

Création de listes

Les listes sont une partie de la vie quotidienne. Les "To-do" listes déterminent ce qui doit être fait. Les recettes fournissent des listes d'ingrédients et de listes d'instructions. Avec une liste on peut décrire presque tout, il est facile de comprendre pourquoi ils sont aussi très populaires en ligne.

Lorsque nous voulons utiliser une liste sur un site Web, HTML propose trois types différents de choix: listes non ordonnées, commandés et de description. Choisir le type de liste à utiliser - ou utiliser une liste - se résume au contenu et à l'option la plus sémantiquement appropriée pour afficher ce contenu.

En plus des trois types différents de listes disponibles dans HTML, il existe plusieurs façons de styler ces listes avec CSS. Par exemple, nous pouvons choisir quel type de marqueur utiliser sur une liste. Le marqueur pourrait être carré, rond, numérique, alphabétique, ou peut-être inexistant. De plus, nous pouvons décider si une liste doit être affichée verticalement ou horizontalement. Tous ces choix jouent des rôles importants dans le style de nos pages Web..

Listes non ordonnées

Une liste non ordonnée est simplement une liste d'articles connexes dont l'ordre n'a pas d'importance. La création d'une liste non ordonnée en HTML s'effectue à l'aide de l'élément de niveau bloc de liste non ordonnée, ``. Chaque élément d'une liste non ordonnée est marqué individuellement à l'aide de l'élément élément de liste, ``.

Par défaut, la plupart des navigateurs ajoutent une marge et un padding verticale à gauche à l'élément `` et précèdent chaque `` élément avec un point. Ce point est appelé le marqueur d'élément de liste, et il peut être modifié en utilisant CSS.

```
1 <ul>
2   <li>Orange</li>
3   <li>Vert</li>
4   <li>Blue</li>
5 </ul>
```

- Orange
- Green
- Blue

Demo de listes non ordonnée

Liste Ordonnée

L'élément de liste ordonnée, ``, fonctionne comme l'élément de liste non ordonnée. Les éléments de liste individuels sont créés de la même manière. La principale différence entre une liste ordonnée et une liste non ordonnée est qu'avec une liste ordonnée, l'ordre dans lequel les éléments sont présentés est important.

Comme l'ordre compte, au lieu d'utiliser un point comme marqueur d'élément de liste par défaut, une liste ordonnée utilise des nombres.

```
1 <ol>
2   <li>Emincer les oignons</li>
3   <li>Faites revenir les oignons</li>
4   <li>Ajouter le boeuf</li>
5 </ol>
```

Les listes commandées ont également des attributs uniques disponibles pour eux, `start` et `reverse`.

1. Emincer les oignons
2. Faites revenir les oignons
3. Ajouter le boeuf

Démo de listes ordonnées

Démarrer

L'attribut `start` définit le numéro à partir duquel une liste ordonnée doit démarrer. Par défaut, les listes ordonnées commencent à 1. Toutefois, il peut y avoir des cas où une liste doit commencer à 30 ou un autre numéro. Lorsque nous utilisons l'attribut `start` sur l'élément ``, nous pouvons identifier exactement le numéro dont une liste ordonnée doit commencer le comptage.

L'attribut `start` accepte uniquement des valeurs entières, même si les listes ordonnées peuvent utiliser des systèmes de numérotation différents, tels que des numéros romains.

```
1 <ol start="30">
2   <li>Emincer les oignons</li>
3   <li>Faites revenir les oignons</li>
4   <li>Ajouter le boeuf</li>
5 </ol>
```

```
30. Emincer les oignons
31. Faites revenir les oignons
32. Ajouter le boeuf
```

Démo attribut start

Reverse

L'attribut `reversed`, lorsqu'il est utilisé sur l'élément ``, permet à une liste d'apparaître dans l'ordre inverse. Une liste ordonnée de cinq éléments numérotés de 1 à 5 peut être inversée et ordonnée de 5 à 1. L'attribut `reversed` est un attribut booléen et, en tant que tel, il n'accepte aucune valeur. Il est vrai ou faux. Faux est la valeur par défaut; La valeur devient vraie lorsque le nom d'attribut `reversed` apparaît sur l'élément ``.

```
1 <ol reversed>
2   <li>Emincer les oignons</li>
3   <li>Faites revenir les oignons</li>
4   <li>Ajouter le boeuf</li>
5 </ol>
```

```
3. Emincer les oignons
2. Faites revenir les oignons
1. Ajouter le boeuf
```

Démo attribut reversed

L'attribut value

L'attribut `value` peut être utilisé sur un élément `` individuel dans une liste ordonnée pour modifier sa valeur dans la liste. Le numéro de tout élément de liste apparaissant sous un élément de liste avec un attribut de valeur sera recalculé en conséquence.

