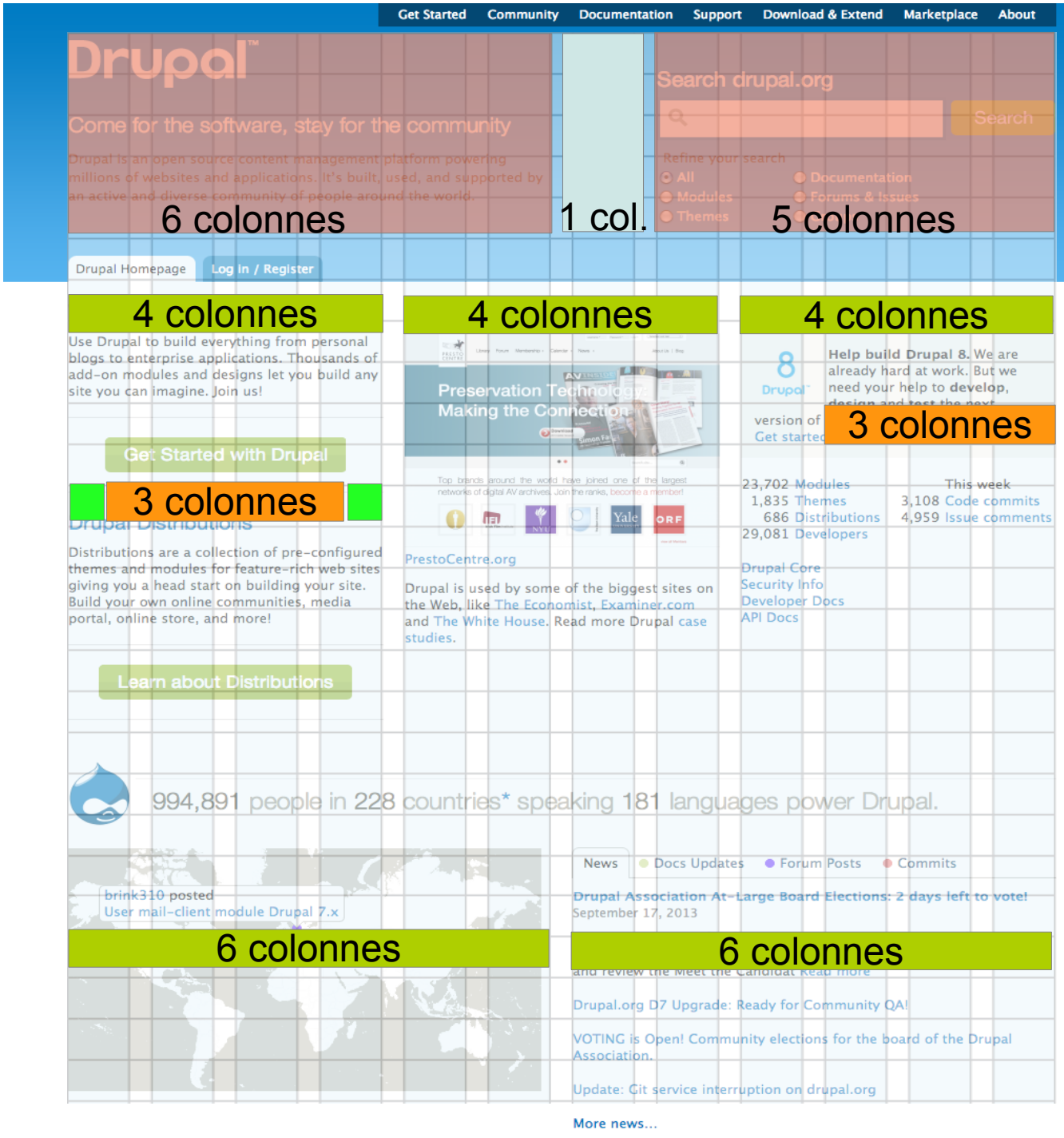
