



# 50 nuances de grilles

photo CC Vladimir Kudinov <https://unsplash.com/@madbyte>

# je suis alsacien



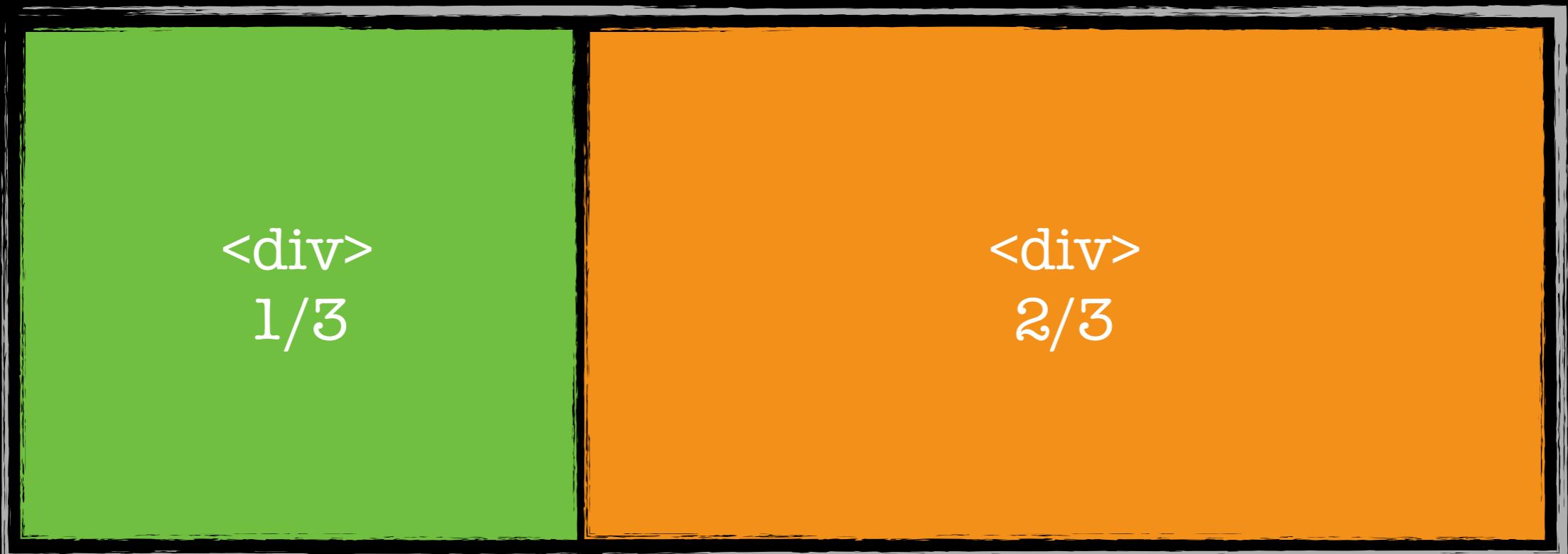
raphaël goetter  
**alsacreations** 



# échauffement

rapide mise en bouche

# TODO!



L'une de ces deux  
méthodes utilise  
Bootstrap

sauras-tu la retrouver ? 🤔

```
<div class="container">  
  <div class="row no-gutters">  
    <div class="col-4">div</div>  
    <div class="col-8">div</div>  
  </div>  
</div>  
  
.col-4 {  
  flex: 0 0 33.333333%;  
  max-width: 33.333333%;  
}  
.col-8 {  
  flex: 0 0 66.666667%;  
  max-width: 66.666667%;  
}  
  
.row {  
  display: flex;  
  flex-wrap: wrap;  
  margin-right: -15px;  
  margin-left: -15px;  
}  
  
.no-gutters {  
  margin-right: 0;  
  margin-left: 0;  
}  
  
> .col,  
> [class*="col-"] {  
  padding-right: 0;  
  padding-left: 0;  
}  
}
```



```
<div class="container">  
  <div>div</div>  
  <div>div</div>  
</div>
```



```
.container {  
  display: grid;  
  grid-template-columns: 1fr 2fr;  
}
```

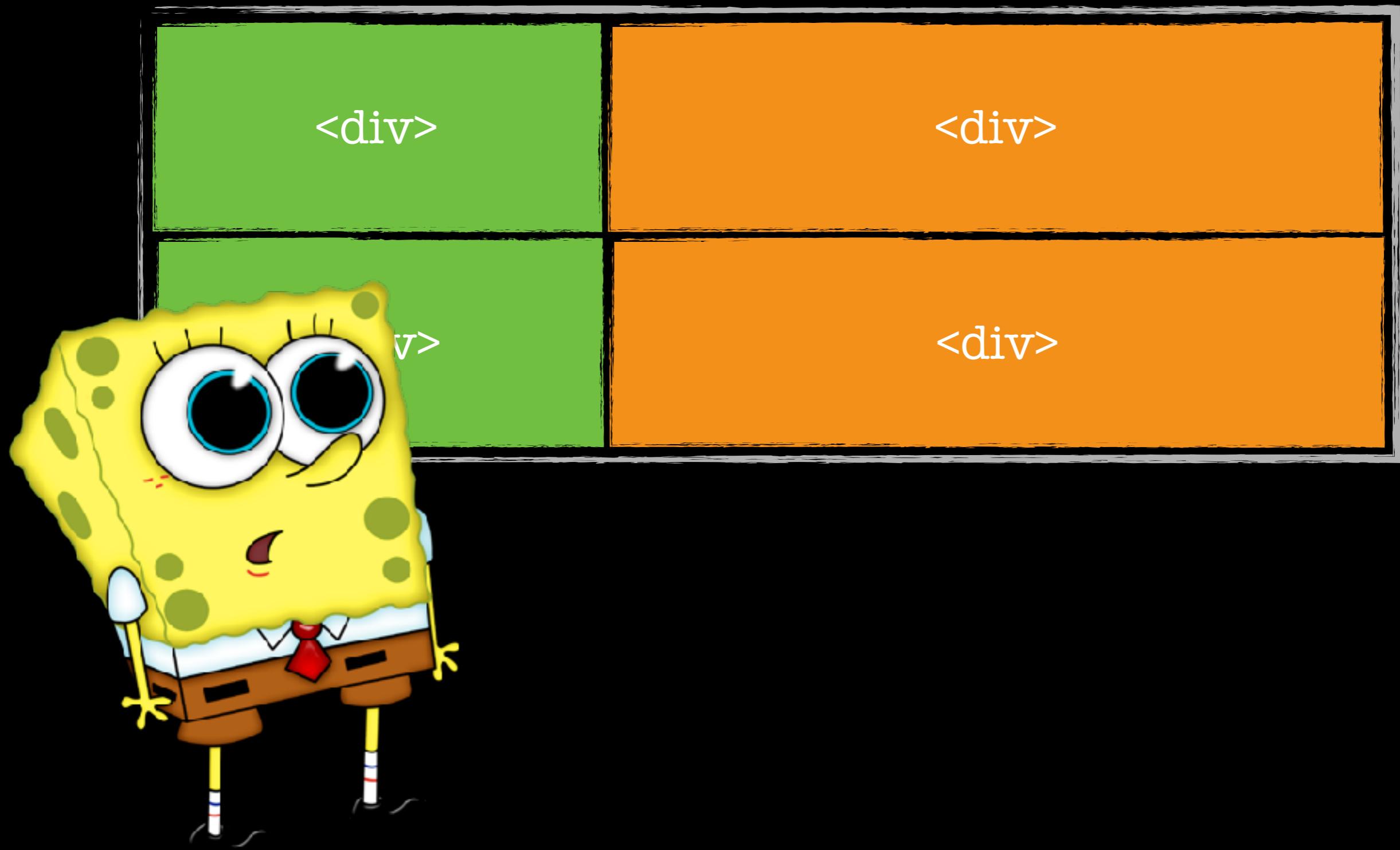
# bootstrap 4

```
<div class="container">
  <div class="row no-gutters">
    <div class="col-4">div</div>
    <div class="col-8">div</div>
  </div>
</div>
.row {
  display: flex;
  flex-wrap: wrap;
  margin-right: -15px;
  margin-left: -15px;
}
.col-4 {
  flex: 0 0 33.333333%;
  max-width: 33.333333%;
}
.col-8 {
  flex: 0 0 66.666667%;
  max-width: 66.666667%;
}
.no-gutters {
  margin-right: 0;
  margin-left: 0;
}
> .col,
> [class*="col-"] {
  padding-right: 0;
  padding-left: 0;
}
```

# grid layout

```
<div class="container">
  <div>div</div>
  <div>div</div>
</div>
.container {
  display: grid;
  grid-template-columns: 1fr 2fr;
}
```

# TODO!



```
<div class="container">  
  <div class="row no-gutters">  
    <div class="col-4">div</div>  
    <div class="col-8">div</div>  
  </div>  
  <div class="row no-gutters">  
    <div class="col-4">div</div>  
    <div class="col-8">div</div>  
  </div>  
</div>
```

```
.col-4 {  
  flex: 0 0 33.33333%;  
  max-width: 33.33333%;  
}  
.col-8 {  
  flex: 0 0 66.66667%;  
  max-width: 66.66667%;  
}
```

```
.row {  
  display: flex;  
  flex-wrap: wrap;  
  margin-right: -15px;  
  margin-left: -15px;  
}  
.no-gutters {  
  margin-right: 0;  
  margin-left: 0;  
}> .col,  
> [class*="col-"] {  
  padding-right: 0;  
  padding-left: 0;  
}  
}
```



```
<div class="container">  
  <div>div</div>  
  <div>div</div>  
  <div>div</div>  
  <div>div</div>  
</div>
```



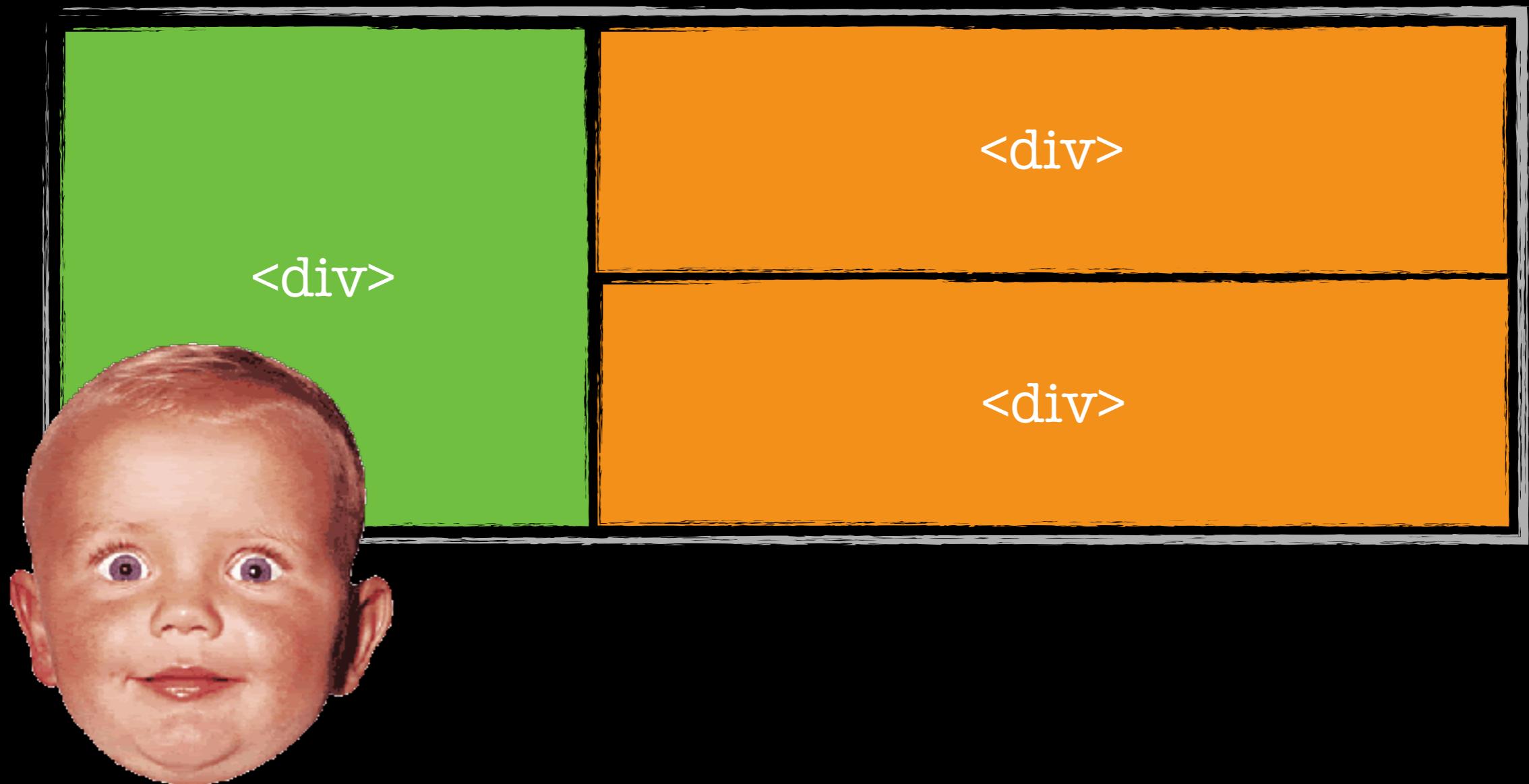
```
.container {  
  display: grid;  
  grid-template-columns: 1fr 2fr;  
}
```

# bootstrap

# grid layout



# TODO!



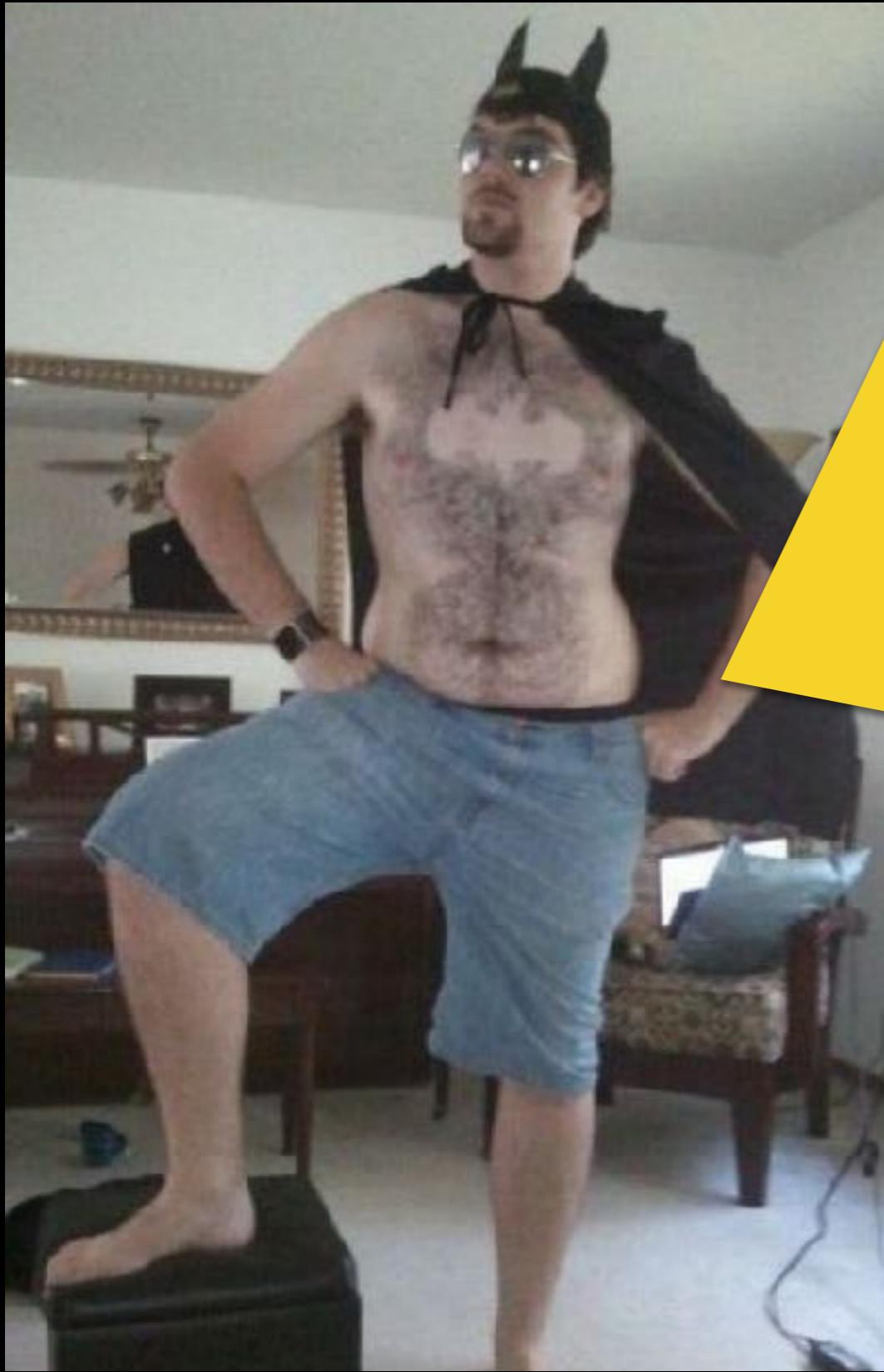


```
<div class="container">  
  <div>div</div>  
  <div>div</div>  
  <div>div</div>  
</div>
```



```
.container {  
  display: grid;  
  grid-template-columns: 1fr 2fr;  
}  
div:first-child {  
  grid-row: span 2;  
}
```

# bootstrap



# grid layout



# bootstrap.min.css (juste la grille)

```
html{font-family:sans-serif;-ms-text-size-adjust:100%;-webkit-text-size-adjust:100%}body{margin:0}article,aside,details,figcaption,figure,footer,header,hgroup,main,menu,nav,section,summary{display:block}audio,canvas,progress,video{display:inline-block;vertical-align:baseline}audio:not([controls]) {display:none; height:0} [hidden],template{display:none} a{background-color:transparent} a:active,a:hover{outline:0} abbr[title]{border-bottom:1px dotted} b, strong{font-weight:bold} dfn{font-style:italic} h1{font-size:2em; margin:0.67em 0} mark{background:#ff0; color:#000} small{font-size:80%} sub,sup{font-size:75%; line-height:0; position:relative; vertical-align:baseline} sub{top:-0.25em} sup{bottom:-0.25em} img{border:0} svg:not(:root){overflow:hidden} figure{margin:1em 40px} hr{ webkit-box-sizing:content-box; moz-box-sizing:content-box; box-sizing:content-box; height:0} pre{overflow:auto} code,kbd,pre,samp{font-family:monospace;font-size:1em} button,input,optgroup,select,textarea{color:inherit;font:inherit; margin:0} button{overflow:visible} button,select{text-transform:none} button,html,input[type="button"],input[type="reset"],input[type="submit"]{-webkit-appearance:button; cursor:pointer} button[disabled],html,input[disabled]{cursor:default} button,:moz-focus-inner{border:0;padding:0} input{line-height:normal} input[type="checkbox"],input[type="radio"],input{width:1em; border:1px solid black; border-radius:2px; border-sizing:border-box; box-sizing:border-box; padding:0} input[type="number"]::-webkit-outer-spin-button, input[type="number"]::-webkit-inner-spin-button{height:auto} input[type="search"]{border:1px solid black; border-radius:2px; border-sizing:content-box; box-sizing:content-box; width:100%} input[type="search"]::-webkit-search-cancel-button, input[type="search"]::-webkit-search-decoration{-webkit-appearance:none} fieldset{border:1px solid #ccc; margin:0.2em; padding:0.35em; 0.25em} legend{border:0; padding:0} textarea{overflow:auto} optgroup{font-weight:bold} table{border-collapse:collapse; border-spacing:0} td,th{padding:0} *{-webkit-box-sizing:border-box; -moz-box-sizing:border-box; box-sizing:border-box} html{font-size:10px; -webkit-tap-highlight-color:rgba(0,0,0,0)} body{font-family:"Helvetica Neue", Helvetica, Arial, sans-serif; font-size:14px; line-height:1.42857143; color:#333; background-color:#fff} input,button,select,textarea{font-family:inherit; font-size:inherit; line-height:inherit} a{color:#337ab7; text-decoration:none} a:hover,a:focus{color:#23527c; text-decoration:underline} a:outline{border:2px solid #ccc; outline-offset:-2px} figure{margin:0} img{vertical-align:middle} .img-responsive{display:block; max-width:100%; height:auto} .img-rounded{border-radius:10px} .img-thumbnail{padding:4px; line-height:1.42857143; background-color:#fff; border:1px solid #ddd; border-radius:4px; -webkit-transition:all .2s ease-in-out; transition:all .2s ease-in-out} t{transition:all .2s ease-in-out; display:inline-block; max-width:100%; height:auto} .img-circle{border-radius:50%} hr{margin-top:20px; margin-bottom:20px; border:0; border-top:1px solid #eee} :sr-only{position: absolute; width:1px; height:1px; margin:-1px; padding:0; overflow:hidden; clip:rect(0,0,0,0)} :sr-only-focusable:active,.sr-only-focuse
```

# tu me montres ?

The image shows a screenshot of a web development environment, likely a browser developer tools or a code editor like VS Code. On the left, there are two panes: one for HTML and one for CSS.

**HTML:**

```
1 <div class="container">
2   <div class="salade">salade</div>
3   <div class="tomate">tomate</div>
4   <div class="oignon">oignon</div>
5   <div class="picon">picon bière</div>
6   <div class="bretzel">bretzel</div>
7 </div>
```

**CSS:**

```
1 .container {
2   border: 4px dotted hotpink;
3 }
4
5
6
7
8
9
10
11
12
```

On the right, there is a visual representation of the rendered HTML elements. Each element is a colored rectangle with its text content inside. They are arranged vertically and have a dashed pink border around them.