Par exemple, si le deuxième élément de liste a une valeur d'attribut de 9, le numéro sur ce marqueur d'élément de liste apparaîtra comme s'il s'agissait du neuvième élément. Tous les articles suivants seront numérotés à partir de 9.

```
1 <ol>
2   <li>Emincer les oignons</ li>
3   <li value ="9">Faites revenir les oignons</ li>
4   <li>Ajouter le boeuf</ li>
5 </ol>
```

```
1. Emincer les oignons
9. Faites revenir les oignons
10. Ajouter le boeuf
```

Demo attribut Value

Listes de description

Un autre type de liste vu en ligne (mais pas aussi souvent que les listes non ordonnées ou ordonnées) est la liste de description. Les listes de description sont utilisées pour décrire plusieurs termes et leurs descriptions, par exemple dans un glossaire.

La création d'une liste de description en HTML s'effectue à l'aide de l'élément de niveau de bloc de liste de description, `<dl>`. Au lieu d'utiliser un élément `` pour marquer les éléments de liste, la liste de description nécessite deux éléments au niveau du bloc: l'élément de terme de description, `<dt>` et l'élément de description, `<dd>`.

Une liste de description peut contenir de nombreux termes et descriptions, l'un après l'autre. En outre, une liste de description peut avoir plusieurs termes par description, ainsi que des descriptions multiples par terme. Un terme unique peut avoir plusieurs significations et justifier de multiples descriptions. Inversement, une description unique peut convenir à plusieurs termes.

Lors de l'ajout d'une liste de description, l'élément `<dt>` doit venir avant l'élément `<dd>`. Le terme de définition et la description qui le suit directement correspondent les uns aux autres; Ainsi, l'ordre de ces éléments est important.

Par défaut, l'élément `<dl>` inclura des marges verticales, tout comme les éléments `` et ``. En outre, l'élément `<dd>` inclut une marge gauche par défaut.

HTML

```
1 <dl>
2   <dt>Etude</dt>
3   <dd>Temps et l'attention nécessaires à l'acquisition des
4   connaissances sur un sujet académique,en particulier par
5   le biais de livres</dd>
6   <dt>Conception</dt>
7   <dd>Un plan ou dessin réalisé pour montrer l'apparence
8   et fonction ou fonctionnement d'un bâtiment, vêtement, ou
9   autre objet avant qu'il ne soit construit ou fait</dd>
10  <dd>But, planification ou intention qui existe ou est
11  pensée pour exister derrière une action, fait ou objet
12  matériel</dd>
13  <dt>Commerce</dt>
14  <dt>Travail</dt>
15  <dd>Occupation régulière d'une personne,profession ou
16  commerce</dd>
17 </dl>
```

CSS

```
1 dd{
2   margin-left: 40px;
3 }
```

Etude	Temps et l'attention nécessaires à l'acquisition des connaissances sur un sujet académique,en particulier par le biais de livres
Conception	Un plan ou dessin réalisé pour montrer l'apparence et fonction ou fonctionnement d'un bâtiment, vêtement, ou autre objet avant qu'il ne soit construit ou fait
	But, planification ou intention qui existe ou est pensée pour exister derrière une action, fait ou objet matériel
Commerce	
Travail	Occupation régulière d'une personne,profession ou commerce

Démo de listes de description

Listes imbriquées

Une caractéristique qui rend les listes extrêmement puissantes est leur capacité à être imbriquées. Chaque liste peut être placée dans une autre liste. Elles peuvent être imbriquées continuellement. Mais le potentiel de nicher des listes indéfiniment ne veut pas dire qu'il faut le faire. Les listes doivent encore être réservées spécifiquement là où elles contiennent la plus grande valeur sémantique.