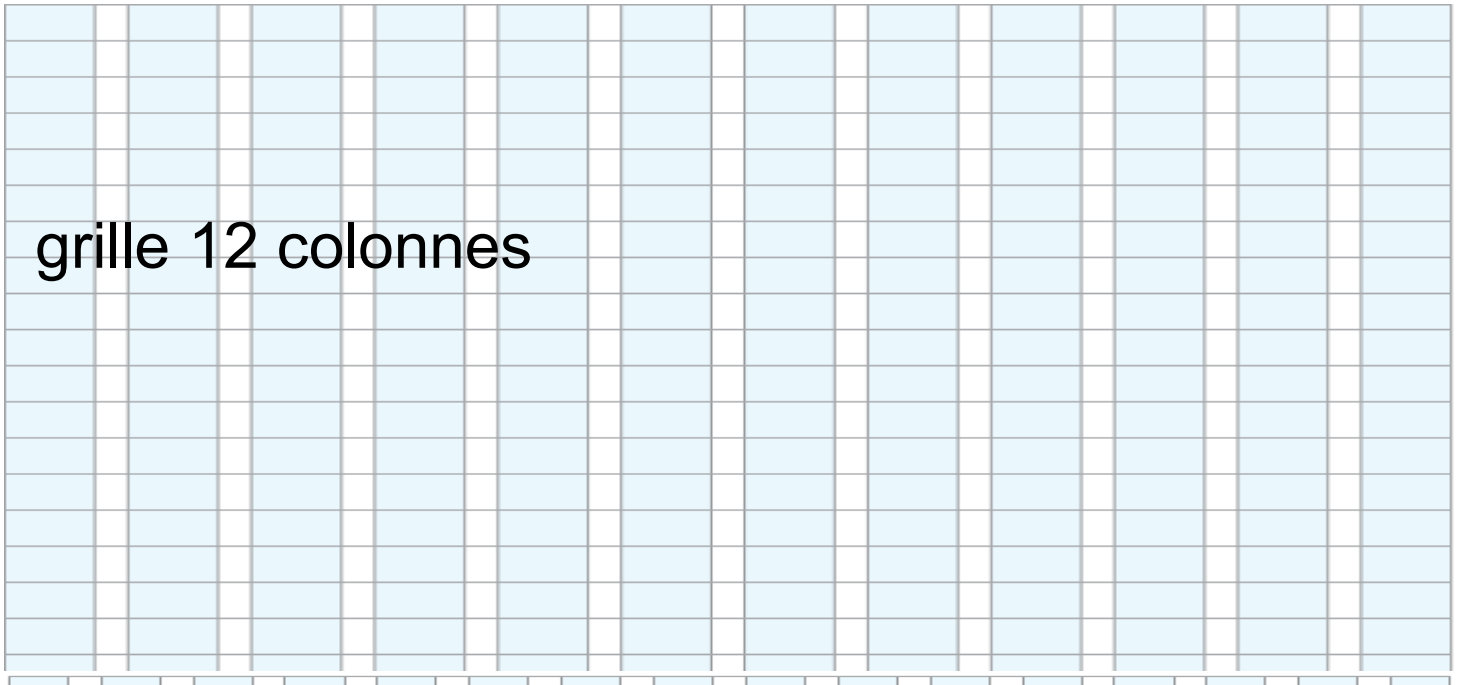
Grilles & positionnement d'éléments