- salade (yellow)
- tomate (pink)
- oignon (green)
- picon bière (blue, highlighted)
- bretzel (grey)



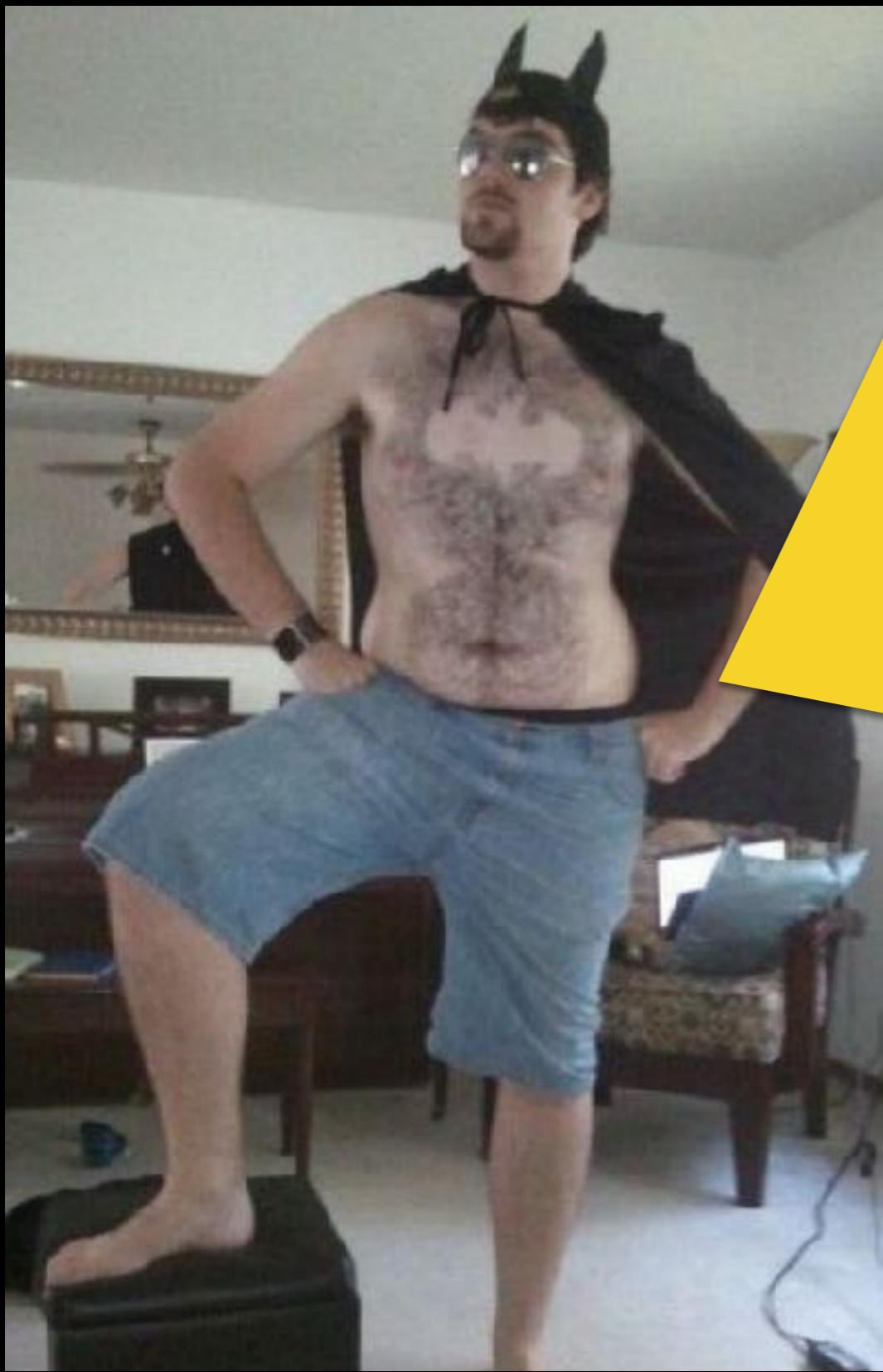
MAIS ON  
FAiSAiT CA EN 1996  
AVEC <TABLE> !

HEIN MARTY ?



**NOPE**

# <table> layout



# grid layout



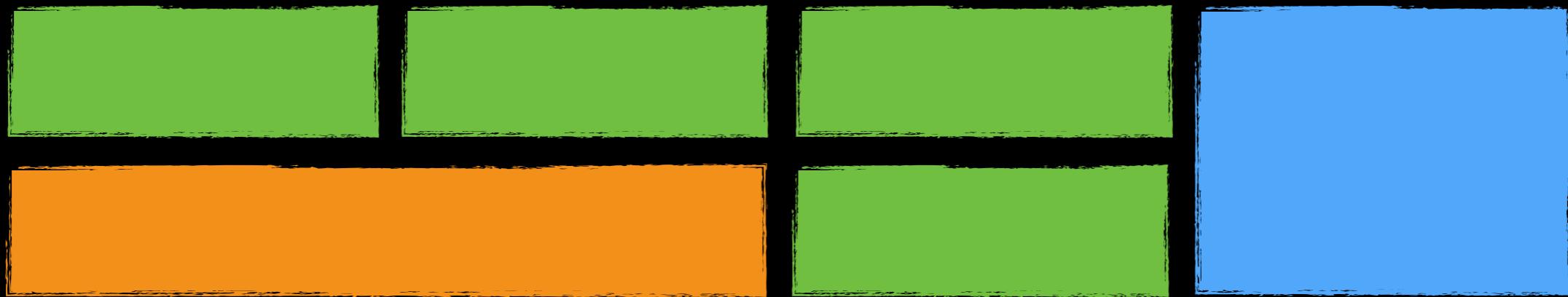
A scene from the movie Back to the Future. On the left, Doc Brown (played by Christopher Lloyd) is shown from the waist up, wearing his signature white lab coat over a green shirt, holding a small electronic device with a antenna. On the right, Marty McFly (played by Michael J. Fox) stands looking up at Doc. Two speech bubbles originate from their heads. The top bubble is large and oval-shaped, containing the text "OUI MAIS ON FAiSAiT CA EN 2014 AVEC FLEXBOX!". The bottom bubble is smaller and more rounded, containing the text "ACHETEZ LE LiVRE DE RAPHAEL".

OUI MAIS ON  
FAiSAiT CA EN 2014  
AVEC FLEXBOX !

ACHETEZ LE  
LiVRE DE  
RAPHAEL



# grid layout



# flexbox

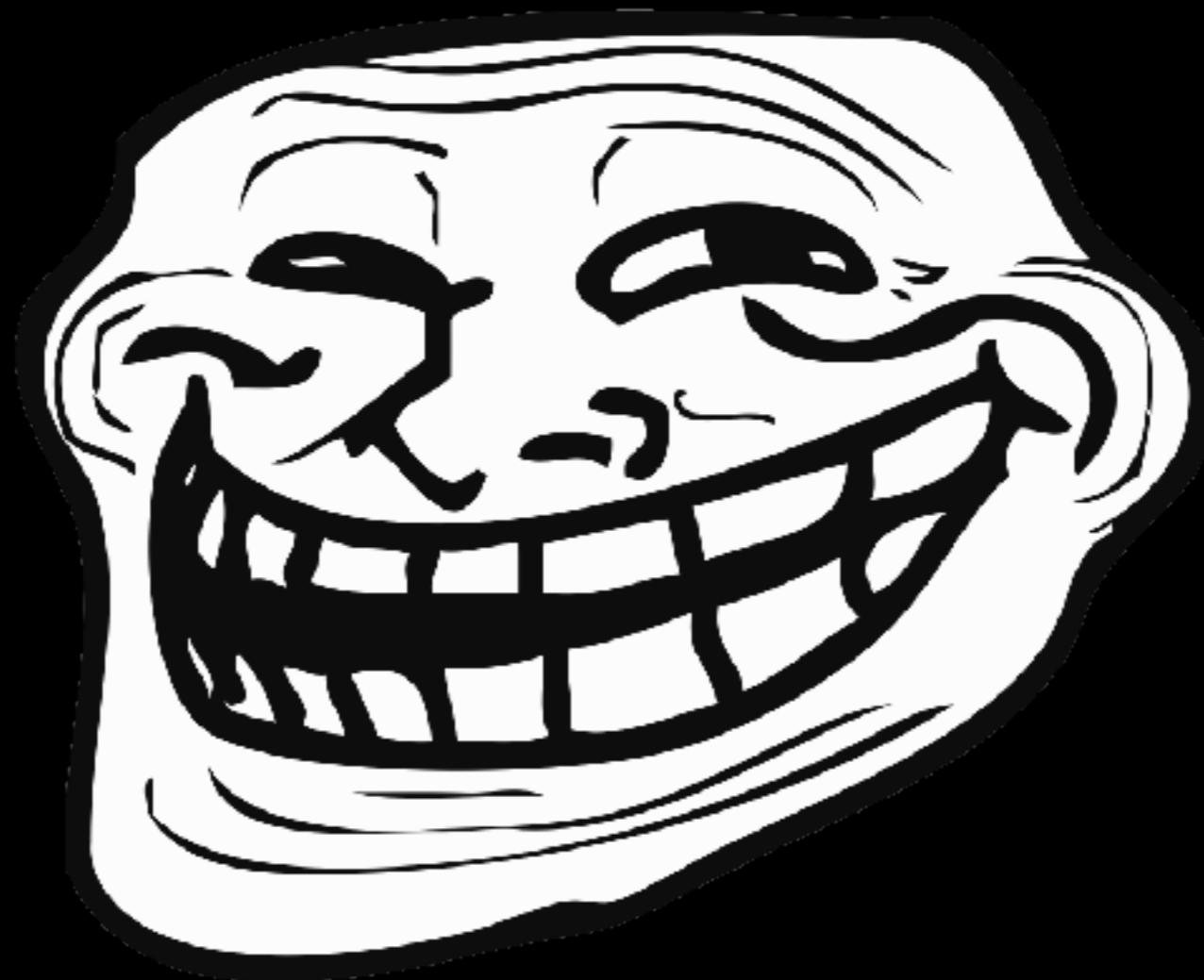




A scene from the movie Back to the Future. On the left, Doc Brown (played by Christopher Lloyd) is shown from the waist up, wearing his signature lab coat and holding a small electronic device with a antenna. On the right, Marty McFly (played by Michael J. Fox) stands looking at Doc. A large, hand-drawn style speech bubble originates from Doc's head, containing the text "EUH MAIS ON FAIT CA AUJOURD'HUI AVEC BOOTSTRAP !". A smaller, separate speech bubble originates from Marty's head, containing the text "OU KNACSS".

EUH MAIS ON FAiT  
CA AUJOURD'HUi  
AVEC BOOTSTRAP !

OU  
KNACSS



# bootstrap.min.css (juste la grille)

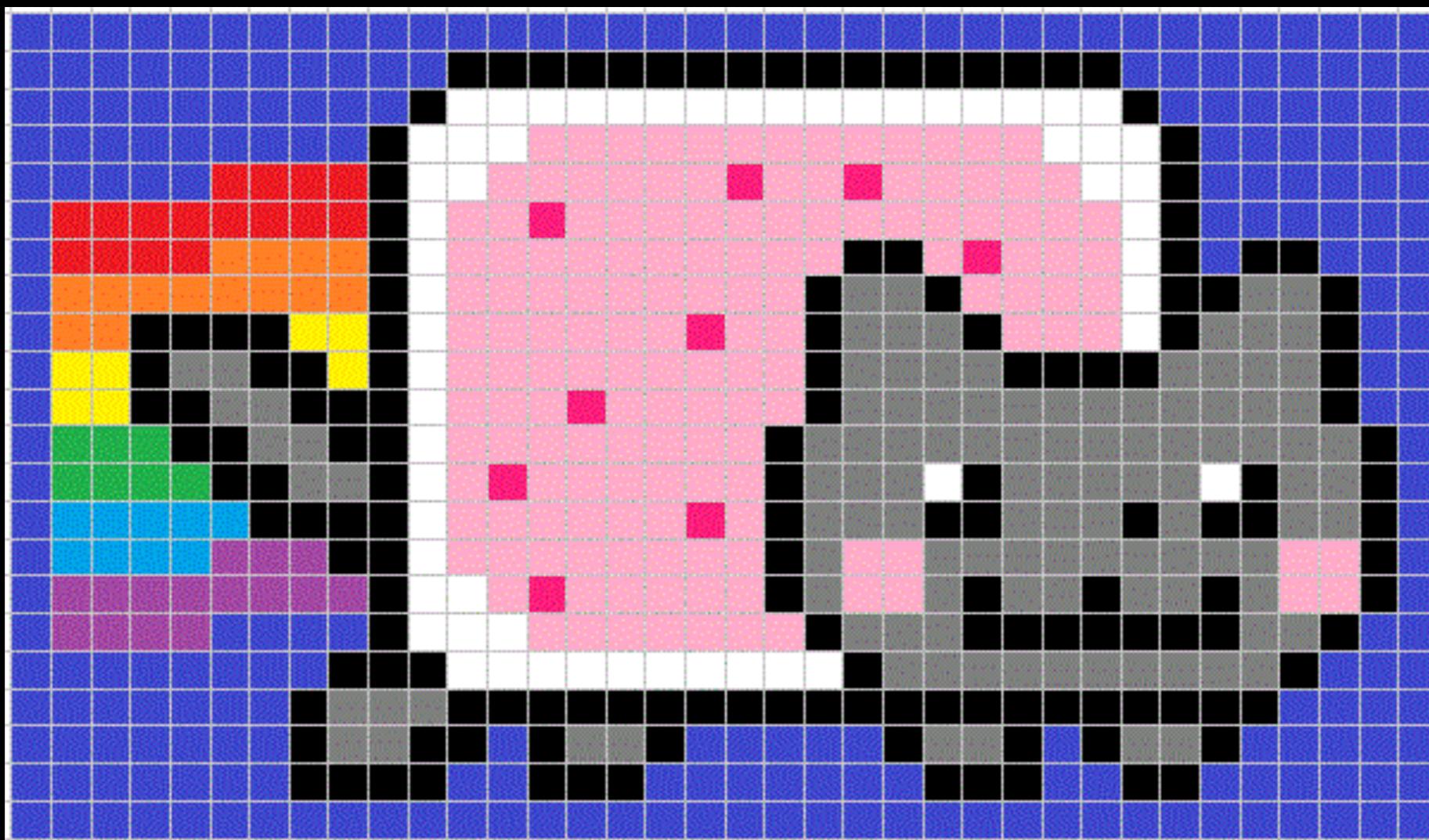
```
font-family: sans-serif; -ms-text-size-adjust: 100%; -webkit-text-size-  
adjust: 100%; body { margin: 0; } article, aside, details, figcaption, figure, footer, header, hgroup, main, menu, nav, section, summary { display: block; } audio, canvas, progress, video { display: inline-block; vertical-align: baseline; } audio { display: none; height: 0; } template { display: none; } a { background-color: transparent; } a:active, a:hover { outline: 0; } b { border-bottom: 1px dotted; } b, strong { font-weight: bold; } dfn { font-style: italic; } h1 { font-size: 2em; margin: 0.67em 0; } mark { background: #ff0; color: #000; } small { font-size: 80%; } sub, sup { font-size: 75%; line-height: 0; position: relative; vertical-align: baseline; } sub { top: -0.25em; } sup { bottom: -0.25em; } img { border: 0; } svg { not: root { overflow: hidden; } } figure { margin: 1em 40px; } hr { webkit-box-sizing: content-box; box-sizing: content-box; height: 0; } pre { overflow: auto; } code, kbd, pre, samp { font-family: monospace, monospace; font-size: 1em; } button, input, optgroup, select, textarea { color: inherit; font: inherit; margin: 0; } button { overflow: visible; } button, select { text-transform: none; } button, html input[type="button"], input[type="reset"], input[type="submit"] { -webkit-appearance: button; cursor: pointer; } button[disabled], html input[disabled] { cursor: default; } button::-moz-focus-inner, input::-moz-focus-inner { border: 0; padding: 0; } input { line-height: normal; } input[type="checkbox"], input[type="radio"] { -webkit-appearance: button; box-sizing: border-box; box-sizing: border-box; border: 0; padding: 0; } input[type="number"] { -webkit-appearance: spinbutton; } input[type="number"] { -webkit-appearance: searchfield; -webkit-appearance: textfield; -webkit-box-sizing: content-box; box-sizing: content-box; border: 1px solid #ccc; margin: 0 2px; padding: 0.35em 0.625em; } legend { border: 0; padding: 0; } textarea { overflow: auto; } optgroup { font-weight: bold; } table { border-collapse: collapse; border-spacing: 0; } td, th { padding: 0; } ::-webkit-box-sizing: border-box; box-sizing: border-box; ::-webkit-tap-highlight-color: rgba(0, 0, 0, 0); } body { font-family: "Helvetica Neue", Helvetica, Arial, sans-serif; font-size: 14px; line-height: 1.42857143; color: #333; background-color: #fff; } input, button, select, textarea { font-family: inherit; font-size: inherit; line-height: inherit; } a { color: #337ab7; text-decoration: none; } a:hover, a:focus { color: #23527c; text-decoration: underline; } a:focus { outline: 2px auto -webkit-autowidth; -webkit-focus-ring-color: outline-offset: -2px; } figure { margin: 0; } img { vertical-align: middle; } img { responsive: display: block; max-width: 100%; height: auto; } .img-rounded { border-radius: 5px; } .img-thumbnial { padding: 4px; line-height: 1.42857143; background-color: #fff; border: 1px solid #ddc; border-radius: 4px; -webkit-transition: all 0.2s ease-in-out; transition: all 0.2s ease-in-out; } display: inline-block; max-width: 100%; height: auto; } .img-circle { border-radius: 50%; } hr { margin-top: 20px; margin-bottom: 20px; border: 1px solid #aaa; } sr-only { position: absolute; width: 1px; height: 1px; margin: -1px; padding: 0; overflow: hidden; clip: rect(0, 0, 0, 0); border: 0; } sr-only-focusable: active, sr-only-  
focusable: focus { position: static; width: auto; height: auto; margin: 0; overflow: visible; clip: auto; } [role="button"] { cursor: pointer; } container { margin-right: auto; margin-left: auto; padding-left: 15px; padding-right: 15px; } @media (min-width: 768px) { .container { width: 750px; } } @media (min-width: 992px) { .container { width: 970px; } } @media (min-width: 1200px) { .container { width: 1170px; } } .container-fluid { margin-right: auto; margin-left: auto; padding-left: 15px; padding-right: 15px; } .row { margin-left: -15px; margin-right: -15px; } .col-xs-1, .col-sm-1, .col-md-1, .col-lg-1, .col-xs-2, .col-sm-2, .col-md-2, .col-lg-2, .col-xs-3, .col-sm-3, .col-md-3, .col-lg-3, .col-xs-4, .col-sm-4, .col-md-4, .col-lg-4, .col-xs-5, .col-sm-5, .col-md-5, .col-lg-5, .col-xs-6, .col-sm-6, .col-md-6, .col-lg-6, .col-xs-7, .col-sm-7, .col-md-7, .col-lg-7, .col-xs-8, .col-sm-8, .col-md-8, .col-lg-8, .col-xs-9, .col-sm-9, .col-md-9, .col-lg-9, .col-xs-10, .col-sm-10, .col-md-10, .col-lg-10, .col-xs-11, .col-sm-11, .col-md-11, .col-lg-11, .col-xs-12, .col-sm-12, .col-md-12, .col-lg-12 { position: relative; min-height: 1px; padding-left: 15px; padding-right: 15px; } .col-xs-1, .col-xs-2, .col-xs-3, .col-xs-4, .col-xs-5, .col-xs-6, .col-xs-7, .col-xs-8, .col-xs-9, .col-xs-10, .col-xs-11, .col-xs-12 { float: left; } .col-xs-12 { width: 100%; } .col-xs-1 { width: 8.3333333%; } .col-xs-2 { width: 16.6666667%; } .col-xs-3 { width: 25%; } .col-xs-4 { width: 33.3333333%; } .col-xs-5 { width: 41.6666667%; } .col-xs-6 { width: 50%; } .col-xs-7 { width: 58.3333333%; } .col-xs-8 { width: 66.6666667%; } .col-xs-9 { width: 75%; } .col-xs-10 { width: 83.3333333%; } .col-xs-11 { width: 91.6666667%; } .col-xs-12 { width: 100%; } .col-xs-pull-12 { right: 100%; } .col-xs-pull-11 { right: 91.6666667%; } .col-xs-pull-10 { right: 83.3333333%; } .col-xs-pull-9 { right: 75%; } .col-xs-pull-8 { right: 66.6666667%; } .col-xs-pull-7 { right: 50%; } .col-xs-pull-6 { right: 41.6666667%; } .col-xs-pull-5 { right: 33.3333333%; } .col-xs-pull-4 { right: 25%; } .col-xs-pull-3 { right: 16.6666667%; } .col-xs-pull-2 { right: 8.3333333%; } .col-xs-pull-1 { right: 0%; } .col-xs-push-12 { left: 100%; } .col-xs-push-11 { left: 91.6666667%; } .col-xs-push-10 { left: 83.3333333%; } .col-xs-push-9 { left: 75%; } .col-xs-push-8 { left: 66.6666667%; } .col-xs-push-7 { left: 58.3333333%; } .col-xs-push-6 { left: 50%; } .col-xs-push-5 { left: 41.6666667%; } .col-xs-push-4 { left: 33.3333333%; } .col-xs-push-3 { left: 25%; } .col-xs-push-2 { left: 16.6666667%; } .col-xs-push-1 { left: 8.3333333%; } .col-xs-offset-12 { margin-left: 100%; } .col-xs-offset-11 { margin-left: 91.6666667%; } .col-xs-offset-10 { margin-left: 83.3333333%; } .col-xs-offset-9 { margin-left: 75%; } .col-xs-offset-8 { margin-left: 66.6666667%; } .col-xs-offset-7 { margin-left: 50%; } .col-xs-offset-6 { margin-left: 41.6666667%; } .col-xs-offset-5 { margin-left: 33.3333333%; } .col-xs-offset-4 { margin-left: 25%; } .col-xs-offset-3 { margin-left: 16.6666667%; } .col-xs-offset-2 { margin-left: 8.3333333%; } .col-xs-offset-1 { margin-left: 0%; } @media (min-width: 992px) { .col-sm-1, .col-sm-2, .col-sm-3, .col-sm-4, .col-sm-5, .col-sm-6, .col-sm-7, .col-sm-8, .col-sm-9, .col-sm-10, .col-sm-11, .col-sm-12 { float: left; } .col-sm-12 { width: 100%; } .col-sm-11 { width: 91.6666667%; } .col-sm-10 { width: 83.3333333%; } .col-sm-9 { width: 75%; } .col-sm-8 { width: 66.6666667%; } .col-sm-7 { width: 58.3333333%; } .col-sm-6 { width: 50%; } .col-sm-5 { width: 41.6666667%; } .col-sm-4 { width: 33.3333333%; } .col-sm-3 { width: 25%; } .col-sm-2 { width: 16.6666667%; } .col-sm-1 { width: 8.3333333%; } .col-sm-pull-12 { right: 100%; } .col-sm-pull-11 { right: 91.6666667%; } .col-sm-pull-10 { right: 83.3333333%; } .col-sm-pull-9 { right: 75%; } .col-sm-pull-8 { right: 66.6666667%; } .col-sm-pull-7 { right: 58.3333333%; } .col-sm-pull-6 { right: 50%; } .col-sm-pull-5 { right: 41.6666667%; } .col-sm-pull-4 { right: 33.3333333%; } .col-sm-pull-3 { right: 25%; } .col-sm-pull-2 { right: 16.6666667%; } .col-sm-pull-1 { right: 8.3333333%; } .col-sm-push-12 { left: 100%; } .col-sm-push-11 { left: 91.6666667%; } .col-sm-push-10 { left: 83.3333333%; } .col-sm-push-9 { left: 75%; } .col-sm-push-8 { left: 66.6666667%; } .col-sm-push-7 { left: 58.3333333%; } .col-sm-push-6 { left: 50%; } .col-sm-push-5 { left: 41.6666667%; } .col-sm-push-4 { left: 33.3333333%; } .col-sm-push-3 { left: 25%; } .col-sm-push-2 { left: 16.6666667%; } .col-sm-push-1 { left: 8.3333333%; } .col-sm-offset-12 { margin-left: 100%; } .col-sm-offset-11 { margin-left: 91.6666667%; } .col-sm-offset-10 { margin-left: 83.3333333%; } .col-sm-offset-9 { margin-left: 75%; } .col-sm-offset-8 { margin-left: 66.6666667%; } .col-sm-offset-7 { margin-left: 50%; } .col-sm-offset-6 { margin-left: 41.6666667%; } .col-sm-offset-5 { margin-left: 33.3333333%; } .col-sm-offset-4 { margin-left: 25%; } .col-sm-offset-3 { margin-left: 16.6666667%; } .col-sm-offset-2 { margin-left: 8.3333333%; } .col-sm-offset-1 { margin-left: 0%; } } @media (min-width: 1200px) { .col-md-1, .col-md-2, .col-md-3, .col-md-4, .col-md-5, .col-md-6, .col-md-7, .col-md-8, .col-md-9, .col-md-10, .col-md-11, .col-md-12 { float: left; } .col-md-12 { width: 100%; } .col-md-11 { width: 91.6666667%; } .col-md-10 { width: 83.3333333%; } .col-md-9 { width: 75%; } .col-md-8 { width: 66.6666667%; } .col-md-7 { width: 58.3333333%; } .col-md-6 { width: 50%; } .col-md-5 { width: 41.6666667%; } .col-md-4 { width: 33.3333333%; } .col-md-3 { width: 25%; } .col-md-2 { width: 16.6666667%; } .col-md-1 { width: 8.3333333%; } .col-md-pull-12 { right: 100%; } .col-md-pull-11 { right: 91.6666667%; } .col-md-pull-10 { right: 83.3333333%; } .col-md-pull-9 { right: 75%; } .col-md-pull-8 { right: 66.6666667%; } .col-md-pull-7 { right: 58.3333333%; } .col-md-pull-6 { right: 50%; } .col-md-pull-5 { right: 41.6666667%; } .col-md-pull-4 { right: 33.3333333%; } .col-md-pull-3 { right: 25%; } .col-md-pull-2 { right: 16.6666667%; } .col-md-pull-1 { right: 8.3333333%; } .col-md-push-12 { left: 100%; } .col-md-push-11 { left: 91.6666667%; } .col-md-push-10 { left: 83.3333333%; } .col-md-push-9 { left: 75%; } .col-md-push-8 { left: 66.6666667%; } .col-md-push-7 { left: 58.3333333%; } .col-md-push-6 { left: 50%; } .col-md-push-5 { left: 41.6666667%; } .col-md-push-4 { left: 33.3333333%; } .col-md-push-3 { left: 25%; } .col-md-push-2 { left: 16.6666667%; } .col-md-push-1 { left: 8.3333333%; } .col-md-offset-12 { margin-left: 100%; } .col-md-offset-11 { margin-left: 91.6666667%; } .col-md-offset-10 { margin-left: 83.3333333%; } .col-md-offset-9 { margin-left: 75%; } .col-md-offset-8 { margin-left: 66.6666667%; } .col-md-offset-7 { margin-left: 50%; } .col-md-offset-6 { margin-left: 41.6666667%; } .col-md-offset-5 { margin-left: 33.3333333%; } .col-md-offset-4 { margin-left: 25%; } .col-md-offset-3 { margin-left: 16.6666667%; } .col-md-offset-2 { margin-left: 8.3333333%; } .col-md-offset-1 { margin-left: 0%; } } @media (min-width: 1400px) { .col-lg-1, .col-lg-2, .col-lg-3, .col-lg-4, .col-lg-5, .col-lg-6, .col-lg-7, .col-lg-8, .col-lg-9, .col-lg-10, .col-lg-11, .col-lg-12 { float: left; } .col-lg-12 { width: 100%; } .col-lg-11 { width: 91.6666667%; } .col-lg-10 { width: 83.3333333%; } .col-lg-9 { width: 75%; } .col-lg-8 { width: 66.6666667%; } .col-lg-7 { width: 58.3333333%; } .col-lg-6 { width: 50%; } .col-lg-5 { width: 41.6666667%; } .col-lg-4 { width: 33.3333333%; } .col-lg-3 { width: 25%; } .col-lg-2 { width: 16.6666667%; } .col-lg-1 { width: 8.3333333%; } .col-lg-pull-12 { right: 100%; } .col-lg-pull-11 { right: 91.6666667%; } .col-lg-pull-10 { right: 83.3333333%; } .col-lg-pull-9 { right: 75%; } .col-lg-pull-8 { right: 66.6666667%; } .col-lg-pull-7 { right: 58.3333333%; } .col-lg-pull-6 { right: 50%; } .col-lg-pull-5 { right: 41.6666667%; } .col-lg-pull-4 { right: 33.3333333%; } .col-lg-pull-3 { right: 25%; } .col-lg-pull-2 { right: 16.6666667%; } .col-lg-pull-1 { right: 8.3333333%; } .col-lg-push-12 { left: 100%; } .col-lg-push-11 { left: 91.6666667%; } .col-lg-push-10 { left: 83.3333333%; } .col-lg-push-9 { left: 75%; } .col-lg-push-8 { left: 66.6666667%; } .col-lg-push-7 { left: 58.3333333%; } .col-lg-push-6 { left: 50%; } .col-lg-push-5 { left: 41.6666667%; } .col-lg-push-4 { left: 33.3333333%; } .col-lg-push-3 { left: 25%; } .col-lg-push-2 { left: 16.6666667%; } .col-lg-push-1 { left: 8.3333333%; } .col-lg-offset-12 { margin-left: 100%; } .col-lg-offset-11 { margin-left: 91.6666667%; } .col-lg-offset-10 { margin-left: 83.3333333%; } .col-lg-offset-9 { margin-left: 75%; } .col-lg-offset-8 { margin-left: 66.6666667%; } .col-lg-offset-7 { margin-left: 50%; } .col-lg-offset-6 { margin-left: 41.6666667%; } .col-lg-offset-5 { margin-left: 33.3333333%; } .col-lg-offset-4 { margin-left: 25%; } .col-lg-offset-3 { margin-left: 16.6666667%; } .col-lg-offset-2 { margin-left: 8.3333333%; } .col-lg-offset-1 { margin-left: 0%; } } .clear-fix: before, .clear-fix: after, .container: before, .container: after, .container-fluid: before, .row: before, .row: after { content: " "; display: table; clear: both; } .center-block { display: block; margin-left: auto; margin-right: auto; } .pull-right { float: right !important; } .pull-left { float: left !important; } .show { display: block !important; } .in-visible { visibility: hidden; } .text-hide { font: 0/0 a; color: transparent; text-shadow: none; background-color: transparent; border: 0; } .hidden { display: none !important; } .affix { position: fixed; }
```