Une astuce avec les listes de nidification est de savoir par où commencer et terminer chaque liste et élément de liste. En parlant spécifiquement des listes non ordonnées et ordonnées, car c'est là que la plupart des niches se produiront, le seul élément qui peut résider directement dans les éléments `` et `` est l'élément ``. Pour répéter, le seul élément que nous pouvons placer comme un enfant direct des éléments `` et `` est l'élément ``.

Cela dit, une fois à l'intérieur de l'élément ``, l'ensemble des éléments standard peut être ajouté, y compris tout élément `` ou ``.

Pour imbriquer une liste plutôt que de fermer un élément de liste, commencez une nouvelle liste. Une fois la liste imbriquée terminée et fermée, fermer l'élément de la liste qui l'entoure et continuez avec la liste d'origine..

HTML

```
1 <ol>
2   <li>Promener le chien</li>
3   <li>Passer à la blanchisserie</li>
4   <li>
5     Aller à l'épicerie et acheter:
6     <ul>
7       <li>Lait</li>
8       <li>Pain</li>
9       <li>Fromage</li>
10    </ul>
11  </li>
12  <li>Tondre la pelouse</li>
13  <li>Faire le dîner</li>
14 </ol>
```

CSS

```
1 ul{
2   list-style: disc;
3   padding-left: 40px;
4 }
5 ol{
6   list-style: decimal;
7   padding-left: 40px;
8 }
```

1. Promener le chien
2. Passer à la blanchisserie
3. Aller à l'épicerie et acheter:
 - Lait
 - Pain
 - Fromage
4. Tondre la pelouse
5. Faire le dîner

Démos de listes imbriquées

Parce que les listes imbriquées peuvent être un peu délicates - et les styles indésirables peuvent apparaître si cela est mal fait. Les éléments `` et `` peuvent contenir uniquement des éléments ``. L'élément `` peut contenir n'importe quel élément normal comme souhaité; Cependant, l'élément `` doit être un enfant direct d'un élément `` ou ``.

Il est également intéressant de noter que les listes étant imbriquées à l'intérieur d'autres listes, leurs marqueurs d'élément de liste changent en fonction de la profondeur de la liste imbriquée. Dans l'exemple précédent, la liste non ordonnée imbriquée dans la liste ordonnée utilise des cercles creux au lieu de disques solides comme marqueur d'élément de liste. Ce changement se produit car la liste non ordonnée est imbriquée d'un niveau dans la liste ordonnée.

Heureusement, nous avons le contrôle sur la façon dont ces marqueurs d'élément de liste apparaissent à n'importe quel niveau.

Styler les éléments de liste

Les listes non ordonnées et ordonnées utilisent des marqueurs d'élément de liste par défaut. Pour les listes non ordonnées, il s'agit généralement de points, tandis que les listes ordonnées utilisent généralement des nombres. Avec CSS, le style et la position de ces marqueurs d'élément de liste peuvent être ajustés.

Propriété type de style de liste `list-style-type`

La propriété `list-style-type` est utilisée pour définir le contenu d'un marqueur d'élément de liste. Les [valeurs disponibles](#) vont des carrés, des nombres décimaux à la numérotation arménienne, et le style peut être placé sur les éléments ``, `` ou `` dans CSS.

Toute valeur de propriété `list-style-type` peut être ajoutée à des listes non ordonnées ou ordonnées. Dans cet esprit, il est possible d'utiliser un marqueur d'élément de liste numérique sur une liste non ordonnée et un marqueur non numérique sur une liste ordonnée.

HTML

```
1 <ul>
2   <li>Orange</li>
3   <li>Vert</li>
4   <li>Bleu</li>
5 </ ul>
```

CSS

```
1 ul {
2   list-style-type: square;
3 }
```

- Orange
- Vert
- Bleu

Démo de `list-style-type`

Valeurs de list-style-type

Comme mentionné précédemment, la propriété list-style-type est livrée avec une poignée de valeurs différentes. La liste suivante décrit ces valeurs ainsi que leur contenu correspondant.