principe de conception : ALIGNER !

- pour créer de la régularité
- pour créer de l'équilibre
- pour créer des liens entre les éléments


Get Started
Community
Documentation
Support
Download & Extend
Marketplace
About
Drupal
Come for the software, stay for the community
Drupal is an open source content management platform powering millions of websites and applications. It's built, used, and supported by an active and diverse community of people around the world.
Search drupal.org
Search
Refine your search
All
Modules
Themes
Documentation
Forums & Issues
6 colonnes
1 col.
5 colonnes
Drupal Homepage
Log in / Register
4 colonnes
4 colonnes
4 colonnes
Use Drupal to build everything from personal blogs to enterprise applications. Thousands of add-on modules and designs let you build any site you can imagine. Join us!
Get Started with Drupal
3 colonnes
Drupal Distributions
Distributions are a collection of pre-configured themes and modules for feature-rich web sites giving you a head start on building your site. Build your own online communities, media portal, online store, and more!
Learn about Distributions
Preservation Technology: Making the Connection
PrestoCentre.org
Drupal is used by some of the biggest sites on the Web, like The Economist, Examiner.com and The White House. Read more Drupal case studies.
8
Drupal
Help build Drupal 8. We are already hard at work. But we need your help to develop,
version of
Get started
3 colonnes
23,702 Modules
1,835 Themes
686 Distributions
29,081 Developers
This week
3,108 Code commits
4,959 Issue comments
Drupal Core
Security Info
Developer Docs
API Docs
994,891 people in 228 countries* speaking 181 languages power Drupal.
brink310 posted
User mail-client module Drupal 7.x
News
Docs Updates
Forum Posts
Commits
Drupal Association At-Large Board Elections: 2 days left to vote!
September 17, 2013
6 colonnes
6 colonnes
Drupal.org D7 Upgrade: Ready for Community QA!
VOTING is Open! Community elections for the board of the Drupal Association.
Update: Git service interruption on drupal.org
More news...


grille 12 colonnes


grille 16 colonnes

- Pour aligner efficacement : placer les éléments sur une grille de mise en page, caractérisée par
  - largeur/largeur max. totale : 100 %, 80 %, 64em
  - nombre de colonnes : 12, 16, 24
  - largeur des gouttières (gutter) : 0 , 2%, 1em

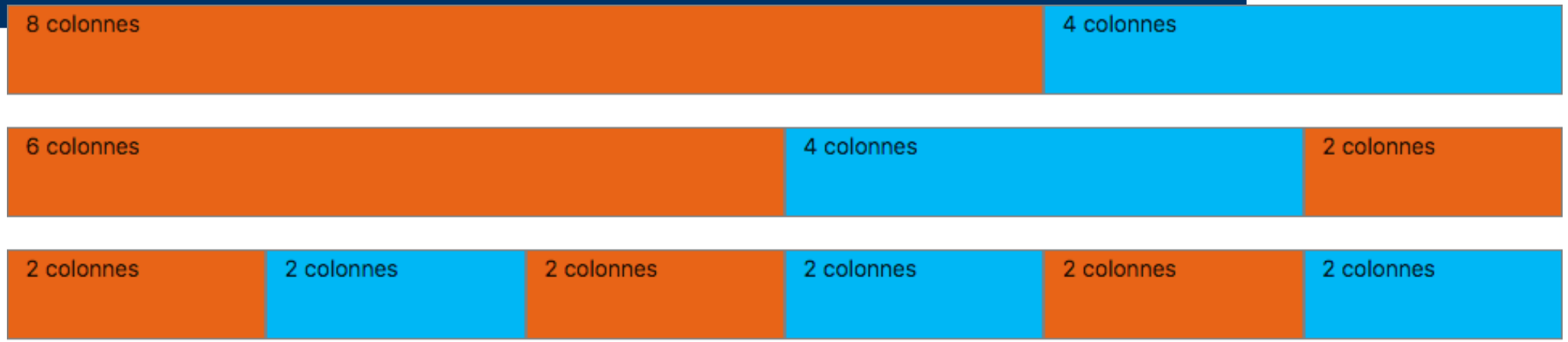
techniques d'intégration

- utiliser une grille générique prédéfinie
  - par exemple fournie par un framework
- calculer de façon ad-hoc les dimensions de chaque élément et les marges nécessaires pour un design donné, en fonction des caractéristiques de la grille utilisée,
- fabriquer sa propre grille générique, éventuellement paramétrée par ses caractéristiques
- utiliser le mode de positionnement `css grid`

utiliser une grille générique existante

- principe : la grille est fournie sous la forme de classes css prédéfinies que l'on insère dans le document html
- les caractéristiques de la grille (largeur, nb colonnes, gouttières) sont fixées par le concepteur de la grille
- intérêt : facile et rapide d'emploi
- inconvénient : adaptation difficile aux besoins particuliers
- exemple : la grille materialize css
  - 12 colonnes, largeur 100 %, gutter 0