## grid layout (5 lignes)

```
.container {  
    display: grid;  
    grid-template-columns: repeat(12, 1fr);  
    grid-gap: 15px;  
}  
  
aside { grid-column: span 3; }  
main { grid-column: span 9; }
```

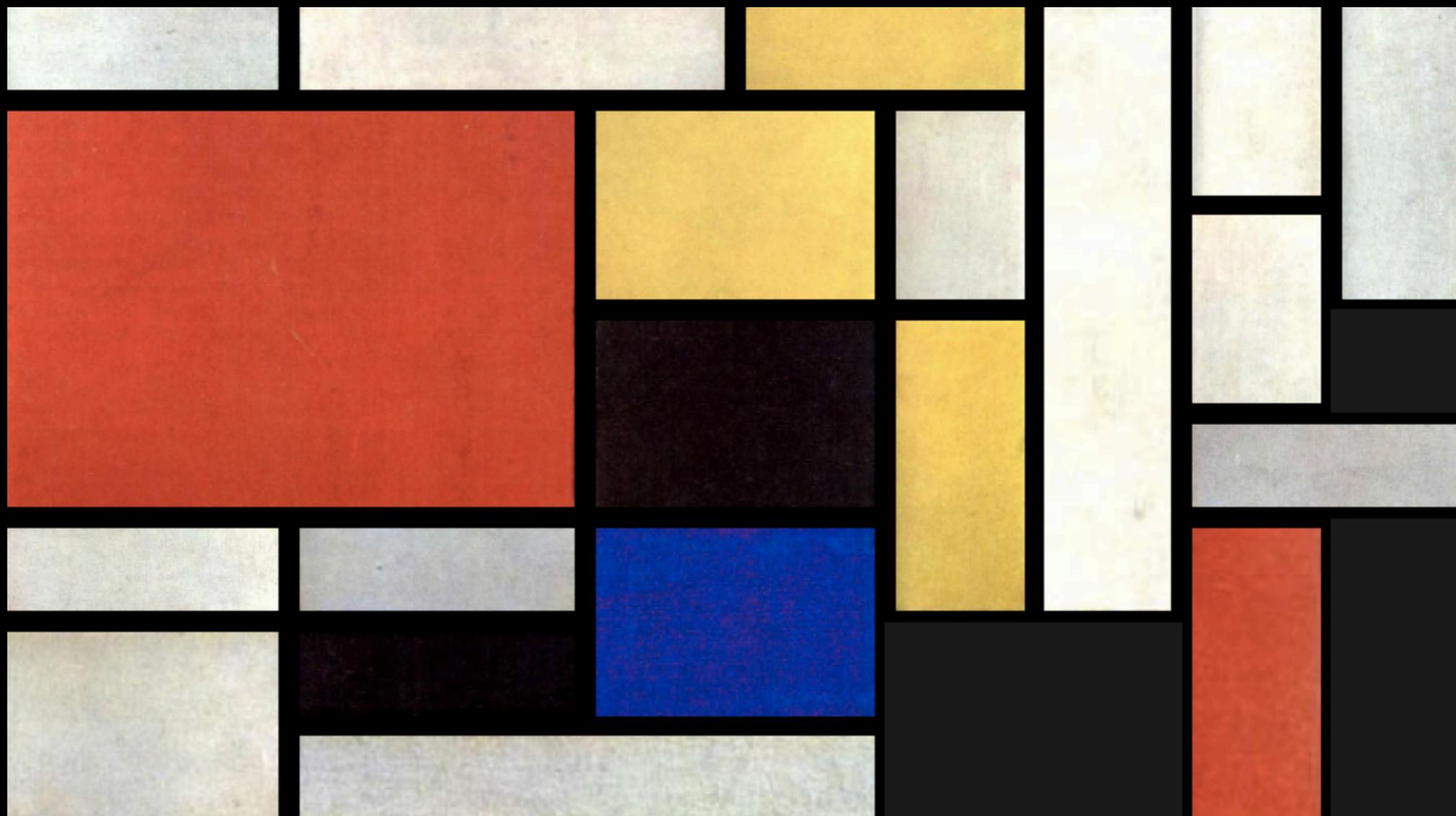


# OK, t'as des exemples ?

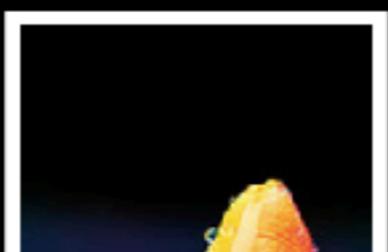


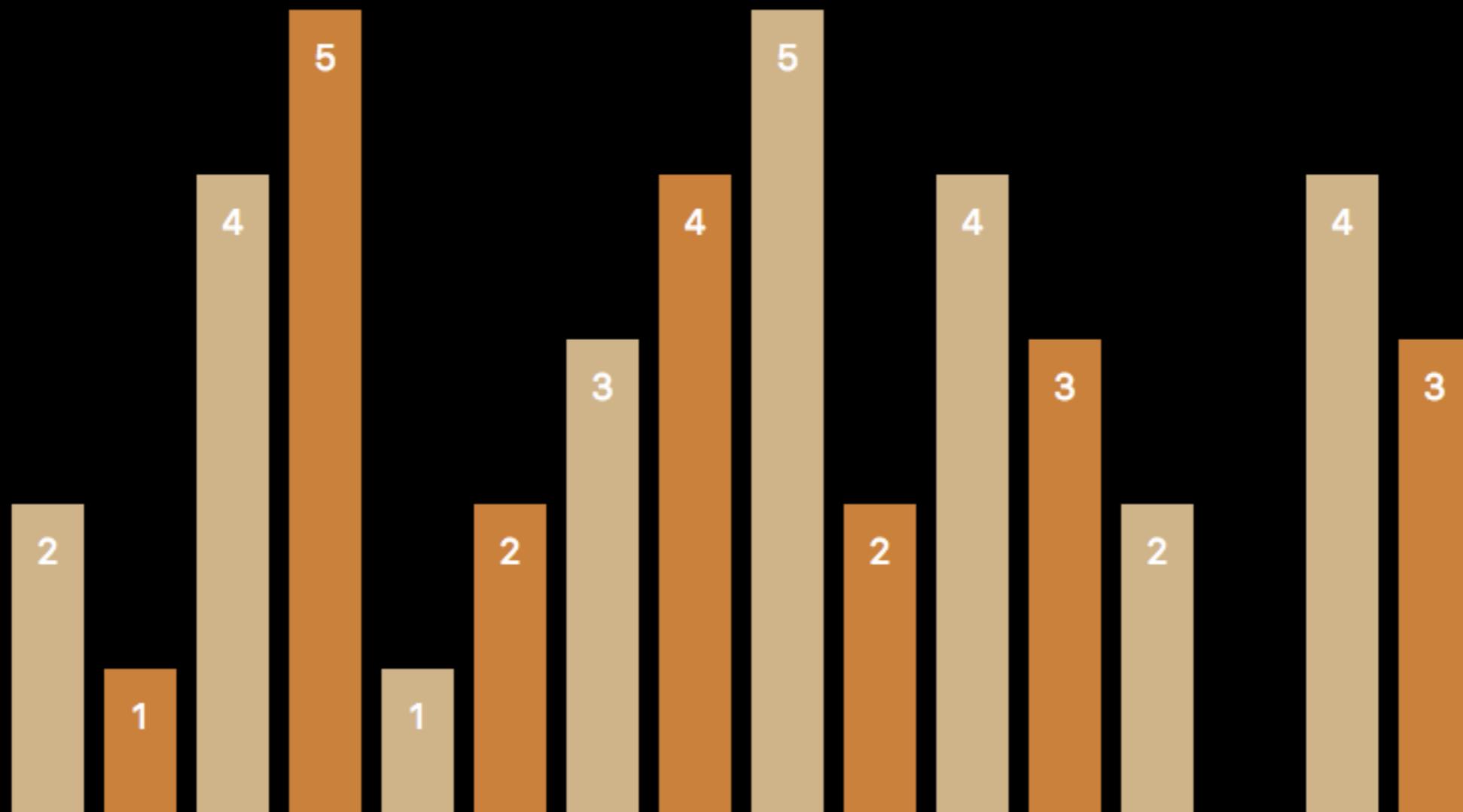


<http://codepen.io/raphaelgoetter/pen/xwaoOP?editors=1100>



<http://labs.jensimmons.com/2016/examples/mondrian-2.html>





<https://codepen.io/raphaelgoetter/pen/aqzOxg>

Lorem Elsass Ipsum mitt picon bière

E gewurtztraminer doch Carola schneck, schmutz a riesling  
de chambon eme rucksack Roger dû hopla geiss,  
jetz Chorquette de Scharrbergheim. Kouglopf ech ich  
wurscht gueut mitt schneck  
jetz a schiss mannelé, knèkes saucisse de  
Niederhausbergen of fill mauls

schéni fleischwurcht schnaps de eme gal nüdle blottkopf,  
de Chulien Roger hop pfourtz! bett mer ech schpeck un  
salami schmutz. Gal!

munster du ftomi! Ponchour bisame. Bibbeleskaas jetz  
rassballa sech choucroute un schwanz geburtstag,  
Chinette dû, ich bier deppfels schiesser. Flammekueche de  
knèkes Seppele gall a hopla geburtstag,  
alles frau Chulia Roberts oder knäckes dû blottkopf. Noch  
bredele schissabibala, yeuh e schmutz.

hopla !