Valeurs de list-style-type	Contenu
none	Aucun élément de liste
disc	Un cercle rempli
circle	Un cercle creux
square	Un carré
decimal	Nombres décimaux
decimal-leading-zero	Nombres décimaux avec zéros initiaux
lower-roman	Chiffres romains minuscules
upper-roman	Chiffres romains majuscules
lower-greek	Minuscules grec classique
lower-alpha / lower-latin	Lettres ASCII minuscules
upper-alpha / upper-latin	Lettres ASCII majuscules
armenian	Pagination traditionnelle arménienne
georgian	Pagination géorgienne traditionnelle

Utilisation d'une image comme marqueur d'élément de liste

Il peut arriver que les valeurs par défaut des propriétés de type liste-style ne suffisent pas et que nous voulions personnaliser notre propre marqueur d'élément de liste. Pour ce faire, il est généralement possible de placer une image de fond sur chaque élément `` dans une liste.

Le processus inclut l'élimination de toute valeur de propriété de type liste-style par défaut et l'ajout d'une image d'arrière-plan et de padding à l'élément ``.

En détail, la valeur `none` pour la propriété `list-style-type` supprimera les marqueurs d'élément de liste existants. La propriété d'arrière-plan identifie une image d'arrière-plan, ainsi que sa position et sa valeur de répétition, si nécessaire. Et la propriété `padding` fournira un espace à gauche du texte pour l'image d'arrière-plan.

HTML

```
1 <ul>
2   <li>Orange</li>
3   <li>Vert</li>
4   <li>Bleu</li>
5 </ul>
```

CSS

```
1 li {
2   background: url("arrow.png") 0 50% no-repeat;
3   list-style-type: none;
4   padding-left: 12px;
5 }
```

- > Orange
- > Vert
- > Bleu

Demo Liste Image

Style de liste Propriété de position

Par défaut, le marqueur de l'élément de liste se trouve à gauche du contenu de l'élément ``. Ce positionnement de style de liste est décrit comme extérieur, ce qui signifie que tout le contenu apparaîtra directement à droite, en dehors du marqueur d'élément de liste. En utilisant la propriété `list-style-position`, nous pouvons changer la valeur par défaut à `outside`, `inside` ou `inherit`.

La valeur de la propriété extérieure place le marqueur de l'élément de liste à gauche de l'élément `` et ne permet pas à un contenu d'envelopper au-dessous du marqueur de l'élément de liste. La valeur de propriété intérieure (qui est rarement vue ou utilisée) place le marqueur de l'élément de liste en ligne avec la première ligne de l'élément `` et permet à d'autres contenus d'envelopper au-dessous, selon les besoins.

HTML

```
1 <ul>
2   <li>Croissants ...</li>
3   <li>Pain au chocolat ...</li>
4   <li>Pain aux raisins...</li>
</ul>
```

CSS

```
1 ul{
2   list-style-position: inside;
3 }
```

- Croissants ...
- Pain au chocolat ...
- Pain aux raisins...

List-style-position Propriété Demo

Propriété style de liste abrégée

Les propriétés de style de liste vue jusqu'ici, `list-style-type` et `list-style-position`, peuvent être combinées en une seule valeur de type `liste-style`. Lorsque vous utilisez la propriété `liste-style`, vous pouvez utiliser une ou toutes les valeurs de propriétés de style de liste à la fois. L'ordre de ces valeurs abrégées doit être `list-style-type` suivi d'une `liste-style-position`.

```
1 ul{
2   list-style: circle inside;
3 }
4 ol{
5   list-style: lower-roman;
6 }
```

Affichage horizontal d'une liste

Parfois, nous souhaitons afficher les listes horizontalement plutôt que verticalement. Peut-être que nous voulons diviser une liste en plusieurs colonnes, construire une liste de navigation ou mettre quelques éléments de liste dans une seule ligne.

En fonction du contenu et de l'apparence souhaitée, il existe plusieurs façons d'afficher les listes sous la forme d'une seule ligne, par exemple en créant la valeur de `display` des éléments `` à `inline` ou `inline-block` ou aussi en les faisant "floater".

Affichage d'une liste

La manière la plus rapide d'afficher une liste sur une seule ligne consiste à donner aux éléments `` une valeur de propriété `display` à `inline-block` ou `inline`. Cela place tous les éléments `` dans une seule ligne, avec un seul espace entre chaque élément de liste.

Si les espaces entre chacun des éléments `` sont gênants, ils peuvent être supprimés en utilisant les mêmes techniques que celles décrites dans le cours sur le «Positionnement du contenu».

