

- L'attribution de noms à des lignes avec des suffixes `-start` et `-end` crée implicitement un nom de zone.



De même, lorsque vous spécifiez un nom de zone avec `grid-template-areas`, les noms de ligne avec les suffixes `-start` et `-end` sont générés de manière implicite.

Propriété raccourci: `grid` [1]

`grid`

- Valeurs: `none`, info de ligne/colonne.
- Le raccourci `grid` définit les valeurs pour `grid-template-rows`, `grid-template-columns`, et `grid-template-areas`.

Propriété raccourci: `grid` [2]

- Placez les valeurs liées aux lignes avant la barre oblique (/) et les valeurs liées aux colonnes après:

`grid: rows / columns`

- Exemple:

```
#layout {  
    display: grid;  
    grid: 100px 400px 100px / 200px 1fr 200px;  
}
```

Propriété raccourci: `grid` [3]

- Vous pouvez également inclure des noms de ligne et des noms de zone, dans cet ordre:

```
[start line name] "area names" <track size> [end  
line name]
```

- Exemple:

```
#layout {  
  display: grid;  
  grid:  
    [header-start] "header header header" 100px  
    [content-start] "ads main links" 400px  
    [footer-start] "footer footer footer" 100px  
    / [ads] 200px [main] 1fr [links] 200px;  
}
```

Placement d'éléments à l'aide de lignes de grille

`grid-row-start`

`grid-row-end`

`grid-column-start`

`grid-column-end`

- Valeurs: auto, "line name", span number, span "line name", number "line name"
- Ces propriétés positionnent les éléments de la grille sur la grille en référençant les lignes de la grille où ils commencent et se terminent.
- La propriété est appliquée à l'élément d'élément de grille.

Mise en place d'objets sur la grille [1]

- Par numéro de ligne:

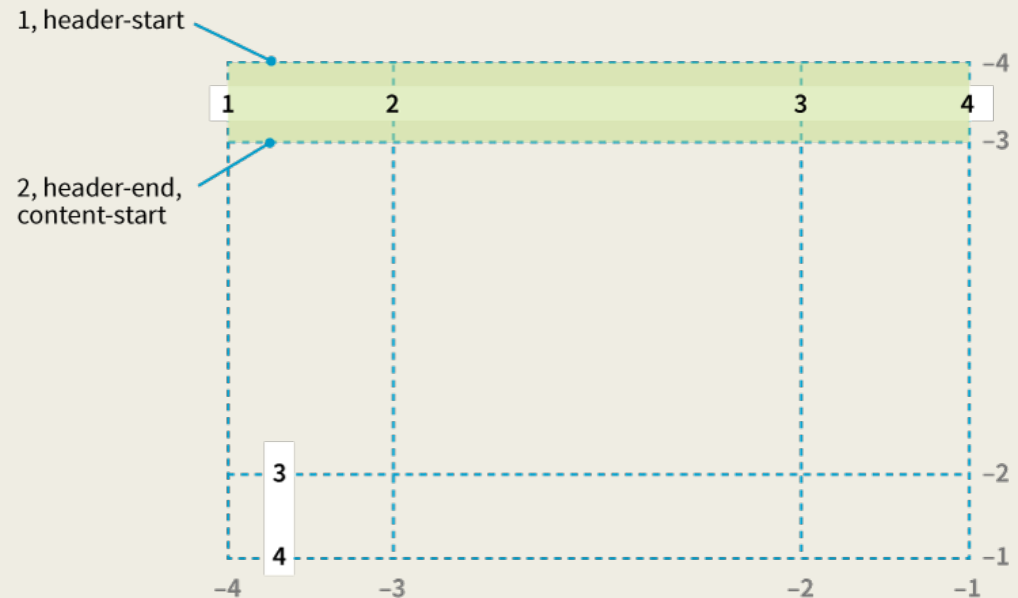
```
#one {  
  grid-row-start: 1;  
  grid-row-end: 2;  
  grid-column-start: 1;  
  grid-column-end: 4;  
}
```

- En utilisant un span:

```
#one {  
  ...  
  grid-column-start: 1;  
  grid-column-end: span 3;  
}
```

- À partir de la dernière ligne de la grille et en arrière:

```
#one {  
  ...  
  grid-column-start: span 3;  
  grid-column-end: -1;  
}
```



- Par nom de ligne:

```
#one {  
  grid-row-start: header-start;  
  grid-row-end: header-end;  
  ...  
}
```

Mise en place d'objets sur la grille [2]

`grid-row`

`grid-column`

- Valeurs: "start line" / "end line"
- Ces propriétés abrégées combinent les propriétés `*-start` et `*-end` en une seule déclaration. Les valeurs sont séparées par une barre oblique (/):