<https://codepen.io/Kseso/full/xqNdmO/>

# Boarding pass

Flight n°  
VY1812      Gate  
D26

Munich  
**MUC**

to  
**BCN**

Passenger      Seat  
Victor Janin    23A

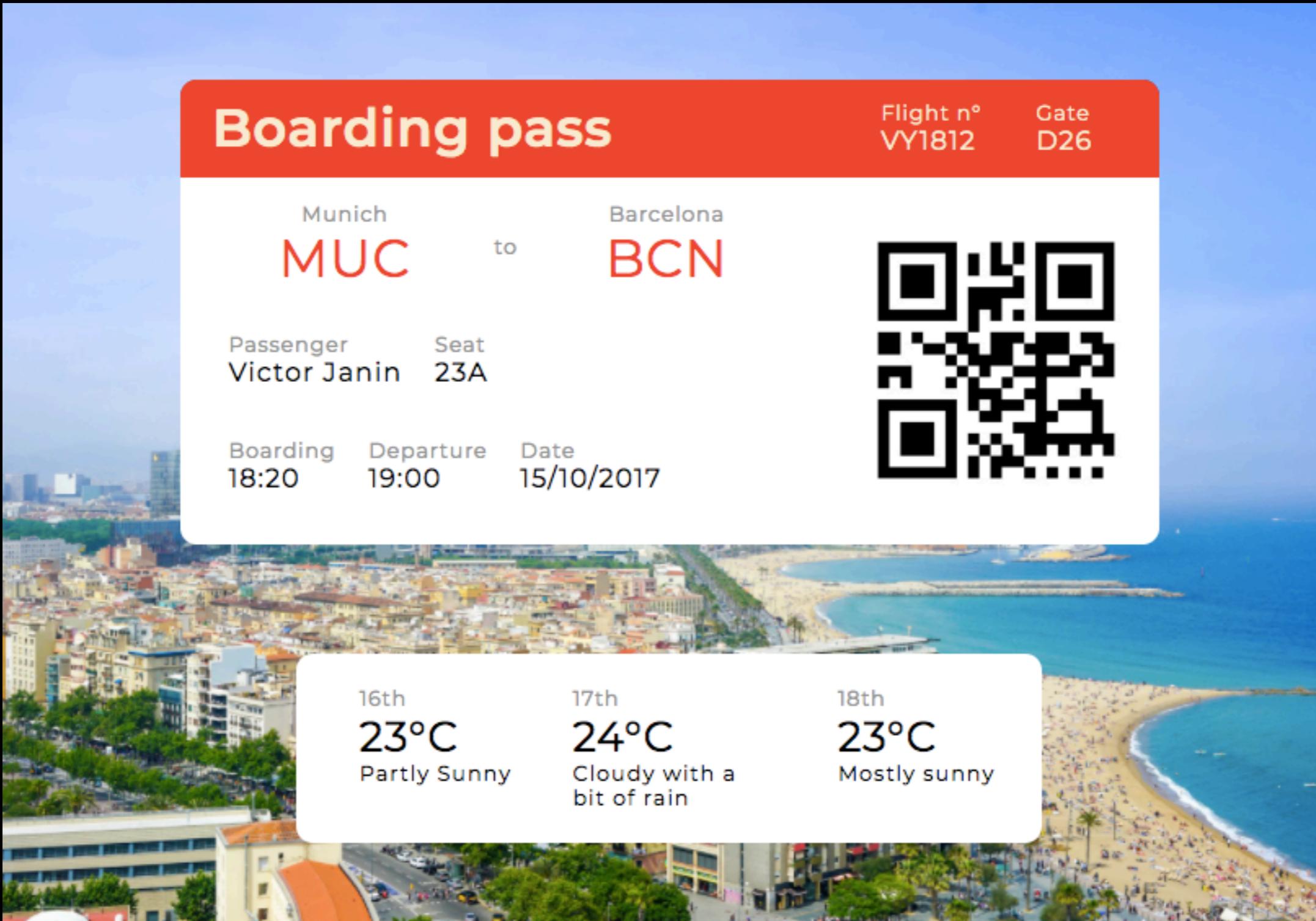
Boarding      Departure      Date  
18:20            19:00            15/10/2017



16th  
**23°C**  
Partly Sunny

17th  
**24°C**  
Cloudy with a  
bit of rain

18th  
**23°C**  
Mostly sunny



<https://codepen.io/victorjanin/pen/oGYvjk>

A|NDY



BAR|EFOOT

<https://codepen.io/andybarefoot/pen/GMvygq>

# glossaire

pour bien comprendre des trucs ensuite



# grid container

élément parent conteneur de la grille

```
<section class="grid-container">  
  <div>Salade</div>  
  <div>Tomate</div>  
  <div>Oignon</div>  
  <div>Picon bière</div>  
</section>
```

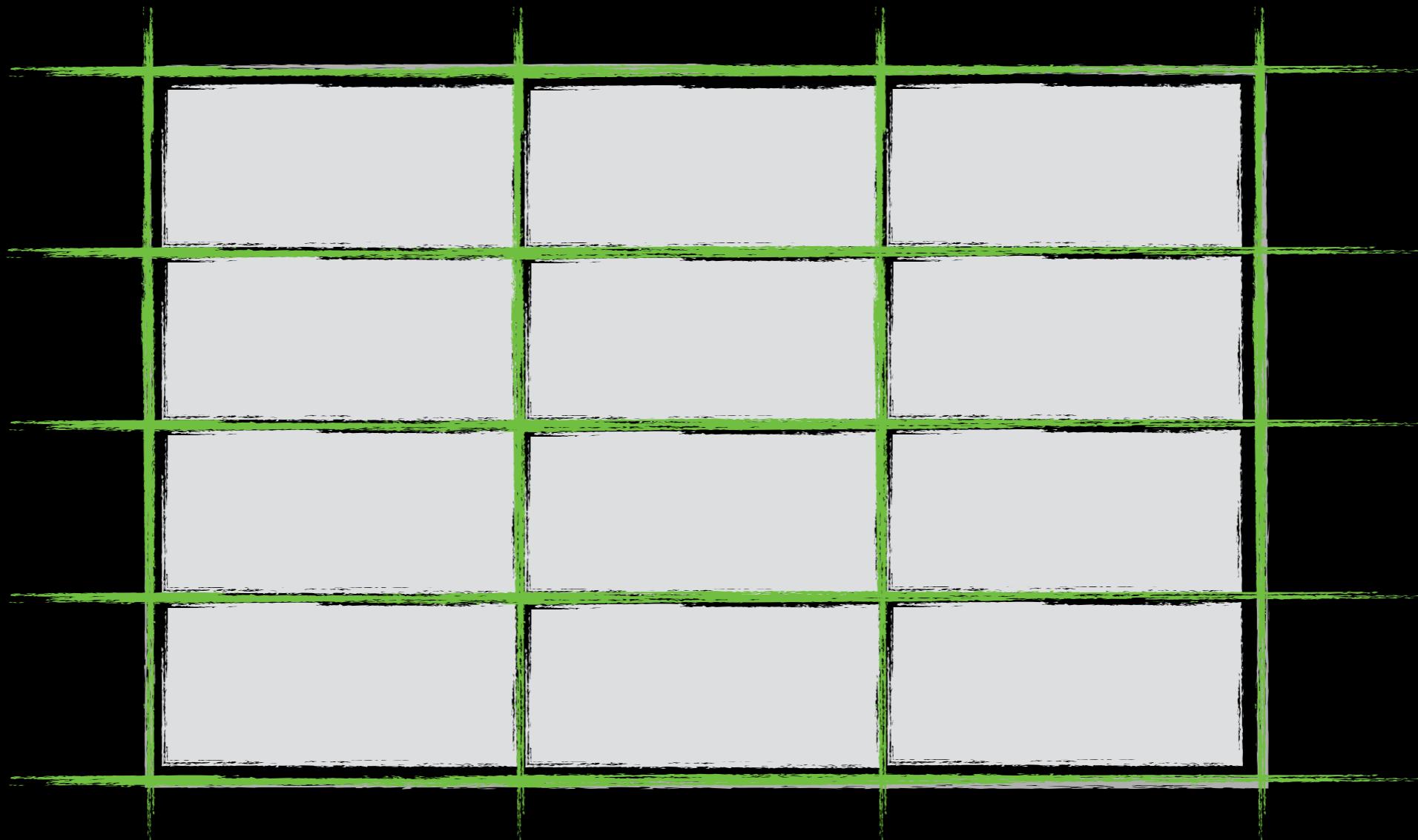
# grid item

enfant direct d'un grid container

```
<section class="grid-container">  
  <div>Salade</div>  
  <div>Tomate</div>  
  <div>Oignon</div>  
  <div>Picon bière</div>  
</section>
```

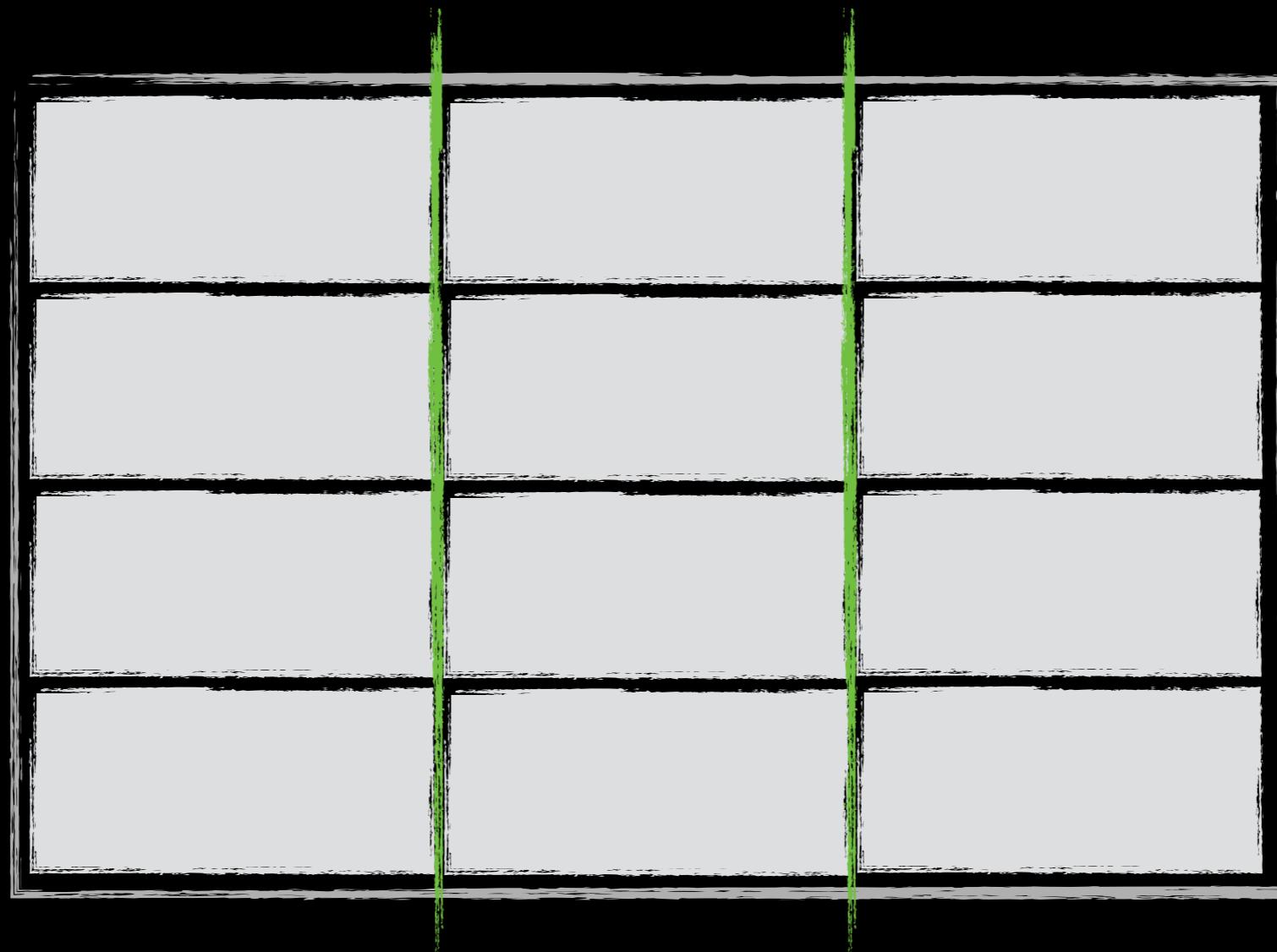
# grid line

ligne divisant la grille



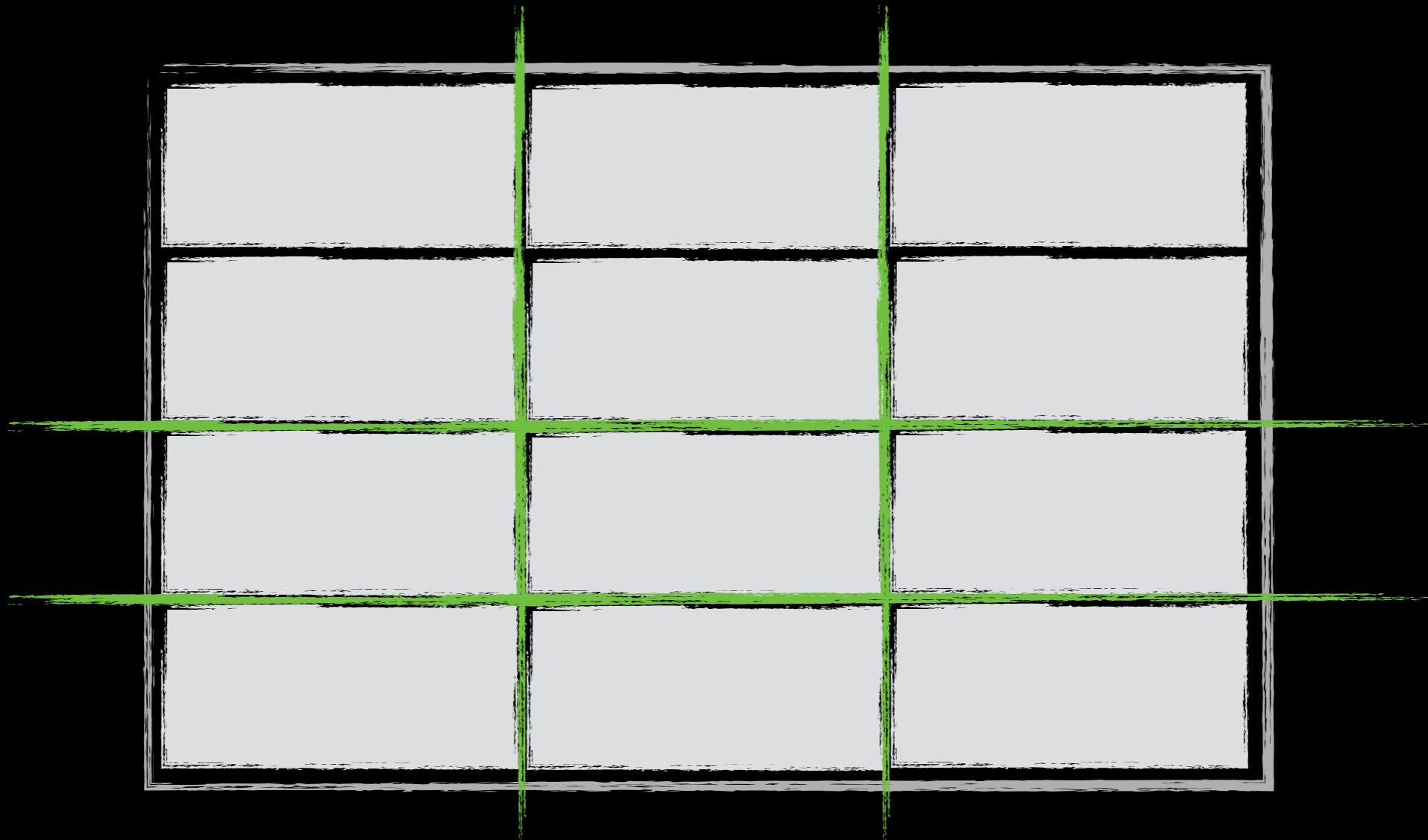
# grid track

une colonne ou une rangée  
*(espace entre deux grid lines)*



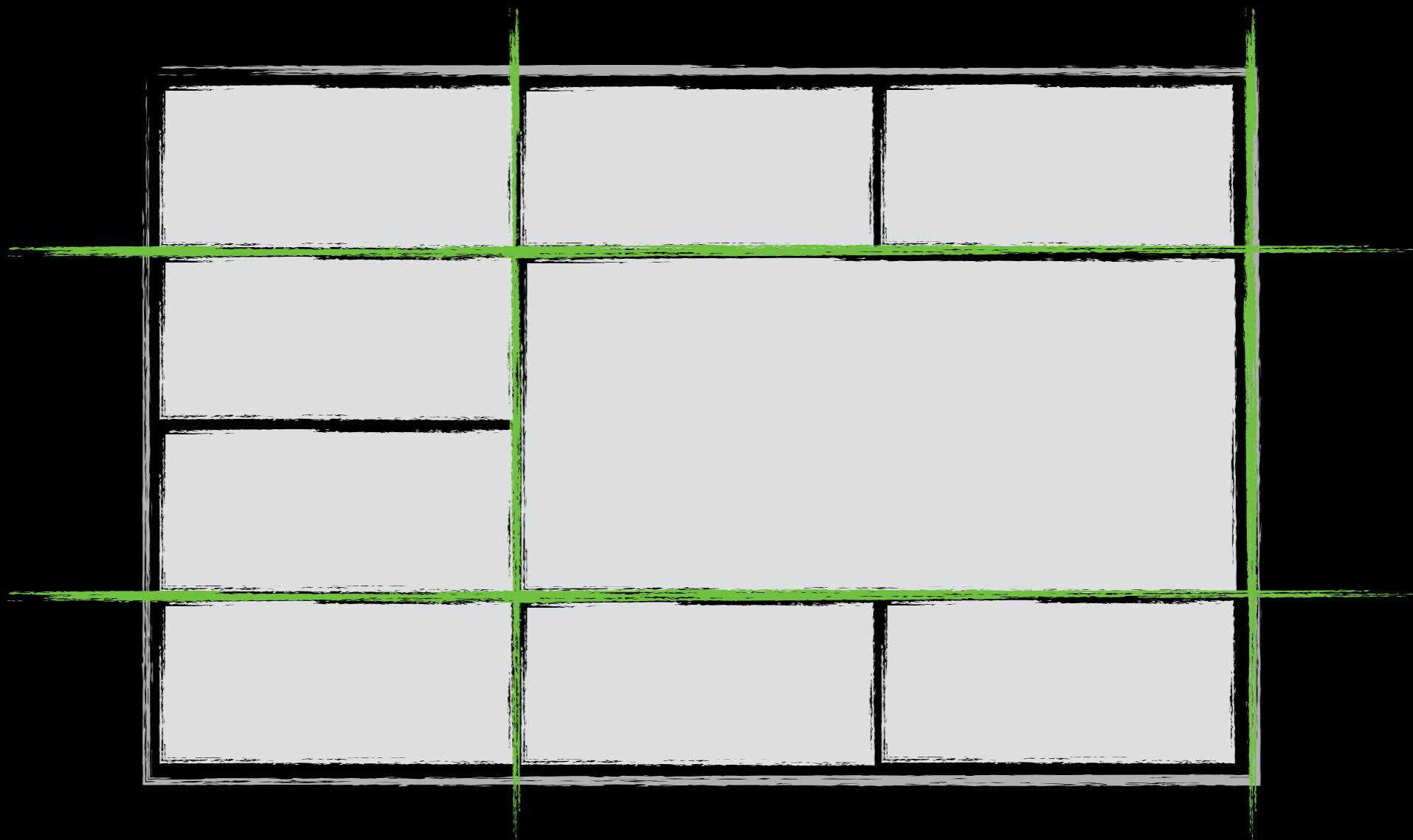
# grid cell

intersection entre deux grid tracks  
*(unité de positionnement dans la grille)*



# grid area

zone d'une ou plusieurs cellules  
*(une ou plusieurs grid cells)*

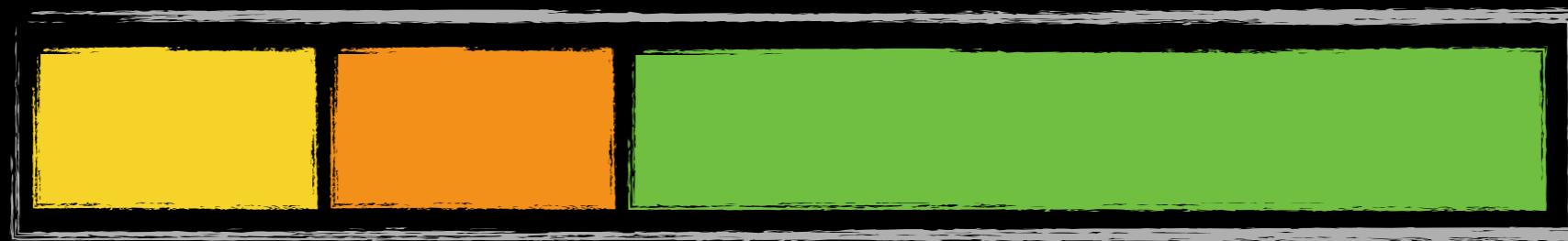




# unités et valeurs

# unité « fr »

fraction de l'espace disponible



```
.grid-container {  
    display: grid;  
    grid-template-columns: 100px 100px 1fr;  
}
```

# unité « fr »

fraction de l'espace disponible



```
.grid-container {  
    display: grid;  
    grid-template-columns: 100px 1fr 1fr;  
}
```

# unité « fr »

fraction de l'espace disponible



```
.grid-container {  
    display: grid;  
    grid-template-columns: 100px 2fr 1fr;  
}
```

# valeur « auto »

taille du contenu



```
.grid-container {  
  display: grid;  
  grid-template-rows: auto 1fr;  
}
```

# minmax()

j'aime bien la  
salade

j'aime bien la salade

j'aime bien la salade

```
.grid-container {  
    display: grid;  
    grid-template-columns: minmax(100px, 20%) 1fr 1fr;  
}
```