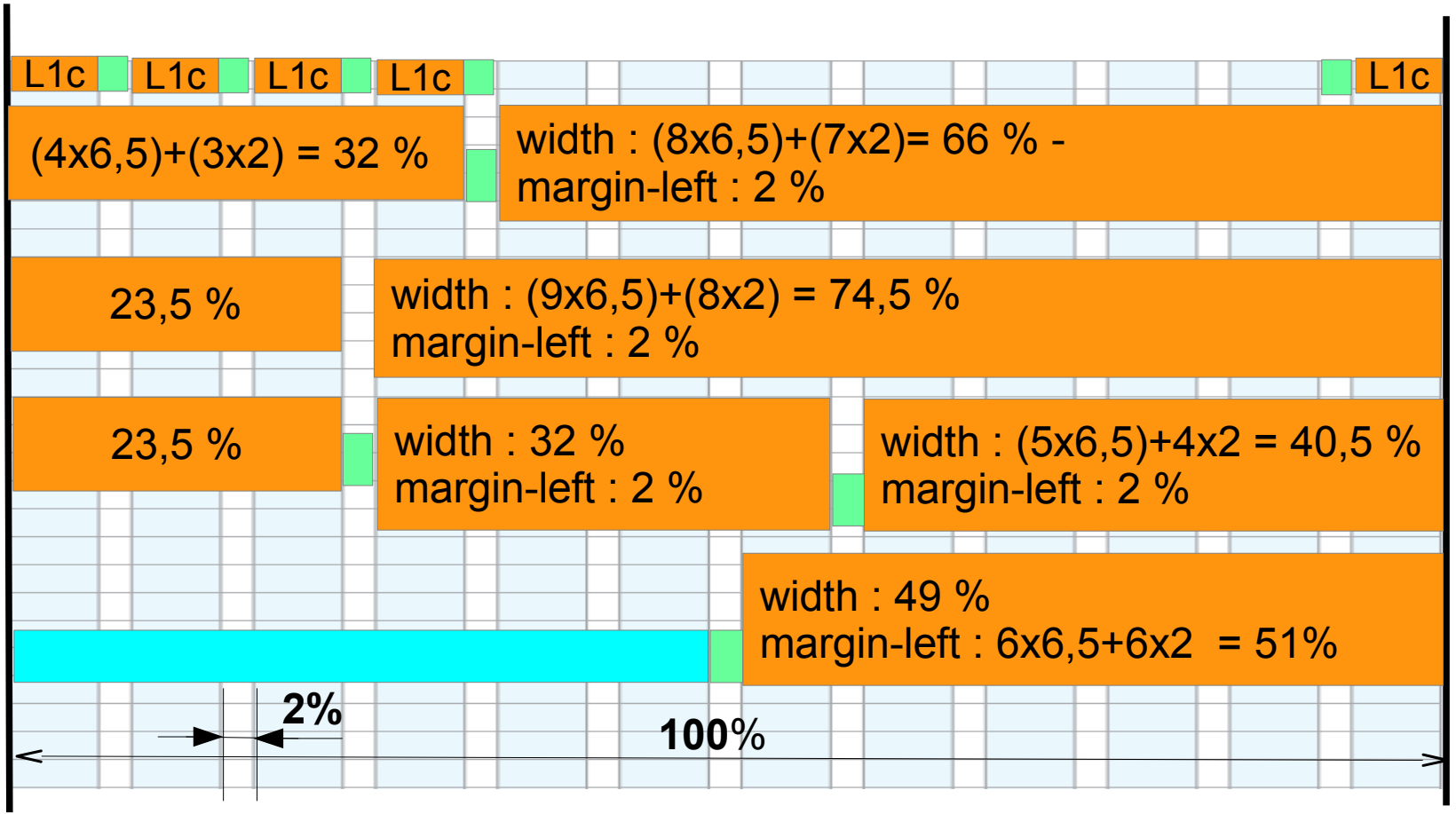
```
<div class="row">
    <div class="col m8"> 8 colonnes </div>
    <div class="col m4"> 4 colonnes </div>
</div>
<div class="row">
    <div class="col m6"> 6 colonnes</div>
    <div class="col m4"> 4 colonnes</div>
    <div class="col m2"> 2 colonnes</div>
</div>

<div class="row" >
    <div class="col m2" >2 colonnes</div>
    ...
</div>
```