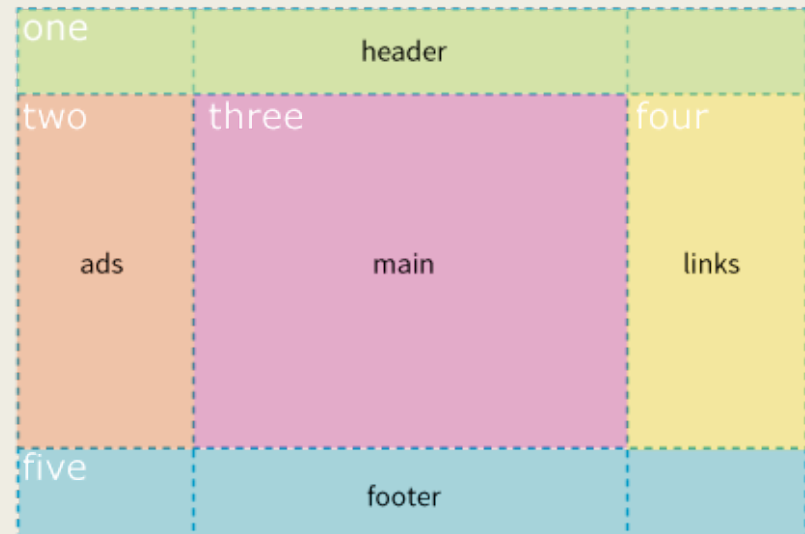
```
#one {  
    grid-row: 1 / 2;  
    grid-column: 1 / span 3;  
}
```

Placer des éléments sur la grille à l'aide de zones

grid-area

- Valeurs: nom de la zone, 1 à 4 identificateurs de ligne
- Positionne un élément dans une zone créée avec grid-template-areas:

```
#one { grid-area: header; }  
#two { grid-area: ads; }  
#three { grid-area: main; }  
#four { grid-area: links; }  
#five { grid-area: footer; }
```



Comportement implicite de la grille

Le système de disposition de la grille fait automatiquement certaines choses pour vous (comportement implicite):

- Génération de noms de lignes «-start» et «-end» lorsque vous nommez une zone (et inversement)
- Insertion séquentielle d'éléments dans les cellules de la grille si vous ne les placez pas explicitement
- Ajouter des lignes et des colonnes à la volée au besoin pour ajuster les éléments

Pistes générées automatiquement

`grid-auto-rows`
`grid-auto-columns`

- Valeurs: Liste des tailles de piste
- Fournit une ou plusieurs tailles de piste pour les pistes automatisées. Si vous fournissez plus d'une valeur, cela agit comme un motif répétitif.
- Exemple:
`grid-template-columns: repeat(3, 1fr);`
`grid-auto-rows: 200px;`

Direction de flux et densité [1]

`grid-auto-flow`

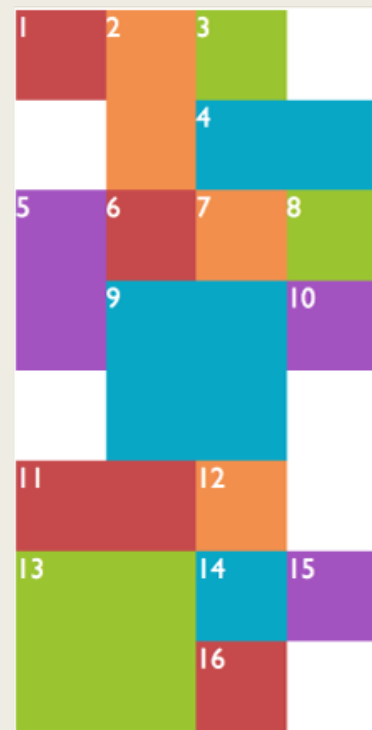
- Valeurs: `row` ou `column`, `dense` (optionnel)
- Spécifie si vous souhaitez que les éléments entrent par `row` ou par `column`. La valeur par défaut est le sens d'écriture du document.
- Exemple:

```
#listings {  
    display: grid;  
    grid-auto-flow: column dense;  
}
```

Direction de flux et densité [2]

- Le mot-clé **dense** indique au navigateur de remplir la grille de la manière la plus dense possible, ce qui permet aux éléments de s'afficher dans le désordre.

Default flow pattern



Dense flow pattern



Propriété `grid` (revisit )

- Utilise le mot cl  `auto-flow` dans la propri t  raccourci `grid` pour indiquer que les lignes ou les colonnes doivent  tre g n r es automatiquement.

- Exemple:

```
grid: auto-flow 200px / repeat(3, 1fr);
```

- Comme `auto-flow` est inclus avec les informations sur les lignes, `grid-auto-flow` est d fini sur `row`.

Espacement entre les pistes [1]

`grid-row-gap`, `grid-column-gap`

- Valeurs: Longueur (ne doit pas être négatif)

`grid-gap`

- Valeurs: `grid-row-gap` `grid-column-gap`
- Ajoute de l'espace entre les pistes et les pistes de la grille
- **Remarque:** ces noms de propriété seront remplacés par `row-gap`, `column-gap` et `gap`, mais les nouveaux noms ne sont pas encore pris en charge.