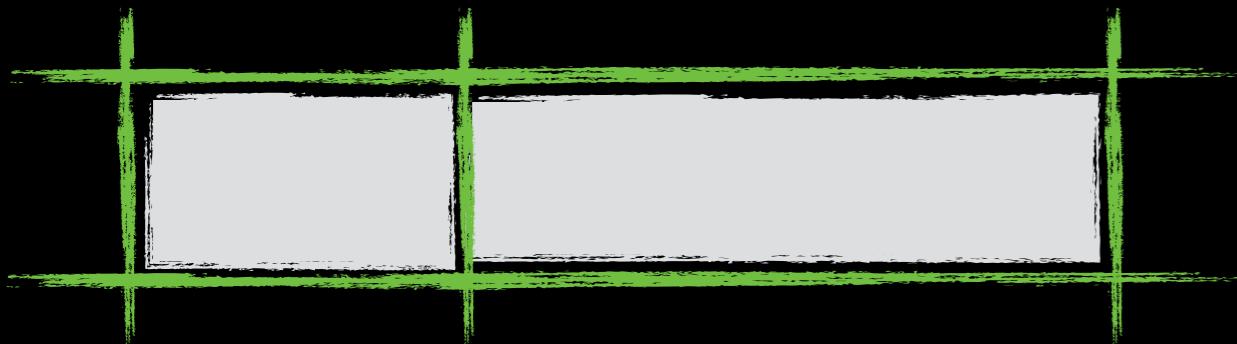


# construction de grilles

# initialisation

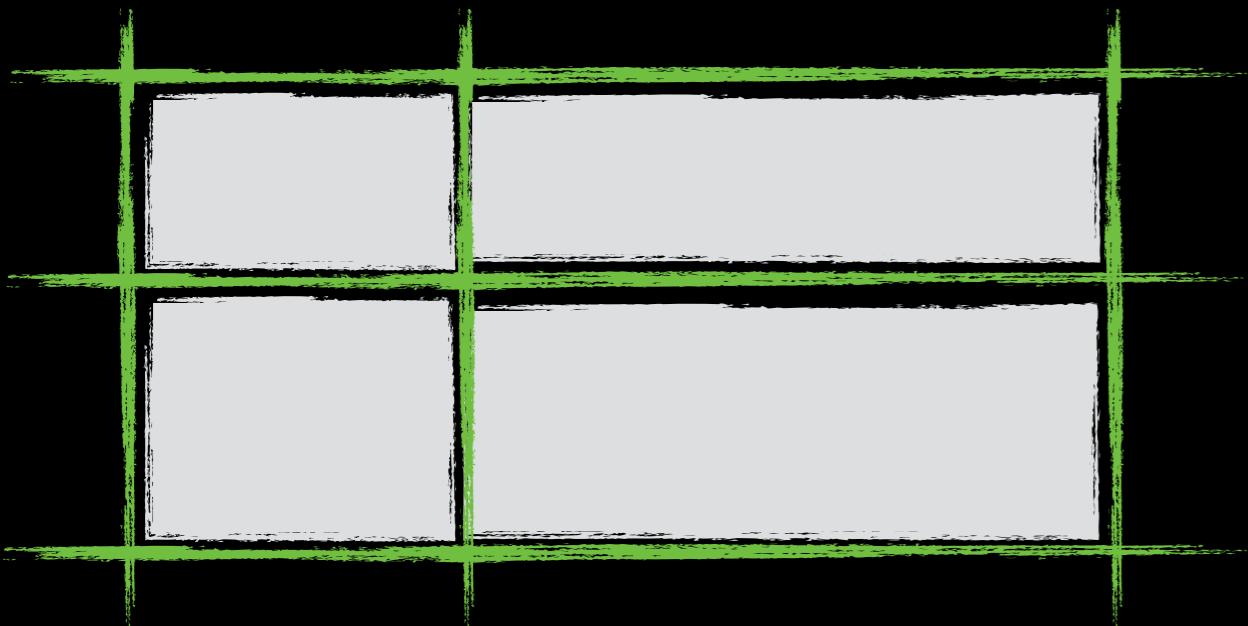
```
.ingrid {  
    display: grid;  
}
```

# colonnes



```
.ingrid {  
  display: grid;  
  grid-template-columns: 100px 1fr;  
}
```

# lignes



```
.ingrid {  
  display: grid;  
  grid-template-columns: 100px 1fr;  
  grid-template-rows: 50px 1fr;  
}
```

# positionnement



# positionnement « auto »

*bah ça marche tout seul, gars*

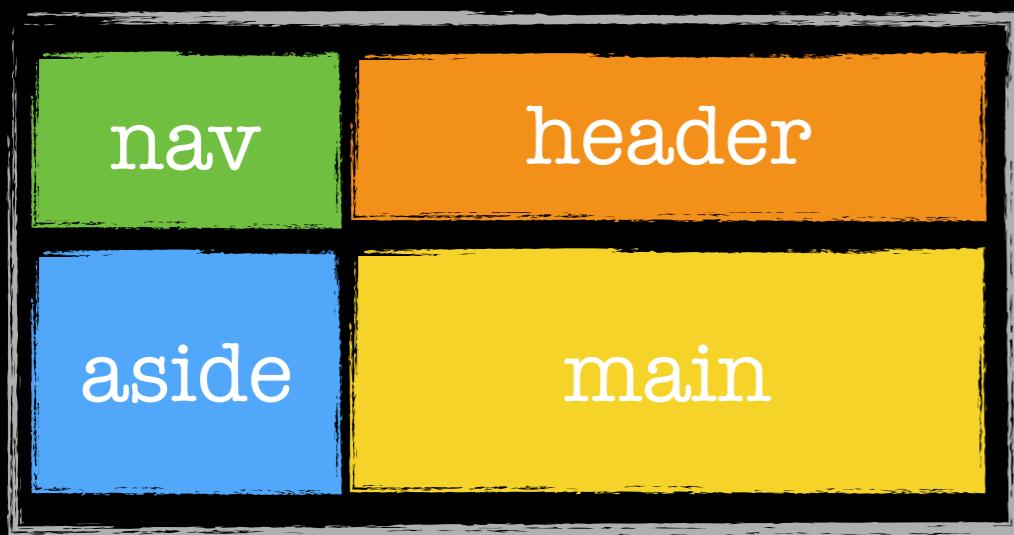
# positionnement explicite

*on place avec grid-column et grid-row*

# positionnement par « area »

*on place avec grid-area*

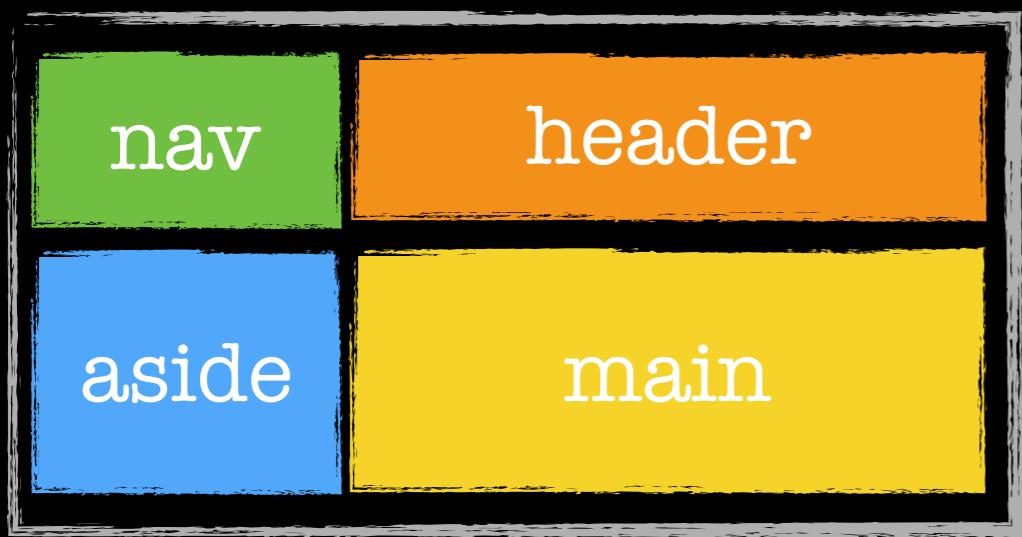
# positionnement « auto »



```
.hagrid {  
    display: grid;  
    grid-template-columns: 100px 1fr;  
}
```

```
<nav>nav</nav>  
<header>header</header>  
<aside>aside</aside>  
<main>main</main>
```

# positionnement explicite

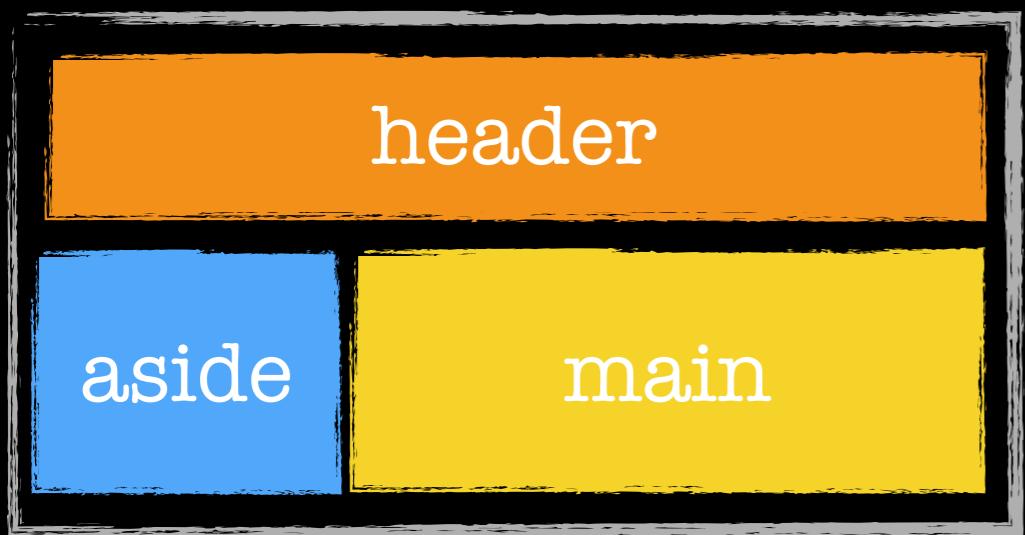


```
<nav>nav</nav>  
<header>header</header>  
<aside>aside</aside>  
<main>main</main>
```

```
.hagrid {  
    display: grid;  
    grid-template-columns: 100px 1fr;  
}  
  
nav {  
    grid-column: 1;  
    grid-row: 1;  
}  
  
main {  
    grid-column: 2;  
    grid-row: 2;  
}
```

 raccourci de :  
grid-column-start: 1;  
grid-column-end: 2;  
grid-row-start: 1;  
grid-row-end: 2;

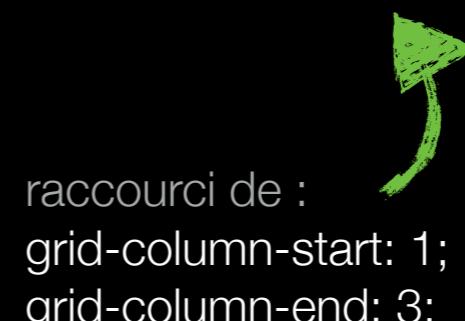
# positionnement explicite



```
<header>header</header>  
<aside>aside</aside>  
<main>main</main>
```

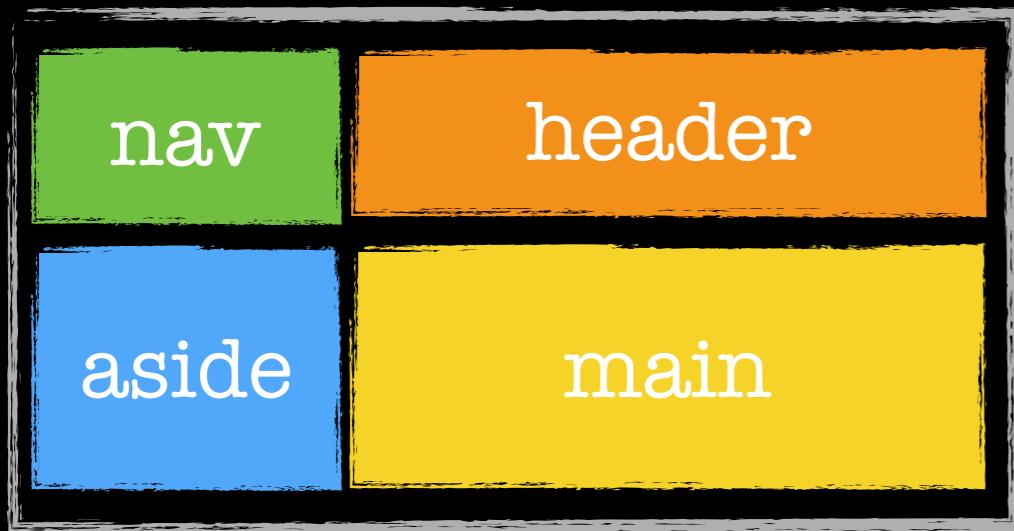
```
.hagrid {  
    display: grid;  
    grid-template-columns: 100px 1fr;  
}
```

```
header {  
    grid-column: 1 / 3;  
}
```



raccourci de :  
grid-column-start: 1;  
grid-column-end: 3;

# positionnement par « area »

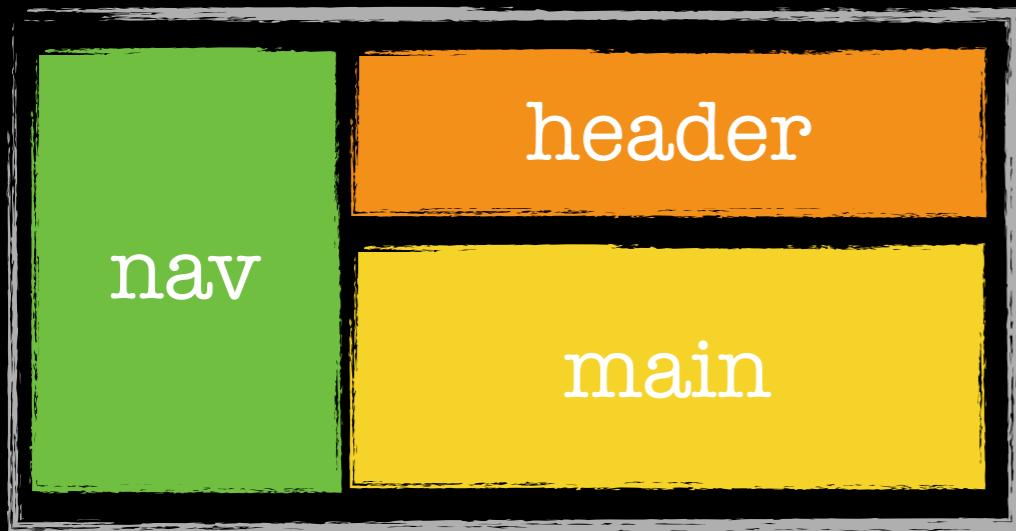


```
.hagrid {  
    display: grid;  
    grid-template-columns: 100px 1fr;  
    grid-template-areas: "a b"  
                        "c d";  
}
```

```
<nav>nav</nav>  
<header>header</header>  
<aside>aside</aside>  
<main>main</main>
```

```
nav { grid-area: a; }  
header { grid-area: b; }  
aside { grid-area: c; }  
main { grid-area: d; }
```

# positionnement par « area »



```
<nav>nav</nav>  
<header>header</header>  
<main>main</main>
```

```
.hagrid {  
    display: grid;  
    grid-template-columns: 100px 1fr;  
    grid-template-areas: "a b"  
                        "a c";  
}
```

```
nav { grid-area: a; }  
header { grid-area: b; }  
main { grid-area: c; }
```

# alignements



# alignement horizontal

**justify-content**      alignement horizontal des grid-cells

**justify-items**      alignement horizontal des grid-items

**justify-self**      alignement individuel d'un grid-item

# alignement vertical

**align-content**      alignement vertical des grid-cells

**align-items**      alignement vertical des grid-items

**align-self**      alignement individuel d'un grid-item

# **justify-content**

alignement horizontal des grid cells

# **align-content**

alignement vertical des grid cells

# justify-content

Salade

Tomate

justify-content: normal;

Salade

Tomate

justify-content: start;

Salade

Tomate

justify-content: end;

Salade

Tomate

justify-content: center;

Salade

Tomate

justify-content: space-between;

# **justify-items**

alignement horizontal des grid items

# **align-items**

alignement vertical des grid items

# justify-items

Salade

Tomate

justify-items: normal;

Salade

Tomate

justify-items: start;

Salade

Tomate

justify-items: end;

Salade

Tomate

justify-items: center;

Salade

Tomate

justify-items: space-between;

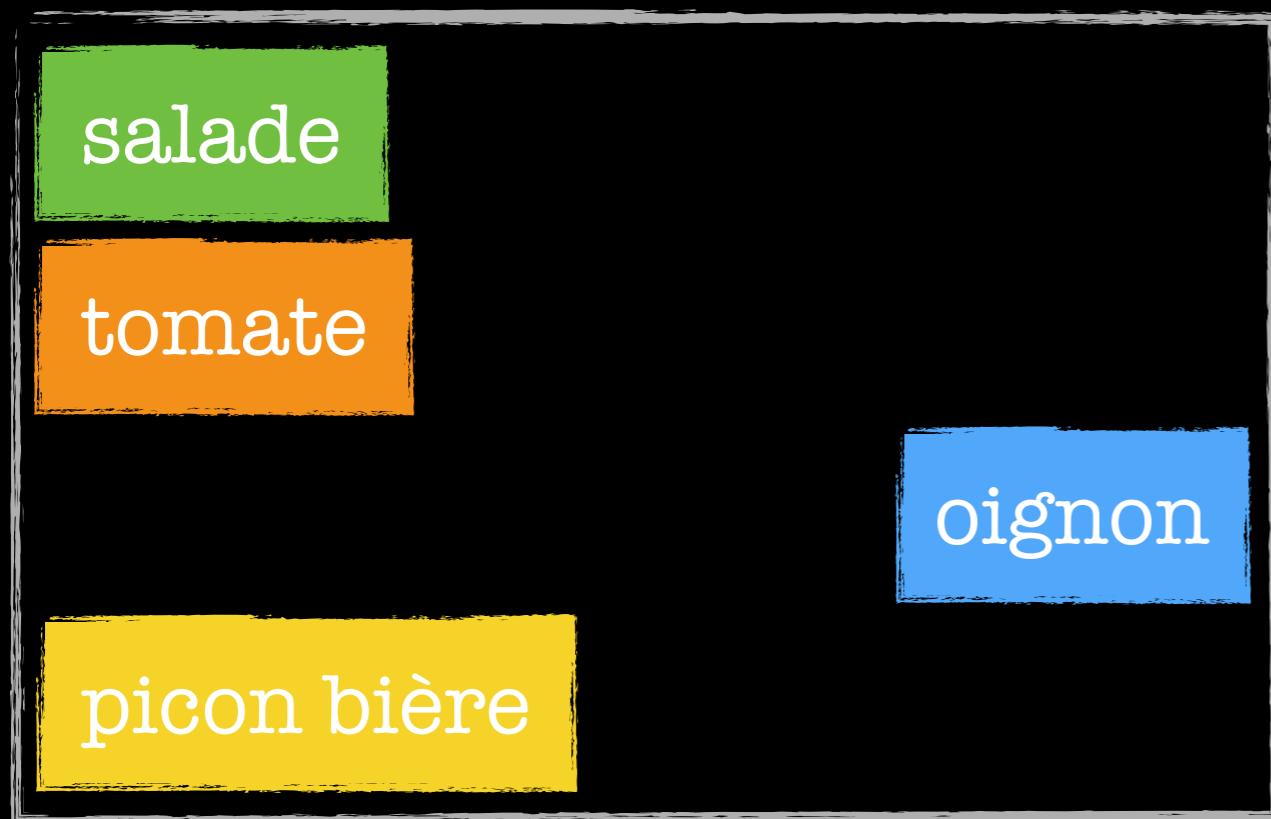
# **justify-self**

alignement horizontal d'un grid item

# **align-self**

alignement vertical d'un grid item

# justify-self



```
.grid-container {  
  display: grid;  
  justify-items: start;  
}  
.oignon {  
  justify-self: end;  
}
```

# en plein dans le mille !

salade

tomate

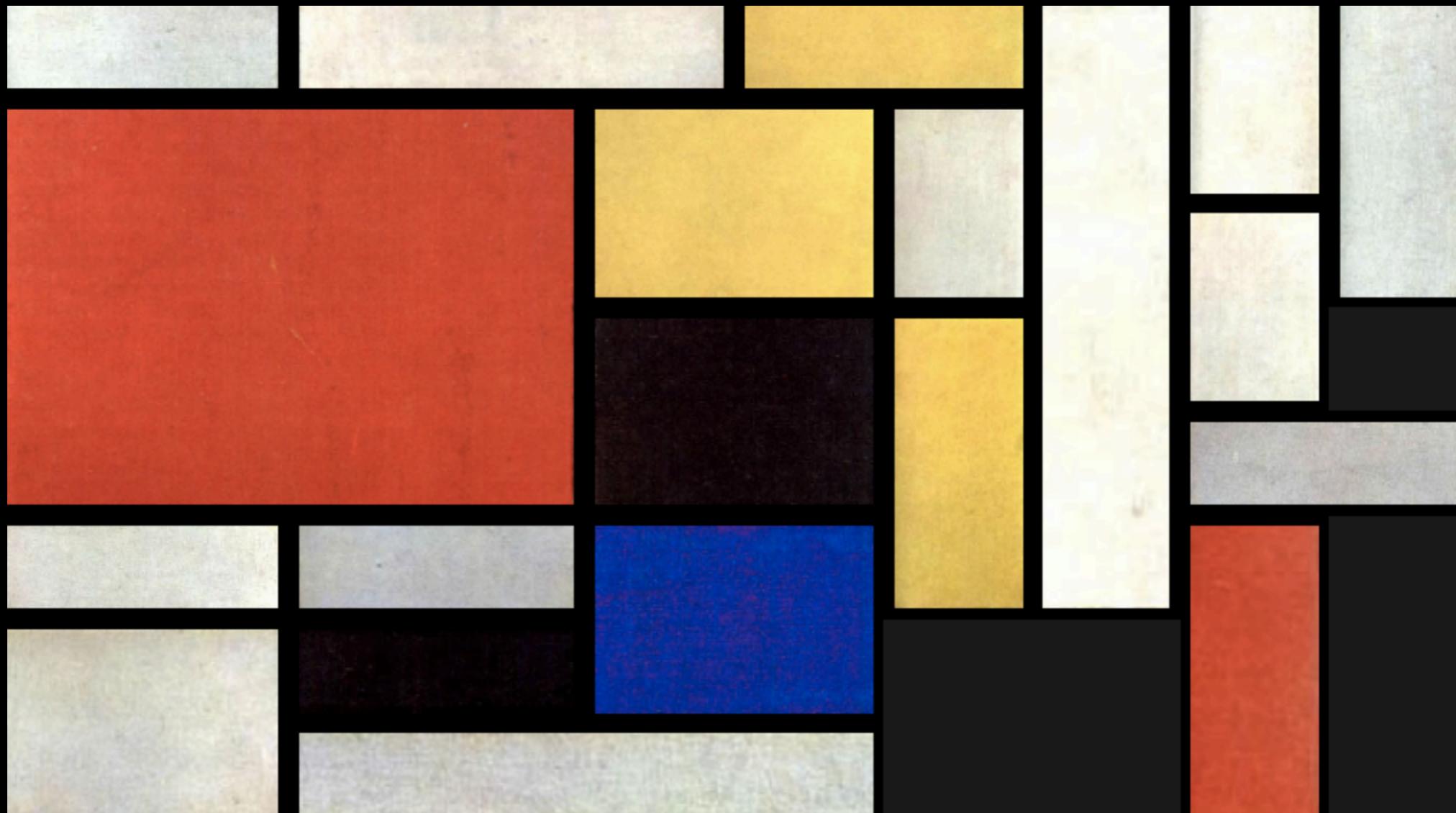
oignon

picon bière

```
.grid-container {  
    display: grid;  
    justify-content: center;  
    align-content: center;
```