placement ad-hoc

- utiliser flex pour placer les éléments horizontalement
- on se base sur les caractéristiques de la grille utilisée pour la conception pour calculer les largeurs des différents éléments
- dimensionner en % pour avoir des éléments fluides

- largeur 1 colonne L1c = (100% - 11 gouttières) / 12
  - L1c = (100 – 22) / 12 = 6,5 %
- gouttière : marge gauche
- largeur n colonnes : (n x L1c) + (n-1) x gouttière
- décalage : (nxL1c) + n x gouttière



construire sa grille générique

- principe la grille est fournie sous la forme d'un ensemble de classes css
  - conteneur global éventuel
  - ligne
  - cellules de toutes les tailles possibles
  - décalages (offset)

réalisation (flex)

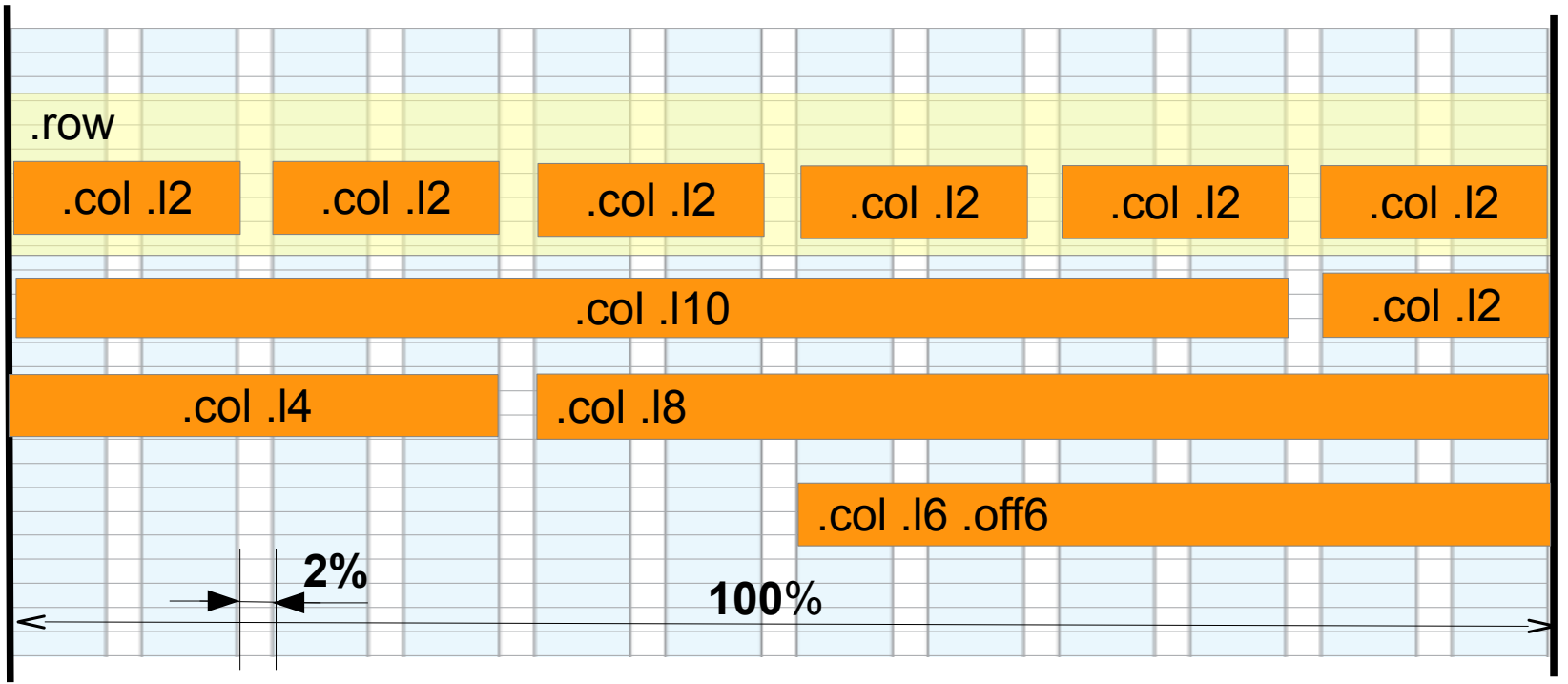
- **conteneur global** : fixe la largeur de la grille et éventuellement les espacements gauches et droits
- **conteneur de ligne** de type flex
  - `display : flex ; flex-direction : row`
- **les éléments** :
  - taille définie avec width ou flex-basis ; en % (fluidité)
  - `flex-grow : 0` pour éviter qu'ils grandissent si la ligne n'est pas complète
  - `flex-shrink : 1 / nowrap` : pour qu'ils rétrecissent de manière identique si la ligne est trop remplie
  - `flex-shrink : 0 / wrap` : pour qu'ils gardent leur taille et débordent sur la ligne suivante si la ligne est trop remplie