Plus souvent qu'autrement, nous utiliserons la valeur de la propriété `inline-block` plutôt que la valeur de la propriété `inline`. La valeur de la propriété `inline-block` nous permet d'ajouter facilement des marges verticales et d'autres espaces aux éléments ``, alors que la valeur de la propriété `inline` ne l'est pas.

Lorsque vous modifiez la valeur de la propriété d'affichage en ligne ou en ligne, le marqueur de l'élément de liste, qu'il s'agisse d'une puce, d'un nombre ou d'un autre style, est supprimé.

HTML

```
1 <ul>
2   <li>Orange</li>
3   <li>Vert</li>
4   <li>Bleu</li>
5 </ul>
```

CSS

```
1 li {  
2   display: inline-block;  
3   margin: 0 0px;  
4 }
```

Orange Green Bleu

Démo d'affichage de liste en ligne

Liste "floattante"

La modification de la valeur de la propriété `display` en `inline` ou `inline-block` est rapide; Cependant, il supprime le marqueur de l'élément de liste. Si le marqueur de l'élément de liste est nécessaire, faire flotter chaque élément `` est une meilleure option que de changer la propriété `display`.

La propriété `float` des éléments `` appliquée à tous les éléments `` alignera horizontalement tous les éléments `` directement les uns à côté des autres sans espace entre eux. Lorsque nous faisons flotter chaque élément ``, le marqueur de l'élément de la liste s'affiche par défaut et se trouve en fait sur l'élément `` à côté de celui-ci. Pour éviter que le marqueur d'élément de liste ne soit affiché au-dessus d'autres éléments ``, une marge horizontale `margin` ou un `padding` doit être ajouté.

HTML

```
1 <ul>  
2   <li>Orange</li>  
3   <li>Vert</li>  
4   <li>Bleu</li>  
5 </ul>
```

CSS

```
1 li {  
2   float: left;  
3   margin: 0 20px;  
}
```

- Orange
- Green
- Bleu

Démo de liste "floattante"

Comme lorsqu'il flotte tout élément, cela brise le flux de la page. Nous devons nous rappeler pour effacer nos flotteurs-plus souvent avec le clearfix technique et retourner la page à son débit normal.

Exemple de liste de navigation

Nous allons souvent développer et trouver des menus de navigation à l'aide de listes non ordonnées. Ces listes sont généralement présentées sous forme de listes horizontales, en utilisant l'une des deux techniques mentionnées précédemment. Voici un exemple de menu de navigation horizontale marquée à l'aide d'une liste non ordonnée avec des éléments `` affichés en tant qu'éléments `inline-block`.

HTML

```
1 <nav class="navigation">
2   <ul>
3     <li><a href="#">Profil</a></li><!--
4     --><li><a href="#">Paramètres</a></li><!--
5     --><li><a href="#">Notifications</a></li><!--
6     --><li><a href="#">Déconnexion</a></li>
7   </ul>
8 </nav>
```

CSS

```
1 .navigation ul {
2   font: bold 11px "Helvetica Neue", Helvetica, Arial,
3   sans-serif;
4   margin: 0;
5   padding: 0;
6   text-transform: uppercase;
7 }
8 .navigation li {
9   display: inline-block;
```

```

10 }
11 .navigation a {
12     background: #395870;
13     background: linear-gradient(#49708f, #293f50);
14     border-right: 1px solid rgba(0, 0, 0, .3);
15     color: #fff;
16     padding: 12px 20px;
17     text-decoration: none;
18 }
19 .navigation a:hover {
20     background: #314b60;
21     box-shadow: inset 0 0 10px 1px rgba(0, 0, 0, .3);
22 }
23 .navigation li:first-child a {
24     border-radius: 4px 0 0 4px;
25 }
26 .navigation li:last-child a {
27     border-right: 0;
28     border-radius: 0 4px 4px 0;
29 }

```



Démo de liste de navigation

En pratique

Maintenant que nous savons construire des listes au format HTML et CSS, revenons à notre site Web et voyons où nous pourrions utiliser les listes.

Actuellement les menus de navigation dans les éléments `<header>` et `<footer>` de nos pages se composent d'une poignée d'éléments d'ancrage. Ces éléments d'ancrage pourraient être mieux organisées dans une liste à puces.