# ouais, Mondrian-Grid !



<http://labs.jensimmons.com/2016/examples/mondrian-2.html>

# un template responsive ?

# chiche !

header

nav

title

article 1

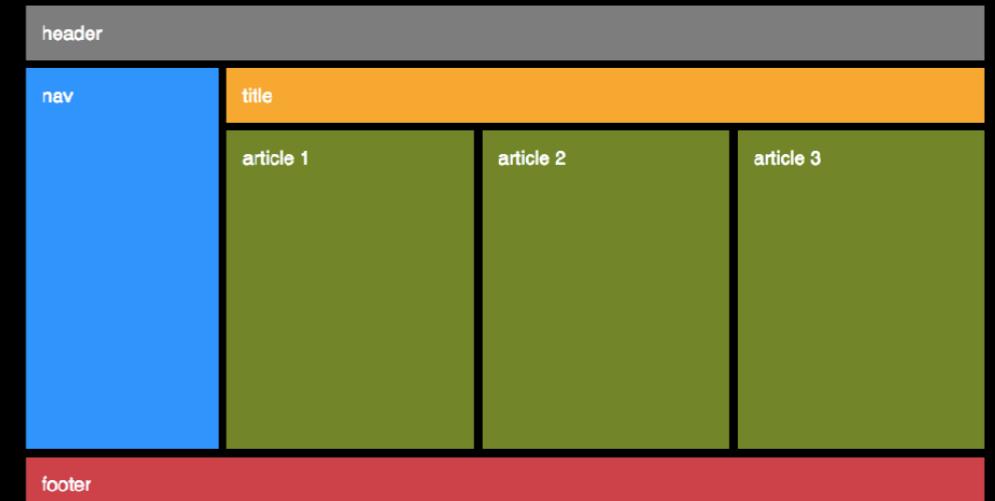
article 2

article 3

footer

# HTML :

```
<div class="container">  
  <header>header</header>  
  <nav>nav</nav>  
  <h1 class="title">title</h1>  
  <article>article 1</article>  
  <article>article 2</article>  
  <article>article 3</article>  
  <footer>footer</footer>  
</div>
```



# placement :

```
header { grid-area: header; }  
nav { grid-area: nav; }  
.title { grid-area: title; }  
article { /* rien */ }  
footer { grid-area: footer; }
```

# CSS mobile first !

```
.container {  
    display: grid;  
    grid-template-columns: 1fr;  
    grid-template-areas: "nav"  
                        "header"  
                        "title"  
                        ". "  
                        ". "  
                        ". "  
                        "footer";  
}
```



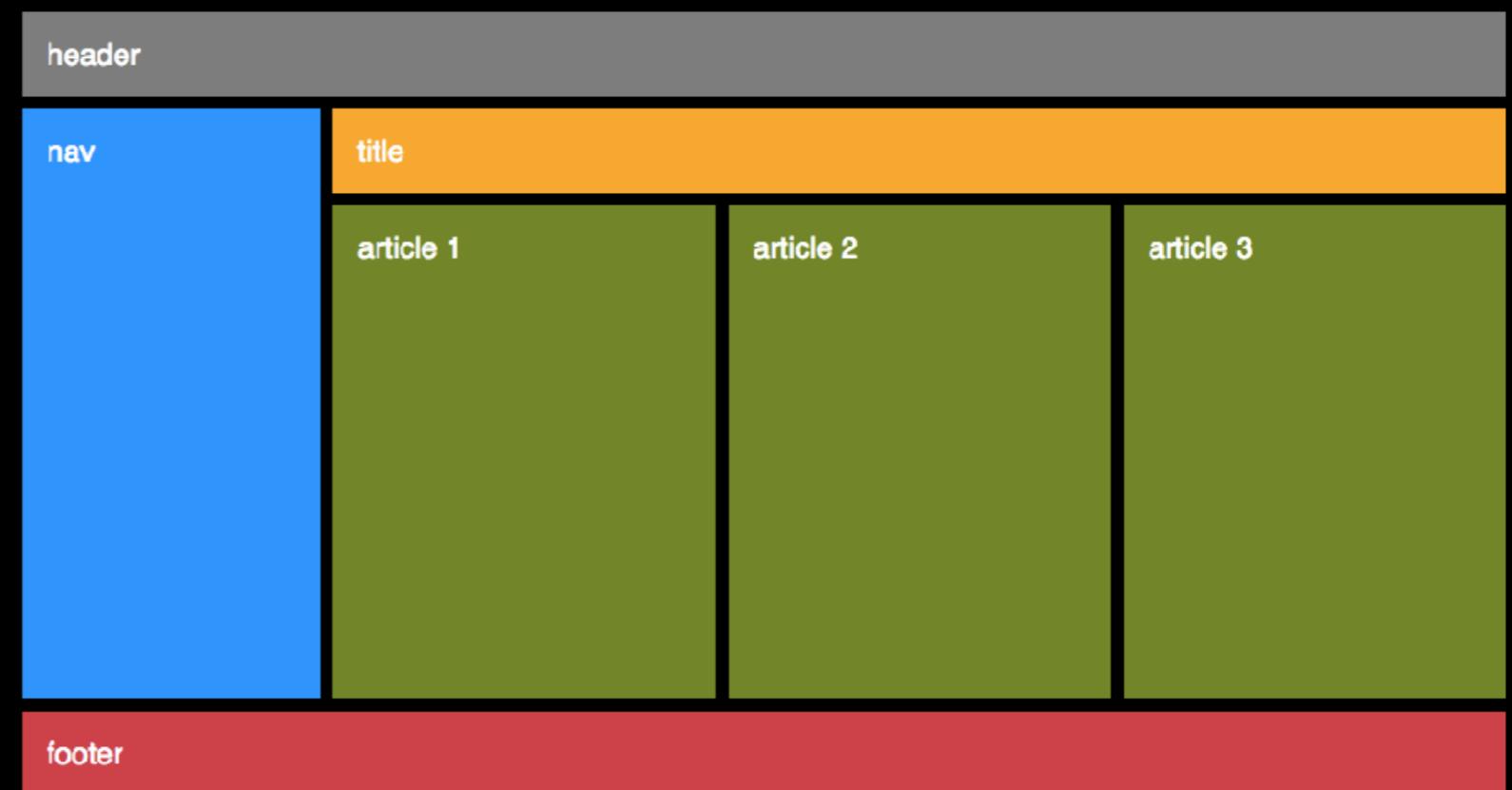
écrans de taille moyenne :

```
@media (min-width: 544px) {  
  .container {  
    grid-template-columns: 10em 1fr;  
    grid-template-areas: "header header"  
                         "title title"  
                         "nav   ."  
                         "nav   ."  
                         "nav   ."  
                         "footer footer";  
  }  
}
```



grands écrans :

```
@media (min-width: 992px) {  
  .container {  
    grid-template-columns: 10em 1fr 1fr 1fr;  
    grid-template-rows: auto auto 1fr auto;  
    grid-template-areas: "header header header header"  
                        "nav title title title"  
                        "nav . . ."  
                        "footer footer footer footer";  
  }  
}
```



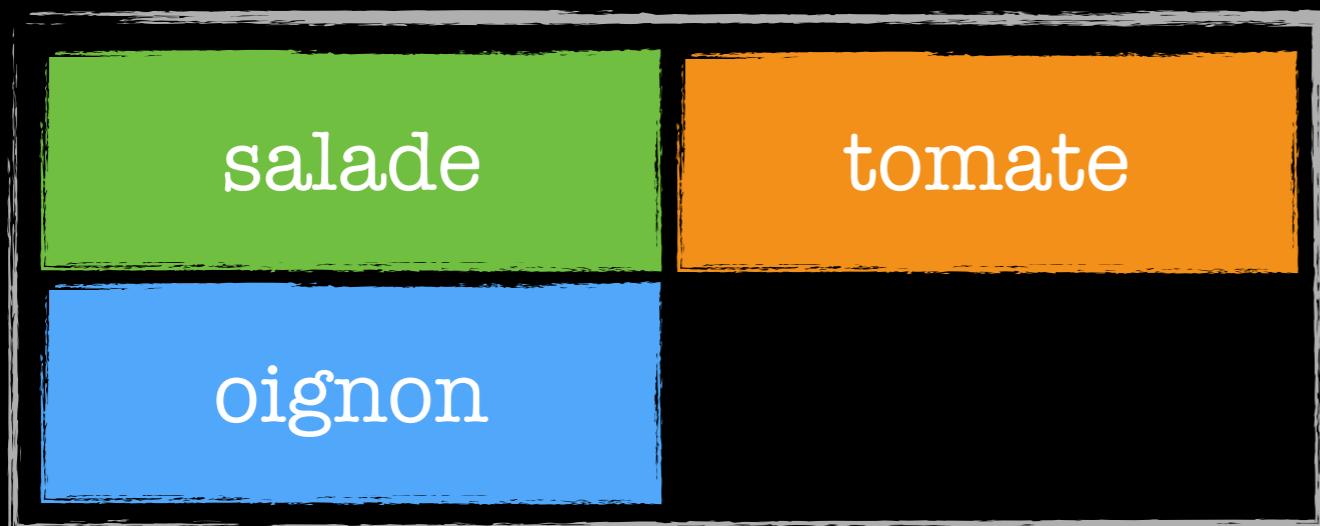
# et c'est pas fini !



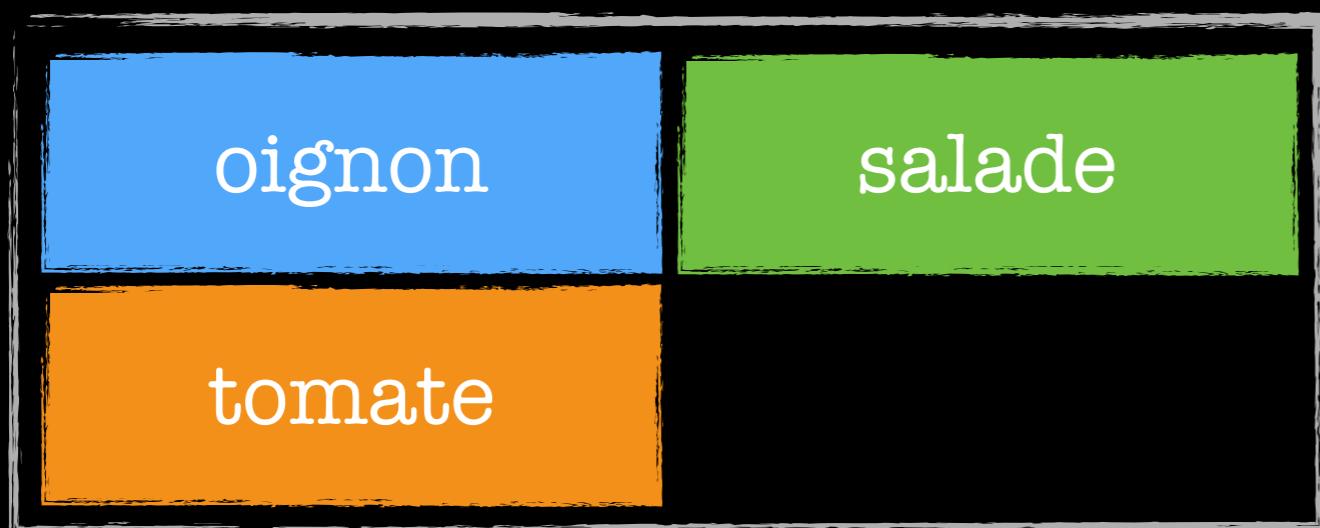
Si Si, il y a  
UN  
RAPPORT AVEC  
LES GRILLES !

changer l'ordre  
des vraies gouttières  
occuper plusieurs tracks  
remplissage intelligent

# changer l'ordre : order

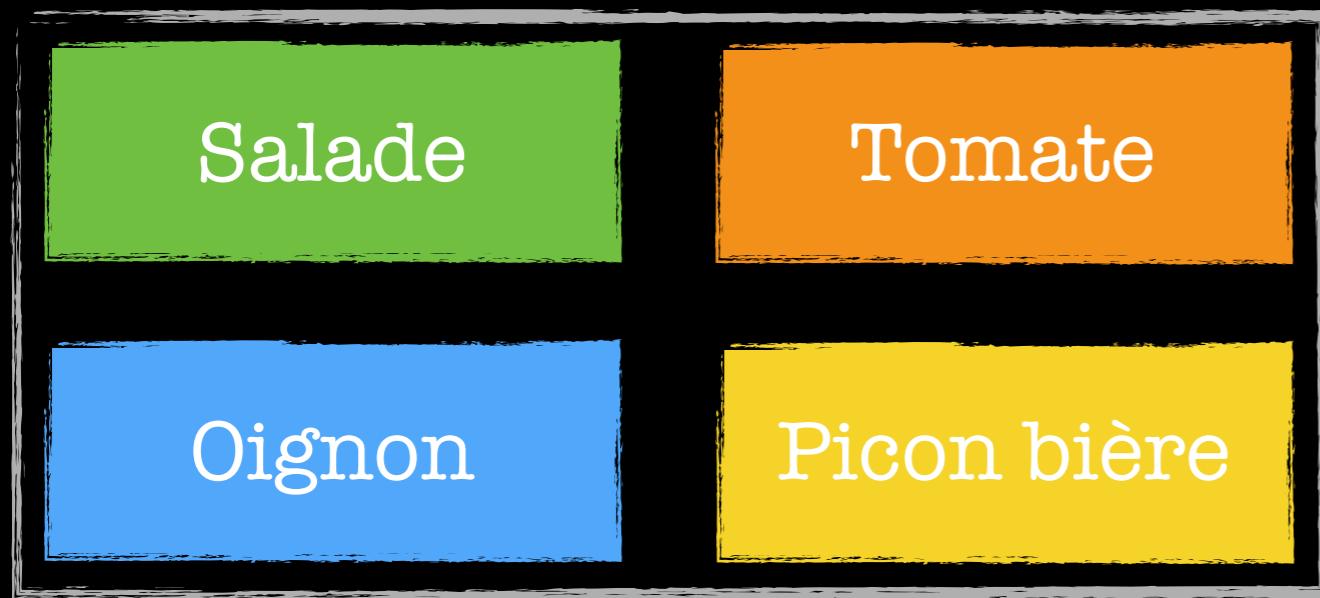


```
.grid-container {  
    display: grid;  
    grid: ". ."; ↗  
}  
raccourci de plein de trucs:  
grid-template-rows + grid-template-columns  
grid-auto-rows + grid-auto-columns  
grid-template-areas + grid-auto-flow
```



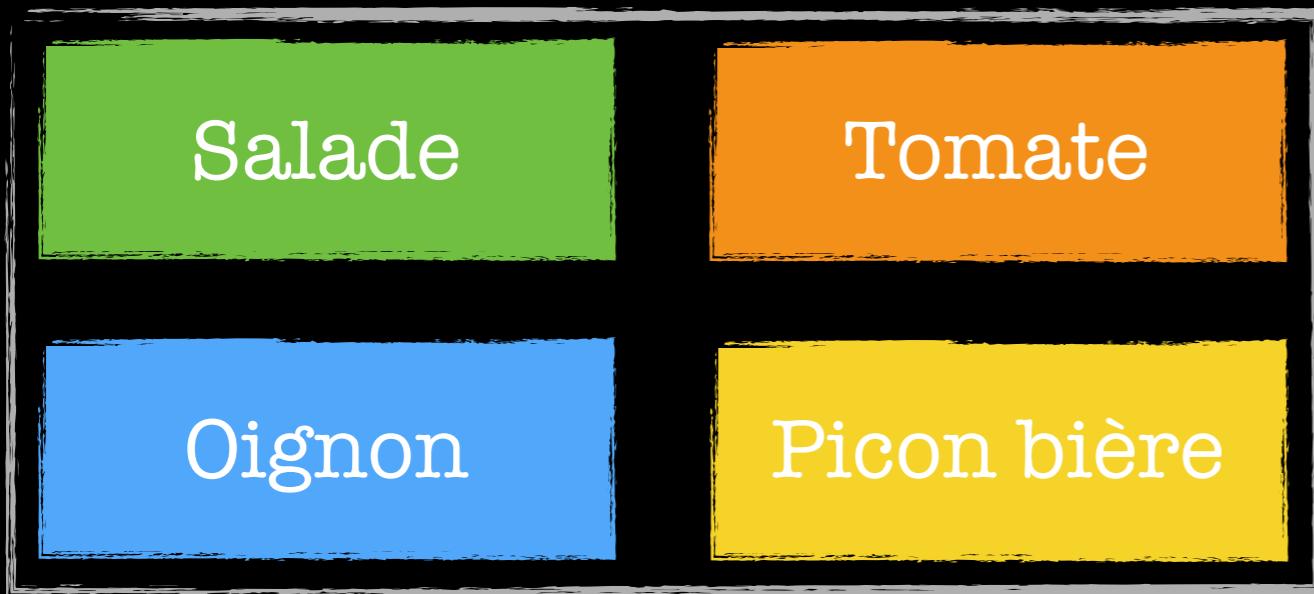
```
.grid-container {  
    display: grid;  
    grid: ". .";  
}  
.oignon {  
    order: -1;  
}
```

# gouttières ?



```
.flex-container {  
    display: flex;  
    flex-wrap: wrap;  
    margin-left: -10px;  
    margin-right: -10px;  
}  
.flex-item {  
    width: calc(50% - 20px);  
    margin-left: 10px;  
    margin-right: 10px;  
}
```

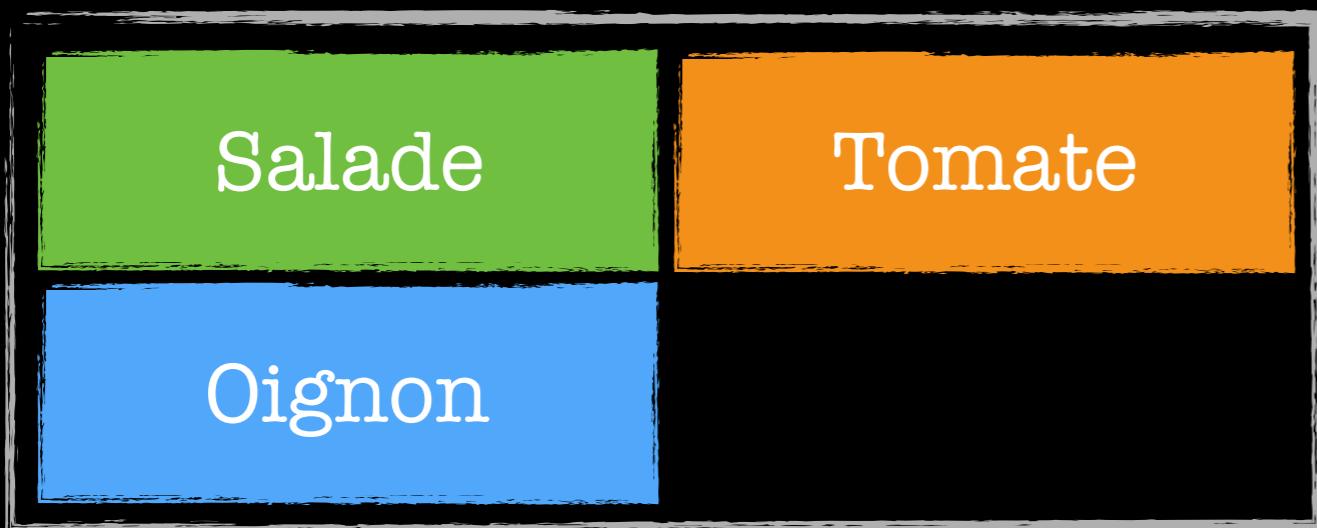
# gouttières !