Espacement entre les pistes [2]

- Si vous voulez un espace égal entre toutes les pistes d'une grille, utilisez un espace au lieu de créer des pistes d'espacement supplémentaires:

grid-gap: 20px 50px;

(Ajoute un espace de 20 px entre les lignes et de 50 px entre les colonnes)



Alignement d'élément [1]

`justify-self`

`align-self`

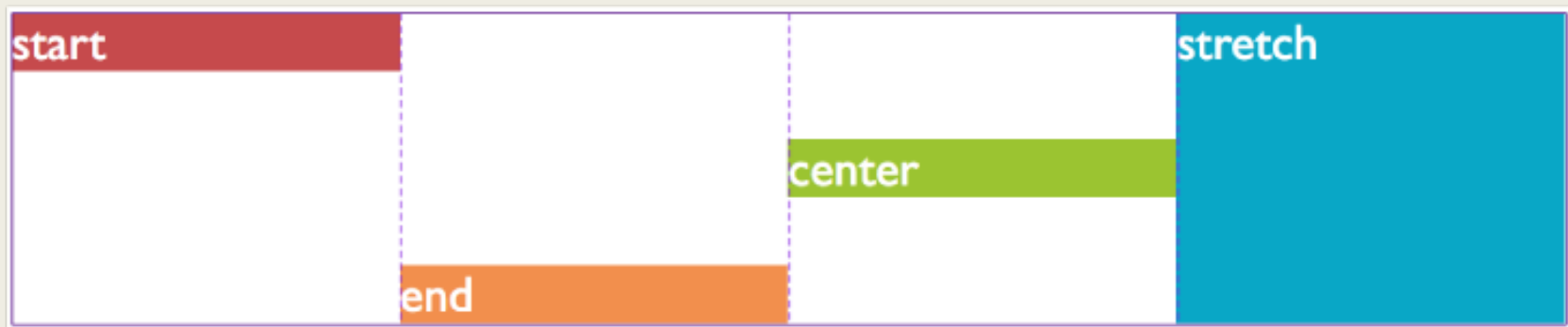
- Valeurs: `start`, `end`, `center`, `left`, `right`, `self-start`, `self-end`, `stretch`, `normal`, `auto`
- Spécifie comment un élément doit être aligné dans la cellule.
- `justifier-self` s'aligne sur l'axe en ligne (horizontalement de gauche à droite).
- `align-self` s'aligne sur l'axe (vertical) du bloc.

Alignement d'élément [2]

justify-self



align-self



Aligner tous les éléments

`justify-items`

`align-items`

- Valeurs: `start`, `end`, `center`, `left`, `right`, `self-start`, `self-end`, `stretch`, `normal`, `auto`
- Alignent les éléments de leurs cellules en une fois.
- Appliqués au conteneur de grille.
- `justify-items` s'aligne sur l'axe en ligne.
- `align-items` s'aligne sur l'axe (vertical) du bloc.

Alignement des pistes [1]

`justify-content`

`align-content`

- Values: `start`, `end`, `center`, `left`, `right`, `stretch`, `space-around`, `space-between`, `space-evenly`
- Lorsque les pistes de la grille ne remplissent pas tout le conteneur, vous pouvez spécifier l'alignement des pistes.
- `justify-content` s'aligne sur l'axe en ligne (horizontalement de gauche à droite).
- `align-content` s'aligne sur l'axe (vertical) du bloc.

Alignement des pistes [2]

justify-content:
start

One	Two	Three	Four	
Five	Six	Seven	Eight	
Nine	Ten	Eleven	Twelve	

end

	One	Two	Three	Four
	Five	Six	Seven	Eight
	Nine	Ten	Eleven	Twelve

center

	One	Two	Three	Four
	Five	Six	Seven	Eight
	Nine	Ten	Eleven	Twelve

space-around

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

space-between

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

space-evenly

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

align-content:
start

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

end

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

center

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

space-around

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

space-between

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

space-evenly

One	Two	Three	Four
Five	Six	Seven	Eight
Nine	Ten	Eleven	Twelve

Revue des propriétés de grid

Propriétés du conteneur de grille

`display: grid | inline-grid`
`grid`
 `grid-template`
 `grid-template-rows`
 `grid-template-columns`
 `grid-template-areas`
`grid-auto-rows`
`grid-auto-columns`
`grid-auto-flow`
`grid-gap`
 `grid-row-gap`
 `grid-column-gap`
`justify-items`
`align-items`
`justify-content`
`align-content`

Propriétés de l'élément de grille

`grid-column`
 `grid-column-start`
 `grid-column-end`
`grid-row`
 `grid-row-start`
 `grid-row-end`
`grid-area`
`justify-self`
`align-self`
`order` (ne fait pas partie du module de grille)
`z-index` (ne fait pas partie du module de grille)