L'utilisation d'une liste non ordonnée et des éléments de liste va donner une structure à nos menus de navigation. Ces nouveaux éléments, cependant, afficherons nos menus de navigation de façon verticale.

Nous allons changer la valeur de `display` de nos éléments `` à `inline-block` pour qu'ils soient tous alignés sur une ligne. Quand nous faisons cela, cependant, nous devons également tenir compte de l'espace

laissé entre chaque élément ``. En se référant au cours sur le "Positionnement de contenu," nous savons que l'ouverture d'un commentaire HTML à la fin d'un élément `` et la fermeture d'un commentaire HTML au début d'un élément `` va supprimer cet espace. En gardant cela à l'esprit, le balisage pour le menu de navigation au sein de notre `<header>` va maintenant ressembler à ceci:

```
1 <nav class="nav nav-principal">
2   <ul>
3     <li>
4       <a href="index.html">Accueil</a></li><!--
5     --><li>
6       <a href="programme.html">Programme</a>
7     </li><!--
8     --><li>
9       <a href="horaires.html">Horaires</a>
10    </li><!--
11    --><li>
12      <a href="carte.html">Lieu</a>
13    </li><!--
14    --><li><a href="inscrire.html">S'inscrire</a></li>
15  </ul>
16 </nav>
```

Le long de ces mêmes lignes, le balisage pour le menu de navigations au sein de notre élément `<footer>` va maintenant ressembler à ceci:

```
1 <nav class="nav">
2   <ul>
3     <a href="index.html">Accueil</a></li><!--
4     --><li>
5       <a href="programme.html">Programme</a>
6     </li><!--
7     --><li>
8       <a href="horaires.html">Horaires</a>
9     </li><!--
10    --><li>
11      <a href="carte.html">Lieu</a>
12    </li><!--
13    --><li><a href="inscrire.html">S'inscrire</a></li>
14
```



```
15 </ul>
    </nav>
```

Il ne faut pas oublier de faire ces changements dans l'ensemble de nos fichiers HTML.

Avec la liste non ordonnée en place, veillons à ce que les éléments de la liste s'alignent horizontalement et nettoyons un peu leurs styles. Nous allons utiliser la classe de nav existante pour aider à cibler nos nouveaux styles.

Nous allons commencer par définir tous les éléments `` dans n'importe quel élément avec la valeur d'attribut de classe de "nav" à afficher en ligne `-inline-block-`, et inclure quelques marges horizontales et être alignés verticalement au haut `-top-` de l'élément.

En outre, nous allons utiliser le sélecteur de pseudo-classe `last-child` pour identifier le dernier élément `` et réinitialiser sa marge droite à 0. Cela garantit que tout l'espace horizontal entre l'élément `` et le bord de son élément parent est supprimé.

Dans notre fichier `style.css`, en dessous de nos styles de navigation existants, ajoutons les fichiers CSS suivants:

```
1 .nav li {
2   display: inline-block;
3   margin: 0 10px;
4   vertical-align: top;
5 }
6 .nav li:last-child {
7   margin-right: 0;
8 }
```

Vous vous demandez peut-être pourquoi notre liste non ordonnée n'a pas inclus de marqueurs d'élément de liste ou de styles par défaut. Ces styles ont été supprimés par la réinitialisation en haut de notre feuille de style. Si nous regardons la réinitialisation, nous verrons que nos éléments ``, `` et `` contiennent tous un `margin` et un `padding` de 0, et nos éléments `` et `` ont une valeur de `list-style` de `none`.

Nos menus de navigation ne sont pas les seuls endroits où nous utiliserons des listes. Nous les utiliserons également sur certaines de nos pages internes, y compris la page "Programme". Ajoutons quelques éléments de programme à notre page.

Dans notre fichier "programme.html" juste en dessous de notre section principale, créons une nouvelle section où nous présenterons tous nos conférenciers. En réutilisant certains styles existants, nous utiliserons un élément `<section>` avec une valeur d'attribut de classe "row" pour envelopper tous nos conférenciers et appliquer un arrière-plan blanc et un padding derrière eux. Dans l'élément `<section>`, nous ajouterons un élément `<div>` avec une valeur d'attribut de classe de la grille pour centrer nos conférenciers sur la page et nous permettre d'utiliser plusieurs colonnes pour ce faire.