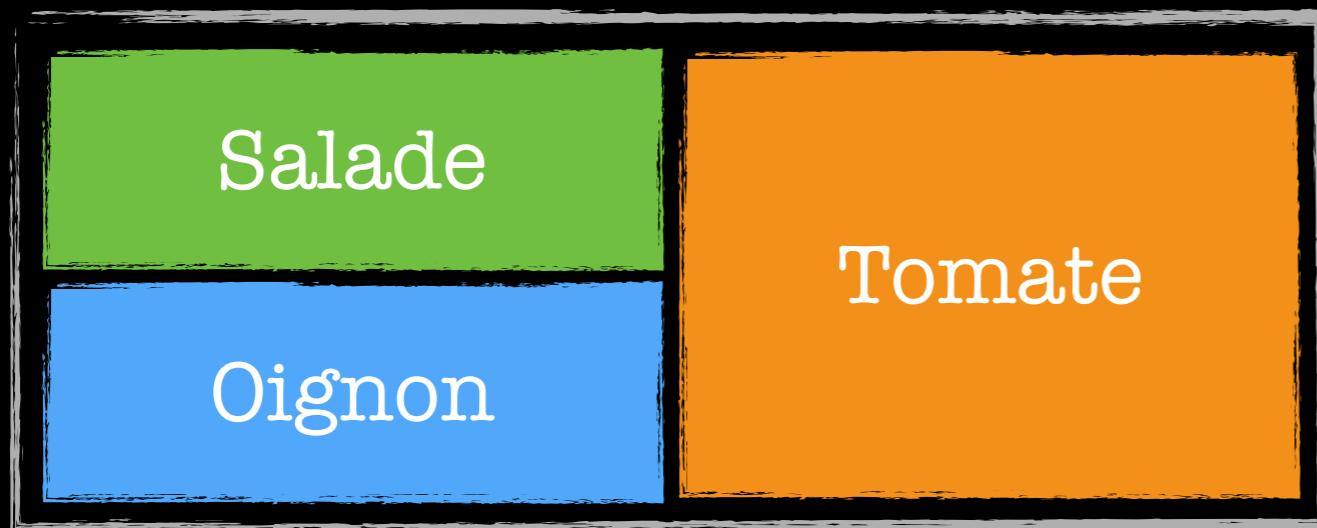
```
.grid-container {  
    display: grid;  
    grid: ". .";  
    grid-gap: 20px;  
}
```



# remplir les tracks : span

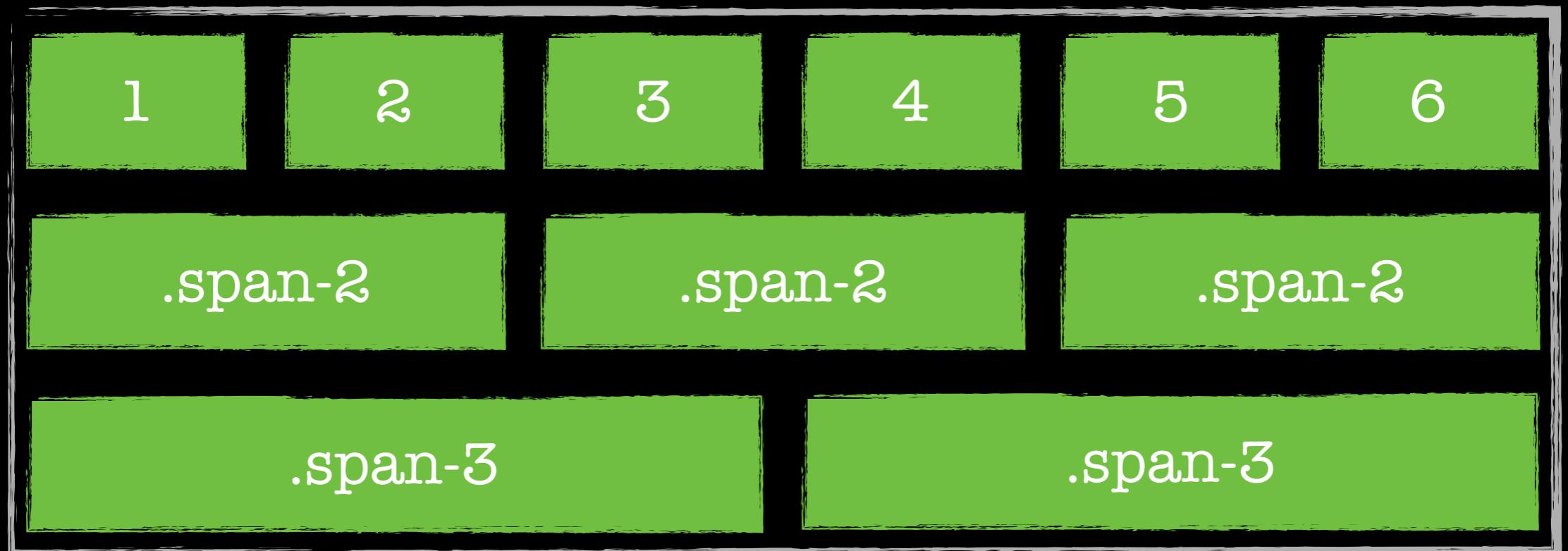


```
.grid-container {  
    display: grid;  
    grid: ". .";  
}
```



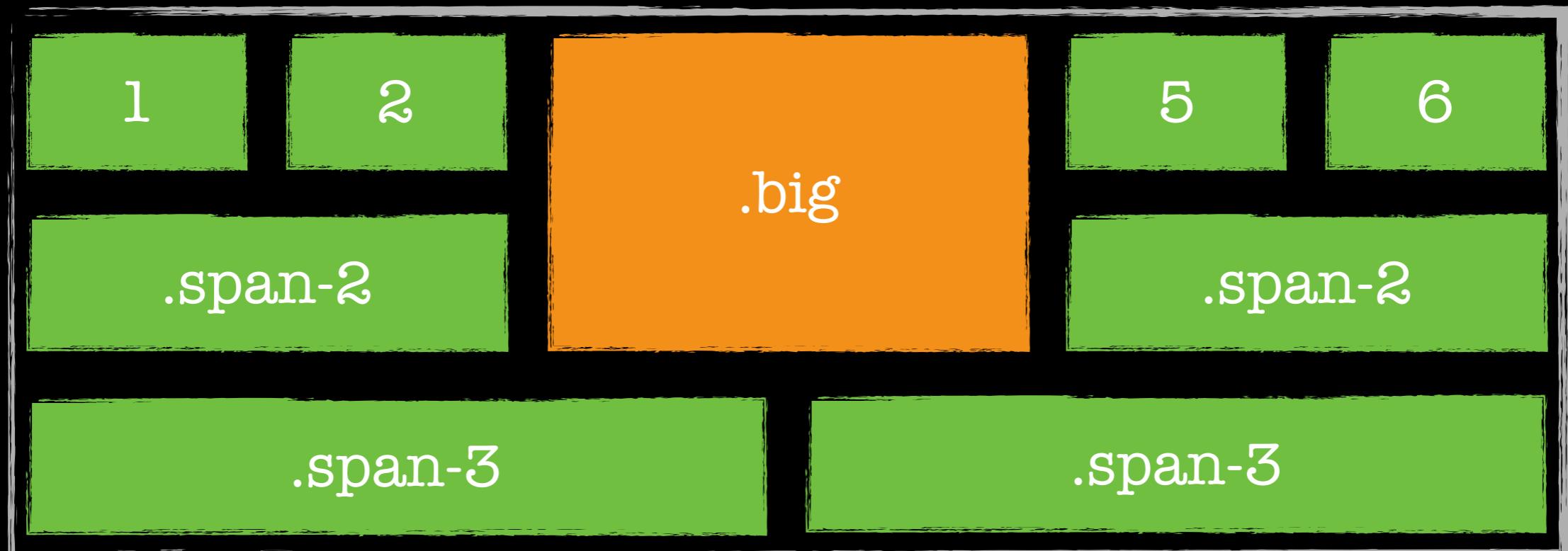
```
.grid-container {  
    display: grid;  
    grid: ". .";  
}  
.tomate {  
    grid-row: span 2;  
}
```

```
.container {  
    display: grid;  
    grid-template-columns: repeat(6, 1fr);  
    grid-gap: 10px;  
}  
  
.span-2 { grid-column: span 2; }  
.span-3 { grid-column: span 3; }
```

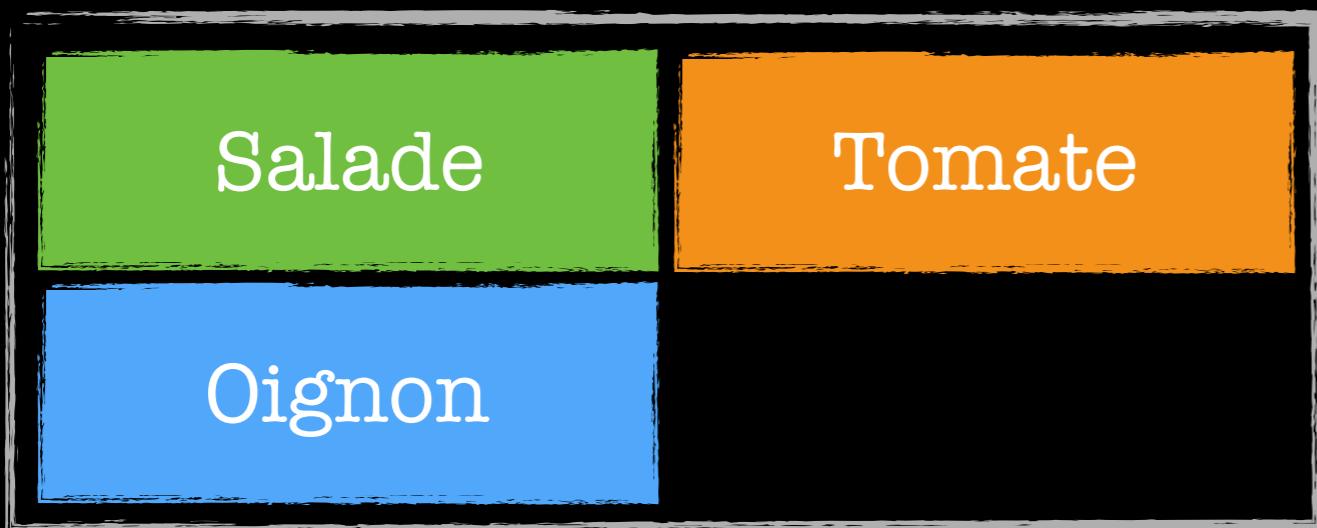


```
.container {  
    display: grid;  
    grid-template-columns: repeat(6, 1fr);  
    grid-gap: 10px;  
}
```

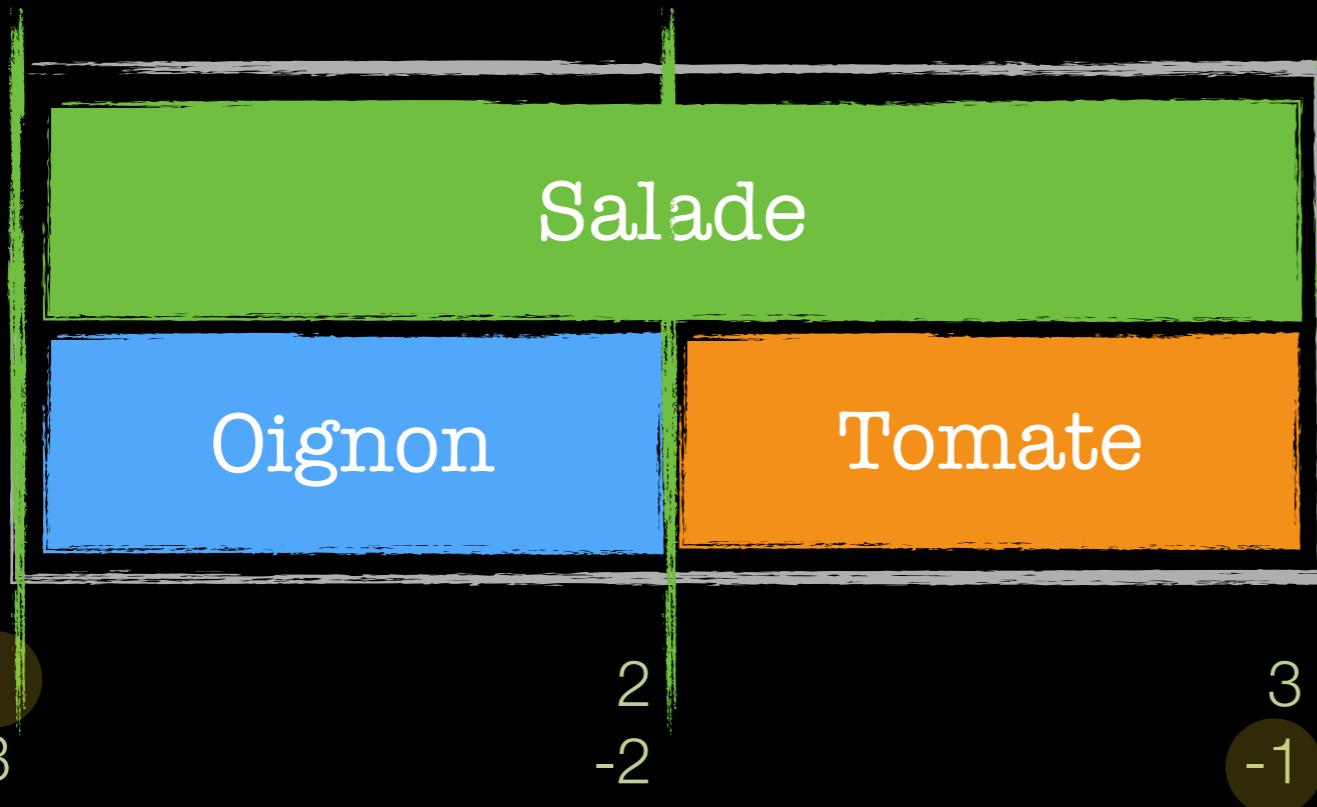
```
.big {  
    grid-row: span 2;  
    grid-column: span 2;  
}
```



# valeurs négatives

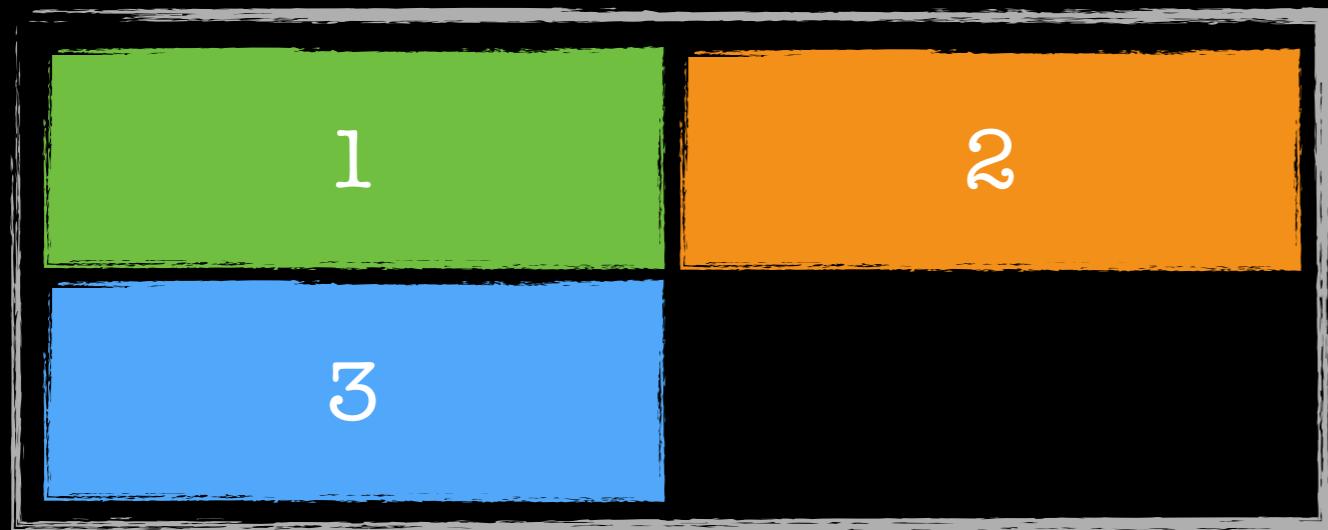


```
.grid-container {  
    display: grid;  
    grid: ". .";  
}
```



```
.grid-container {  
    display: grid;  
    grid: ". .";  
}  
.salade {  
    grid-column: 1 / -1;  
}
```

# grid-auto-flow



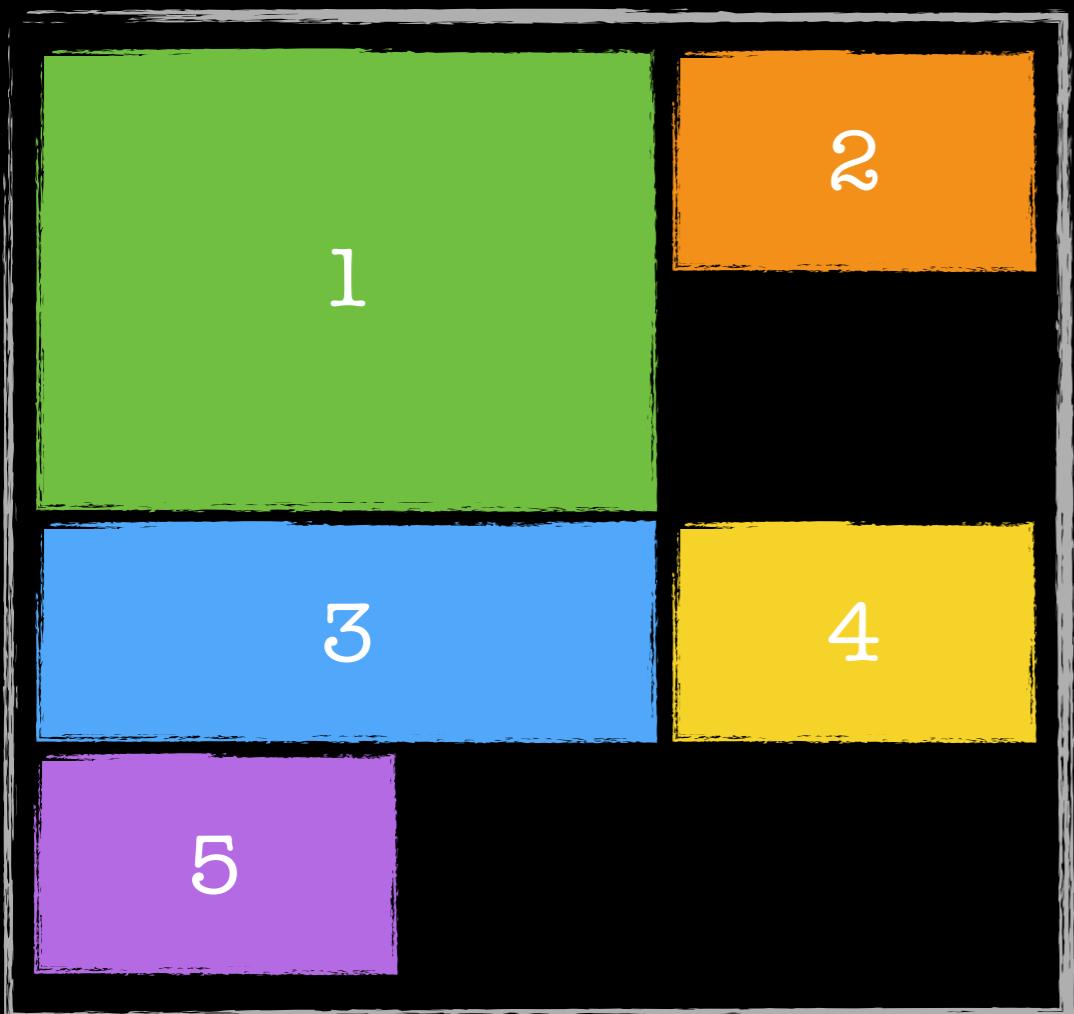
```
.grid-container {  
  display: grid;  
  grid: ". .";  
}
```



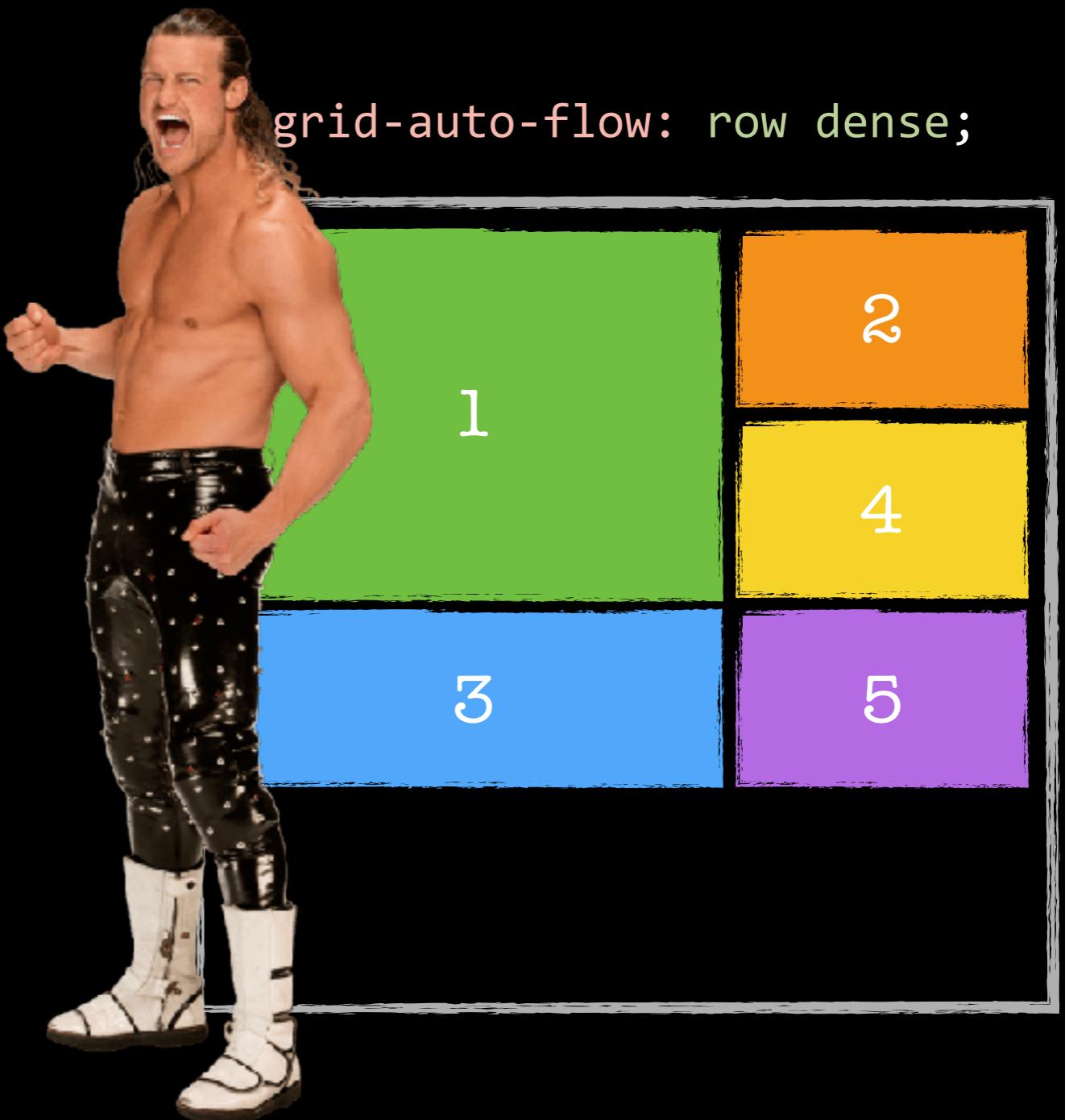
```
.grid-container {  
  display: grid;  
  grid: ". .";  
  grid-auto-flow: column;  
}
```