réalisation (flex)

- les **gouttières** peuvent être réalisées :
  - avec des **marges** appliquées à gauche de chaque élément colonne
  - avec du **padding** à l'intérieur des éléments colonne
- prévoir la possibilité de **supprimer** les goulottes

- les **offsets** : réalisés avec des marges

- largeur 1 colonne L1c = (100% - 11 gouttières) / 12
  - L1c = (100 – 22) / 12 = 6,5 %
- gouttière : marge gauche
- largeur n colonnes : (n x L1c) + (n-1) x gouttière
- décalage : (nxL1c) + n x gouttière



grille adaptative

- pour fabriquer une grille permettant l'adaptation : prévoir des classes actives à différentes largeurs de viewport , définies dans des media-query

```
<div class="row">
  <div class="col s6 m8"> … </div>
  <div class="col s6 m4"> … </div>
</div>
```

```
.s6 {
  flex-basis: calc( (89% / 12) * 6 +  5%) ;
}

@media screen and  (min-width: 56em) {
  .m4 {
    flex-basis: calc( (89% / 12) * 4 + 3%) ;}
  .m8 {
    flex-basis: calc( (89% / 12) * 8 + 7%) ;}
}
```

css grid layout

- mode de placement basé sur la définition d'1 grille
- Le conteneur définit la structure de la grille
  - nbre et largeur des colonnes
  - nbre et hauteur des lignes
- les items sont placés sur cette grille :
  - explicitement : on précise le rectangle occupé par l'item sur la grille
  - implicitement : on précise uniquement la taille de l'item, et il est placé automatiquement
- on peut paramétrer le placement implicite :
  - en ligne, en colonne, occupation maximale
- exemple

```
.grid_container {
  display: grid;
  grid-template-columns : 10% 50px 1fr  2fr; // les colonnes
  grid-template-rows: 5rem 4rem;              // les lignes
  grid-auto-flow: row;                    // placemement implicite
  grid-gap: .2em;  //gouttières           // en ligne
}

.grid_cell { … }


.grid_cell:nth-child(1) {          // placement explicite
  grid-column : 3 / 5;             // sur la grille
  grid-row: 1/3;
  background-color: cyan;
}
.grid_cell:nth-child(5n+2) {       // placement implicite
  background-color: maroon;
  grid-column: span 2;             // largeur 2 colonnes
}
.grid_cell:nth-child(5n+3) {       // placement implicite
  background-color: green;

}
```