Notre section de présentation ressemble donc à ceci:

```
1 <section class="row">
2   <div class="grid">
3     ...
4   </div>
5 </section>
```

A l'intérieur de la grille, chaque orateur sera présenté avec son propre élément `<section>`, qui comprendra deux colonnes.

La première colonne couvrira les deux tiers de l'élément `<section>` et sera délimitée à l'aide d'un élément `<div>`.

La deuxième colonne couvrira le tiers restant de l'élément `<section>` et sera délimitée à l'aide d'un élément `<aside>`, car son contenu est secondaire par rapport à l'interlocuteur et à son exposé spécifique.

En utilisant nos classes existantes `col-2-3` et `col-1-3`, le plan pour une section de conférencier ressemblera à ceci:

```
1 <section id="auteur1">
2   <div class="col-2-3">
3     ...
4   </div><!--
```

```

5   --><aside class="col-1-3">
6   ...
7   </aside>
8 </section>

```

Il ya quelques éléments à noter ici. Tout d'abord, chaque élément `<section>` pour chaque locuteur inclut un attribut ID avec le nom du locuteur comme valeur d'attribut. Plus tard, lorsque nous créerons le calendrier de notre conférence, ces attributs d'ID serviront d'ancres, ce qui nous permettra de faire le lien entre l'horaire et le profil d'un conférencier.

De plus, la balise de fermeture de l'élément `<div>` est suivie de l'ouverture d'un commentaire HTML et la balise d'ouverture de l'élément `<aside>` est précédée de la fermeture d'un commentaire HTML. Étant donné que les classes basées sur des colonnes afficheront ces éléments en tant qu'éléments `inline-block`, nous supprimons l'espace vide qui apparaîtra entre eux.

A l'intérieur de la colonne des deux tiers, marquée avec l'élément `<div>`, nous utiliserons quelques titres et paragraphes pour montrer le nom du conférencier, le titre et l'abrégé du discours et une courte biographie. En incluant ce contenu, une section de conférencier ressemblera à ceci:

```

1 <section id="auteur1">
2   <div class="col-2-3">
3     <h2>Auteur 1</h2>
4     <h5>Développer moins, pour construire plus</h5>
5     <p>...</p>
6     <h5>A propos de l'auteur</ h5>
7     <p>Développeur et concepteur de frontal,
8 l'auteur...</p>
9   </div><!--
10  --><aside class="col-1-3">
11  ...
12  </aside>
</section>

```

Dans la colonne d'un tiers, marquée d'un élément `<aside>`, nous allons ajouter un élément `<div>` avec une valeur d'attribut de classe "auteur-info". Nous utiliserons un élément `<div>` parce que nous allons bientôt ajouter des styles à cet élément. Avant d'entrer dans tous les

styles, cependant, ajoutons une liste non ordonnée dans l'élément `<div>` qui inclut comme éléments de liste des liens pertinents pour l'auteur. Maintenant, notre HTML pour un conférencier ressemblera à ceci:

```
1 <section id="auteur1">
2   <div class="col-2-3">
3     <h2>Auteur 1</h2>
4     <h5>Développer moins, pour construire plus</h5>
5     <p>...</p>
6     <h5>A propos de l'auteur</h5>
7     <p>Développeur et concepteur de frontal,
8 l'auteur...</p>
9   </div><!--
10  --><aside class="col-1-3">
11    <div class="auteur-info">
12      <ul>
13        <li>
14          <a href="https://twitter.com/auteur1">@auteur1</a>
15        </li>
16        <li>
17          <a href="http://www.auteur1.com/">Apprendre avec
18 l'auteur 1</a>
19        </li>
20      </ul>
21    </div>
22  </aside>
23 </section>
```

Ajoutos quelques styles à l'élément `<div>` avec la valeur de classe `auteur-info`,. Nous allons commencer par ajouter une nouvelle section dans notre fichier "style.css" pour les styles de page "Programme". À partir de là, ajoutons une bordure grise solide de 1 pixel avec un rayon de 5 pixels autour de n'importe quel élément qui inclut la valeur d'attribut de classe `auteur-info`. Ensuite, ajoutons une marge supérieure de 88 pixels pour positionner l'élément sur la même ligne verticale que le premier paragraphe de la description de la conversation et ajoutons également 22 pixels de padding vertical à l'intérieur de l'élément pour fournir de la place à la liste imbriquée.