# grid-auto-flow: dense

grid-auto-flow: row;



grid-auto-flow: row dense;



<https://codepen.io/raphaelgoetter/pen/MpJMBo?editors=1100>

# repeat()

```
.grid-container {  
    display: grid;  
    grid-template-columns: repeat(4, 1fr);  
}
```

Salade

Tomate

Oignon

Picon bière

# repeat()

```
.grid-container {  
    display: grid;  
    grid-template-columns: repeat(3, 50px 100px);  
}
```



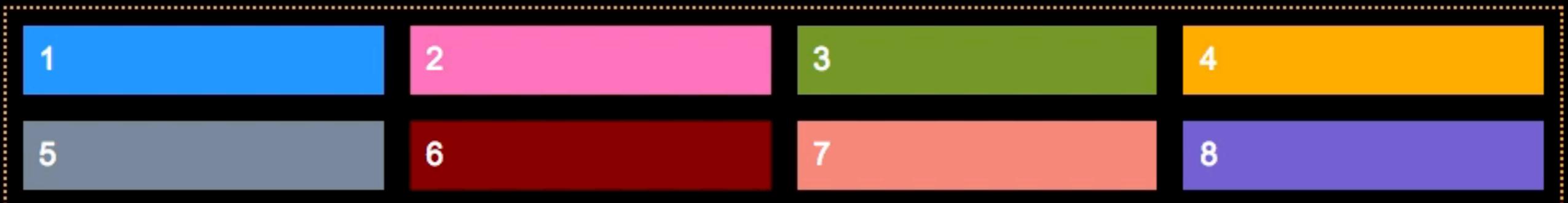
# auto-fit

```
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, 100px);  
}
```



# maxi combo !

```
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));  
}
```



<http://codepen.io/raphaelgoetter/pen/pgRQwv>

# automatisation

base :

```
.grid {  
  display: grid;  
}
```

# automatisation

base :

```
.grid {  
  display: grid;  
  --grid-number: 1;  
  --grid-gutter: 0;  
}
```



VARIABLES CSS ❤

<http://caniuse.com/#feat=css-variables>

# automatisation

base :

```
.grid {  
  display: grid;  
  --grid-number: 1;  
  --grid-gutter: 0;  
  grid-gap: var(--grid-gutter);  
}
```

# automatisation

base :

```
.grid {  
  display: grid;  
  --grid-number: 1;  
  --grid-gutter: 0;  
  grid-gap: var(--grid-gutter);  
  grid-template-columns: repeat(var(--grid-number), 1fr);  
}
```

# automatisation

base :

```
.grid {  
  display: grid;  
  --grid-number: 1;  
  --grid-gutter: 0;  
  grid-gap: var(--grid-gutter);  
  grid-template-columns: repeat(var(--grid-number), 1fr);  
}
```

variantes :

```
.grid-6 {  
  --grid-number: 6;  
}  
  
.grid-gutter-1 {  
  --grid-gutter: 1rem;  
}
```

# automatisation

```
<section class="grid grid-3 grid-gutter-1">  
  <div>Salade</div>  
  <div>Tomate</div>  
  <div>Oignon</div>  
  <div>Picon bière</div>  
</section>
```



<http://codepen.io/raphaelgoetter/pen/LNWbzz?editors=1100>

le mixin Sass de grille le  
plus court du monde

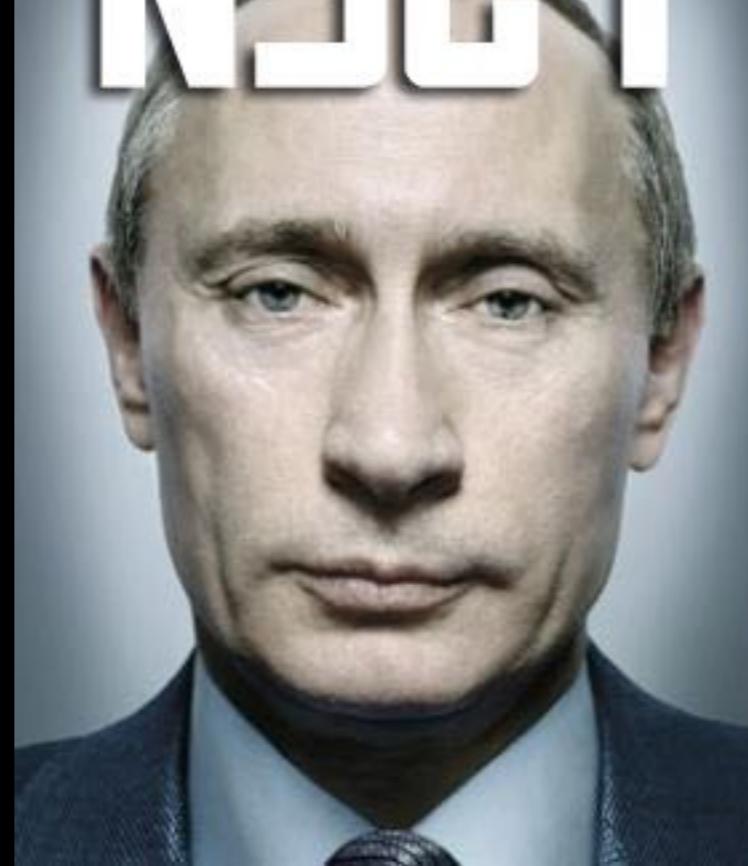
```
@mixin grid($number: 1, $gutter: 0, $bp: 0) {  
  @media (min-width: $bp) {  
    display: grid;  
    grid-template-columns: repeat($number, 1fr);  
    grid-gap: $gutter;  
  }  
}  
.container {  
  @include grid(3, 1rem, 640px);  
}
```



<http://codepen.io/raphaelgoetter/pen/eWOqQg?editors=1100>

**t'en veux encore plus ?  
un framework CSS basé sur Grid  
Layout ?**

NHET



# Grid Layout *est* un framework



CSS Working Group  
@csswg

CSS Grid **\*is\*** a grid system: it doesn't need framework to become one.  
Write clean markup, write clean CSS,  
learn+use CSS Grid for your grid!

Rachel Andrew @rachelandrew  
[buff.ly/2uvj1O](https://buff.ly/2uvj1O) You do not need a CSS Grid based Grid System.

À l'origine en anglais

18:33 - 5 juil. 2017

# Grid Layout en production ?!



					10-11		
natif							

\* ancienne spécification (où il manque des trucs chouettes)

natif							
polyfill							



# amélioration progressive

tu sais c'est quoi ?

# RÈGLES D'UNIVERSALITÉ

**une page web doit :**

1. S'adapter à tous les **visiteurs**  
(Sémantique, Accessibilité : title, alt, contrastes, ARIA)
2. S'adapter à tous les **devices**  
(Responsive, Mobile First)
3. S'adapter à tous les **navigateurs**  
(Amélioration progressive)
4. S'adapter à toutes les **connexions**  
(Performance)

*PS : « s'adapter » != « identique partout »*



mon design s'adapte à tous les devices,  
je pratique le

# *responsive web design*



GG!

BEAU GOSSE!

YEAH!

mon design s'adapte à tous les navigateurs,  
je pratique

# l'amélioration progressive



# amélioration progressive

The screenshot shows a web browser displaying the MDN (Mozilla Developer Network) website. The page title is "Les grilles CSS et l'amélioration progressive". The MDN logo is in the top left, and the Mozilla logo is in the top right. The navigation bar includes links for "TECHNOLOGIES WEB", "DOCUMENTATION DE MOZILLA", "OUTILS DE DÉVELOPPEMENT", "VOTRE AVIS", and a search bar. Below the navigation is a breadcrumb trail: "MDN > Technologies web pour développeurs > CSS > Grilles CSS (CSS Grid) > Les grilles CSS et l'amélioration progressive". To the right of the breadcrumb are buttons for "LANGUES", "MODIFIER", and settings. The main content area starts with a section titled "VOIR AUSSI" containing links like "CSS", "Référence CSS", and various "CSS Grid Layout" guides. Below this is a paragraph about the specification becoming available in major browsers. A large heading "Les navigateurs qui prennent en charge les grilles CSS" is followed by a detailed paragraph about browser support and interoperability. To the right, a sidebar titled "DANS CET ARTICLE" lists related topics such as "Les navigateurs qui prennent en charge les grilles CSS", "La situation pour Internet Explorer et Edge", "L'utilisation d'autoprefixer pour la prise en charge de la grille", "Puis je utiliser les grilles CSS sans danger?", "Commencer à utiliser les grilles CSS en production", "Effectuer la transition depuis une disposition flottante", "Utiliser les requêtes de fonctionnalité (feature queries)", "Surcharger les autres valeurs pour display", "Comment la spécification gère-t-elle cette surcharge?", "Les autres valeurs de display", and "Les éléments flottants".

Au printemps 2017, on voit pour la première fois une spécification majeure être disponible presque simultanément dans différents navigateurs : les grilles CSS (CSS Grid). Les grilles CSS devraient donc être disponibles dans les versions finales de Firefox, Chrome, Opera et Safari. Malgré cela, si ces navigateurs récents permettront à la plupart d'entre nous de profiter de ces nouvelles fonctionnalités, il existe d'anciens navigateurs ou d'autres navigateurs qui ne prennent pas en charge ces fonctionnalités. Dans ce guide, nous allons parcourir différentes stratégies pour gérer cette prise en charge.

## Les navigateurs qui prennent en charge les grilles CSS

En dehors d'Internet Explorer et de Edge, les propriétés et valeurs relatives aux grilles CSS ne sont pas préfixées dans Safari, Chrome, Opera et dans Firefox. Toutes les propriétés et valeurs que nous avons vues dans ce guide fonctionnent de façon interopérable entre ces navigateurs. Cela signifie que si vous écrivez des règles CSS pour paramétrier une grille, le document doit avoir le même rendu dans Firefox, Opera et dans Chrome. Les grilles CSS

# @supports

à la rescousse !

```
.container { fallback en flexbox, float ou table-cell
  display: flex; ↩
}
```

```
@supports (display: grid) {
  .container {
    display: grid;
  }
}
```

# en vrai ?

Register • Sign In

# SLATE

News & Politics   Culture   Technology   Business   Human Interest



**HENRY GABAR**  
**Don't Give Up on the Small City**

**FRANK BOWMAN**  
**Mueller Can't Take Down Trump, so He Must Get Congress to Do It. Here's How.**

**NICK GREENE**  
**Going to Golden State Would Certify LeBron as the Most LeBron Player of All Time**

**JAMELLE BOUIE**  
**The Fight for a White America**  
A century ago, nativists won the battle over immigration. The Trump administration is determined to repeat it.

**POTUS**  
Trump: Orrin Hatch Said I Am Next President in History Orrin Hatch: Um, Not Quite.

**BACKLASH**  
Wolf Gub's Offensive After He's Kicked Off Show for Peddling Haley-Trump Affair Rumors

**CRIME**  
12 Year-Old Girl Arrested in L.A. Middle School Shooting

**MOVIES**  
An Incomplete List of "Explicitly" Straight Characters in Harry Potter

**MUELLER INVESTIGATION**  
Bachmann Accuses Nunes of Giving "Secretly Altered" Version of Memo to White House

<https://slate.com/>

# grid polyfill

The screenshot shows the GitHub repository page for 'css-grid-polyfill'. The repository has 72 commits, 4 branches, 2 releases, 2 contributors, and is licensed under MIT. It was last updated on 6 Sep 2016. The repository description is: "A working implementation of css grids for current browsers."

**Code** Issues (17) Pull requests (1) Projects (0) Wiki Pulse Graphs

72 commits 4 branches 2 releases 2 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

FremyCompany Updated package.js to work around unpublished package Latest commit 7935386 on 6 Sep 2016

File	Description	Time
bin	New build	2 years ago
demo/css-grid	Fixed: scroll preservation on relayout	2 years ago
dcc	Import css-grid from css-polyfills	2 years ago
src	Fixed: scroll preservation on relayout	2 years ago

<https://github.com/FremyCompany/css-grid-polyfill>

# récatipulatif



# récatipulatif

- ✓ syntaxe simple
- ✓ séparation HTML / CSS totale
- ✓ framework intégré
- ✓ gouttière
- ✓ bidirectionnel
- ✓ alignements
- ✓ auto-flow dense
- ✓ réordonnement
- ✓ chevauchements
- ✓ responsive



IE

# QUIZ !

quelles déclarations n'ont aucun effet ?

(= ne s'appliquent pas, ou ne changent pas les valeurs par défaut)

.grid-item

```
.grid-container {  
    display: grid; ✓  
    grid-template: 200px / 1fr; ✓  
    align-items: space-between;  
    flex-direction: row;  
}
```

```
.grid-item {  
    display: inline-flex; ✓  
    align-content: center;  
    grid-gap: 10px;  
    float: left;  
    clear: both;  
    z-index: -1; ✓  
    left: 200px;  
    vertical-align: top;  
    justify-self: center; ✓  
    height: 25%; ✓  
}
```

# ressources



# grid by example

## Grid by Example

A collection of usage examples for the CSS Grid Layout specification.

What is Grid Layout? Learn Grid Layout Browser Support Resources Blog

## CSS Grid Layout

The CSS Grid Layout specification is one of a few new specifications that are redefining how we approach layout for the web. Alongside Flexbox and the Box Alignment Module it will become part of a modern layout system for websites and web applications.

This site is growing collection of [examples](#) and [video](#), with the aim of helping people understand how Grid Layout works.

Grid is starting to ship in browsers. If your browser doesn't have support you may need to enable a flag. [Read more about browser support for Grid](#).

**Get Started with Grid**

**A simple grid**

A	B	C
D	E	F

The simplest example. Create a three column grid. Grid Layout will auto place items one in each cell.

[View example 1](#)

<http://gridbyexample.com/>

# learn CSS grid



A guide to learning CSS grid · @jonsuh

CSS Grid is a powerful tool that allows for two-dimensional layouts to be created on the web. This guide was created as a resource to help you better understand and learn Grid, and was organized in a way I thought made the most sense when learning it.

<http://learncssgrid.com/>

# Gridcheat

**grid layout cheatsheet** by @davidebbo

grid concepts and example		defining the grid	
		<code>grid: 1 / 10px 1fr 1fr 1fr; grid-template-columns: 10px 1fr 1fr 1fr 1fr; grid-template-rows: 1fr 1fr 1fr 1fr; grid-template-areas: 'one' 'two' 'three' 'four';</code>	<code>display: grid; grid-template-rows: 1fr 1fr 1fr 1fr; grid-template-columns: auto auto auto auto; grid-template-areas: 'one' 'two' 'three' 'four'; grid-template-rows: 1fr 1fr 1fr 1fr; grid-template-columns: 10px 1fr 1fr 1fr 1fr; grid-template-areas: 'one' 'two' 'three' 'four';</code>
grid gap	grid area	grid areas	grid
gaps		auto placement	
placing items		container alignment	
<code>grid-row-start: line-name; line-number;</code>	<code>grid-row-end: line-name; line-number;</code>	<code>grid-row: start; grid-row-end: end;</code>	<code>align-content: vertical-align for whole grid;</code>
<code>grid-column-start: line-name; line-number;</code>	<code>grid-column-end: line-name; line-number;</code>	<code>grid-column: start; grid-column-end: end;</code>	<code>start: vertical-align top; space-around: vertical-align space-around; center: vertical-align center; justify-content: horizontal-align for whole grid;</code>
<code>grid-area: short for... everything;</code>	<code>span N: keyword for row or column spanning;</code>	<code>grid-area: short for grid-area;</code>	<code>place-items: flexbox for align-items and justify-items;</code>
patterns		item alignment	
<code>align-self: align-self:center; justify-self: justify-self:center;</code>	<code>grid-column: auto 1fr 1fr 1fr;</code>	<code>grid-area: grid-area:column:row(1, 10px 1fr);</code>	<code>grid-area: grid-area:grid-column:grid-row(1, 1fr 1fr 1fr); justify-self: justify-self:space-between;</code>
resources		further reading	
<a href="#">grid-cheat.com</a>	<a href="#">grid-cheat.com</a>	<a href="#">CSS Grid Examples</a>	<a href="#">CSS Grid Resources</a>
<a href="#">grid-examples.com</a>	<a href="#">grid-examples.com</a>	<a href="#">grid-examples.com</a>	<a href="#">grid-examples.com</a>
<a href="#">Learn CSS Grid</a>	<a href="#">Learn CSS Grid</a>	<a href="#">grid-examples.com</a>	<a href="#">grid-examples.com</a>
<a href="#">CSS Grid Video</a>	<a href="#">CSS Grid Video</a>	<a href="#">grid-examples.com</a>	<a href="#">grid-examples.com</a>
<a href="#">CSS Grid Garden</a>	<a href="#">CSS Grid Garden</a>	<a href="#">grid-examples.com</a>	<a href="#">grid-examples.com</a>

<http://kiwi.gg/gridcheat>

# grid examples

## Grid Examples

Get inspired by real and live sites collected from bloggers, designers and developers that are using CSS Grid Layout.

Are you a designer or an agency and have you been working on a CSS Grid site recently? Then add yours by submitting your work [here!](#)

The screenshot shows a grid of six website examples. From left to right, the websites are:

- wedding-invite.glitch.me
- next.davidlochheadxyz
- lucaslemonnier.com
- lynaandtonic.com
- anemohrt.me
- drktinaallar.com

<http://www.gridexamples.com/>

# Gridbugs

The screenshot shows the GitHub repository page for `rachelandrew / gridbugs`. The repository has 67 stars and 8 forks. It contains 23 commits, 1 branch, 0 releases, and 2 contributors. The latest commit was made on 21 Sep 2017. The repository is licensed under MIT. The README.md file is present.

A curated list of Grid interop issues

css css-grid-layout

23 commits 1 branch 0 releases 2 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

**rachelandrew committed on 21 Sep 2017 update on issue 14** Latest commit szczes on 21 Sep 2017

gitignore	Initial files	8 months ago
LICENSE	Initial commit	8 months ago
README.md	update on issue 14	7 months ago

README.md

## GridBugs

Inspired by [Flexbugs](#) this list aims to be a community curated list of CSS Grid Layout bugs, incomplete implementations and interop issues. Grid shipped into browsers in a highly interoperable state, however there are a few issues - let's document any we find here.

<https://github.com/rachelandrew/gridbugs>

# Grid vs IE

CR Level 1 Property	IE10 Implementation	Autoprefixer	Note
grid-template-columns	-ms-grid-columns	Yes	
grid-template-rows	-ms-grid-rows	Yes	
grid-template-areas	-	No	
grid-template	-	No	Shorthand
grid-auto-columns	-	No	
grid-auto-rows	-	No	
grid-auto-flow	-	No	
grid	-	No	Shorthand
grid-row-start	-ms-grid-row	Yes	
grid-column-start	-ms-grid-column	Yes	
grid-row-end	-	No	Defined by the -ms-grid-row-span property
grid-column-end	-	No	Defined by the -ms-grid-column-span property

<https://rachelandrew.co.uk/archives/2016/11/26/should-i-try-to-use-the-ie-implementation-of-css-grid-layout/>

# CSS grid garden

## GRID GARDEN

◀ Level 11 of 24 ▶

Try using `grid-column` to water these carrots. The `span` keyword also works with this shorthand property so give it a try!

```
1 #garden {  
2   display: grid;  
3   grid-template-columns: 20% 20% 20% 20% 20%;  
4   grid-template-rows: 20% 20% 20% 20% 20%;  
5 }  
6  
7 #water {  
8     
9 }  
10  
11  
12  
13  
14
```

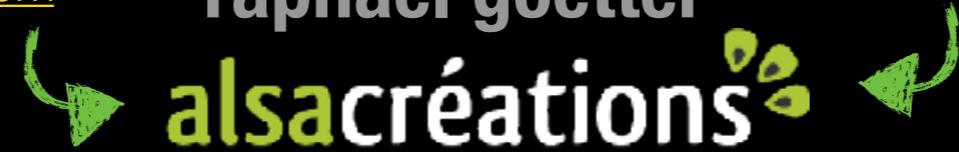
[Next](#)

<http://cssgridgarden.com/>

# merci !

slides sur  
[www.speakerdeck.com/goetter](http://www.speakerdeck.com/goetter)

forum  
communauté  
[www.alsacreations.com](http://www.alsacreations.com)

raphaël goetter  
**alsacreations** 

agence web  
organisme de formation  
[www.alsacreations.fr](http://www.alsacreations.fr)