Enfin, concentrons tout le texte dans l'élément.

Au total, notre CSS pour l'ensemble de règles de la classe `auteur-info` ressemblera à ceci:

```

1  /*
2  =====
3  Auteur
4  =====
5  */
6
7  .auteur-info {
8      border: 1px #dfe2e5 solid;
9      border-radius: 5px;
10     margin-top: 88px;
11     padding: 22px 0;
12     text-align: center;
13 }

```

Prenons une minute pour examiner pourquoi nous utilisons un élément `<div>` ici et les styles correspondants. Nous plaçons un élément `<div>` dans l'élément `<aside>` avec la valeur de classe `col-1-3`, car nous voulons que le padding hérité de la classe `col-1-3` se trouve à l'extérieur de la bordure sur la `<div>`. Ensuite nous incluons une image dans l'élément `<div>`, à côté de la liste non ordonnée; Nous avons donc créé un élément `<div>` plutôt que d'appliquer ces styles directement à l'élément ``. Comme nous ajoutons de plus en plus d'intervenants à la page, nous voulons veiller à ce qu'ils restent une distance verticalement égale l'un de l'autre. Pour ce faire, nous allons créer un ensemble de règles de classe "auteur" qui comprend une marge inférieure de 44 pixels, comme ceci:

```

1  .auteur {
2      margin-bottom: 44px;
3  }

```

Nous pouvons ensuite appliquer cette classe à l'élément `<section>` pour chaque orateur, à condition qu'il ne soit pas le dernier orateur. Nous allons omettre cette classe sur le dernier orateur, car nous ne voulons pas créer de marges inutiles avant notre élément `<footer>`. Avec plus d'un orateur, notre mise en page ressemblera à ceci:

```

1  <section class="row">
2      <div class="grid">

```

```

3      <section class="auteur" id="auteur2">
4          <div class="col-2-3">
5              ...
6          </div><!--
7      --><aside class="col-1-3">
8          ...
9      </aside>
10     </section>
11     <section id="auteur1">
12         <div class="col-2-3">
13             ...
14         </div><!--
15     --><aside class="col-1-3">
16         ...
17     </aside>
18 </section>
19 </div>
20 </section>

```

Remarquez comment le premier élément `<section>` de la classe "auteur", pour Auteur 2, inclut la valeur d'attribut de classe "auteur", qui le sépare verticalement de l'élément `<section>` supérieur celui de l'Auteur 1. Le dernier élément `<section>` de classe "auteur", donc pour Auteur 1, n'inclut pas une valeur d'attribut de classe du haut-parleur afin de le maintenir à une distance appropriée de l'élément `<footer>`.

Nos menus de navigation sont maintenant terminés, et la page du programme prend forme.

Programme

Nous sommes heureux d'accueillir des spécialistes pour présenter les dernières technologies du web.
Préparez vous pour un spectacle grandiose.

Auteur 1

PROGRAMMER MOINS, POUR PRODUIRE PLUS

La conférence proposée donnera une vue 360 sur les technos web pour fabriquer un site de qualité à moindre frais...

A PROPOS D'AUTEUR 1

Développeur, designer, intégrateur en gros full-stack et fier de l'être

@auteur1
auteur1.com

Auteur 2

PROGRAMMER MOINS, POUR PRODUIRE PLUS

La conférence proposée donnera une vue 360 sur les technos web pour fabriquer un site de qualité à moindre frais...

A PROPOS D'AUTEUR 2

Développeur, designer, intégrateur en gros full-stack et fier de l'être

@auteur2
auteur2.com

Auteur 3

PROGRAMMER MOINS, POUR PRODUIRE PLUS

La conférence proposée donnera une vue 360 sur les technos web pour fabriquer un site de qualité à moindre frais...

A PROPOS D'AUTEUR 3

Développeur, designer, intégrateur en gros full-stack et fier de l'être

@auteur3
auteur3.com

Notre page programme après la mise à jour nos menus de navigation et des sections pour les auteurs.