CSS Grids or:

How I Learned to Stop Worrying and Love the new spec

By Ariel Wiznia

Hello!

- I'm a frontend developer currently working at Softvision, Buenos Aires
- I've been working on the web since the marvelous Flash years
- I survived the table layout...
- Embraced the new div way to build websites...
- and now I'm ready for the future!

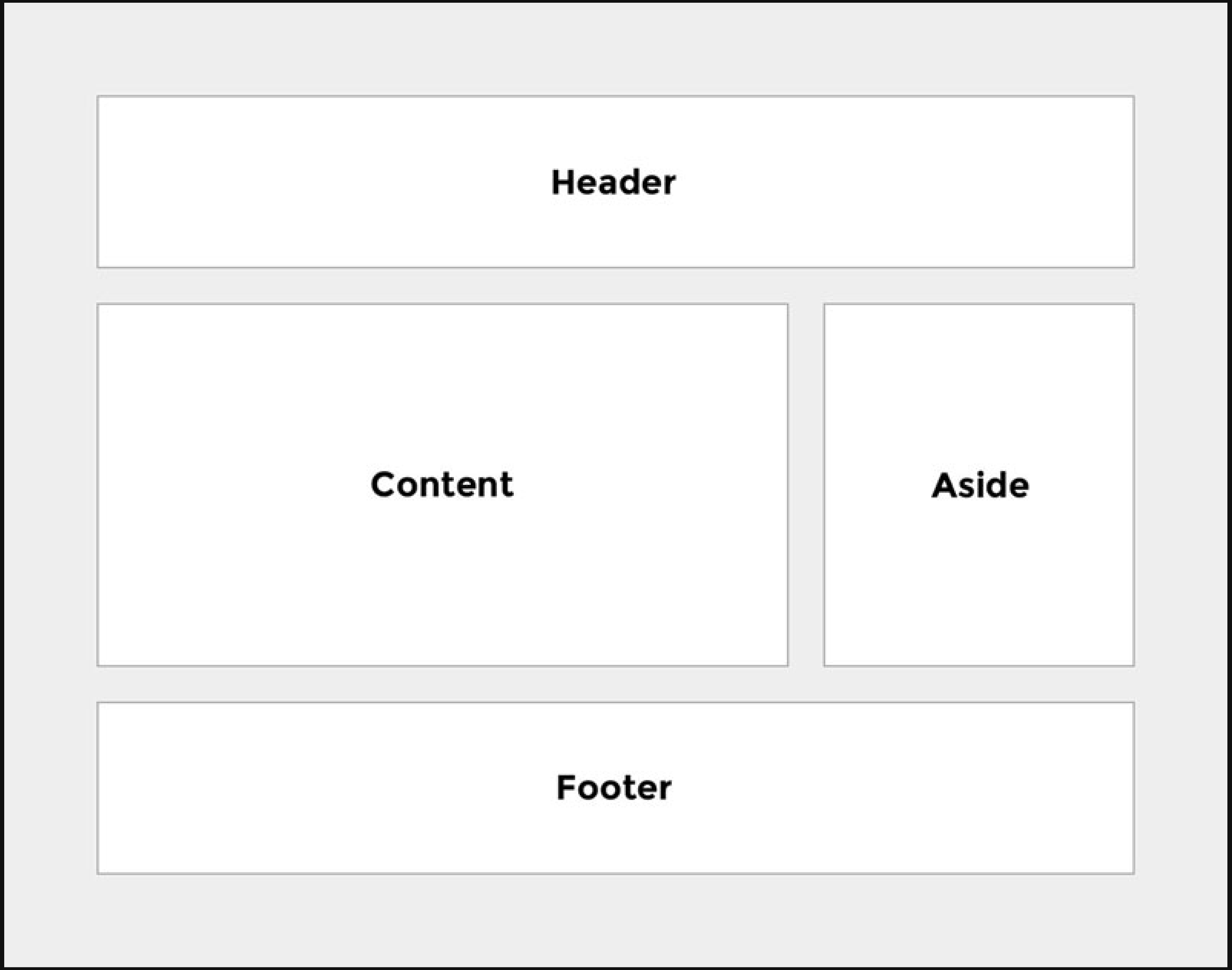
So...what's the future?

GRIDS!

Grids are a new way of organizing a layout via CSS on a page

It allows us to define a structure in which you can control what goes into each row or column and in what order.

Standard grid



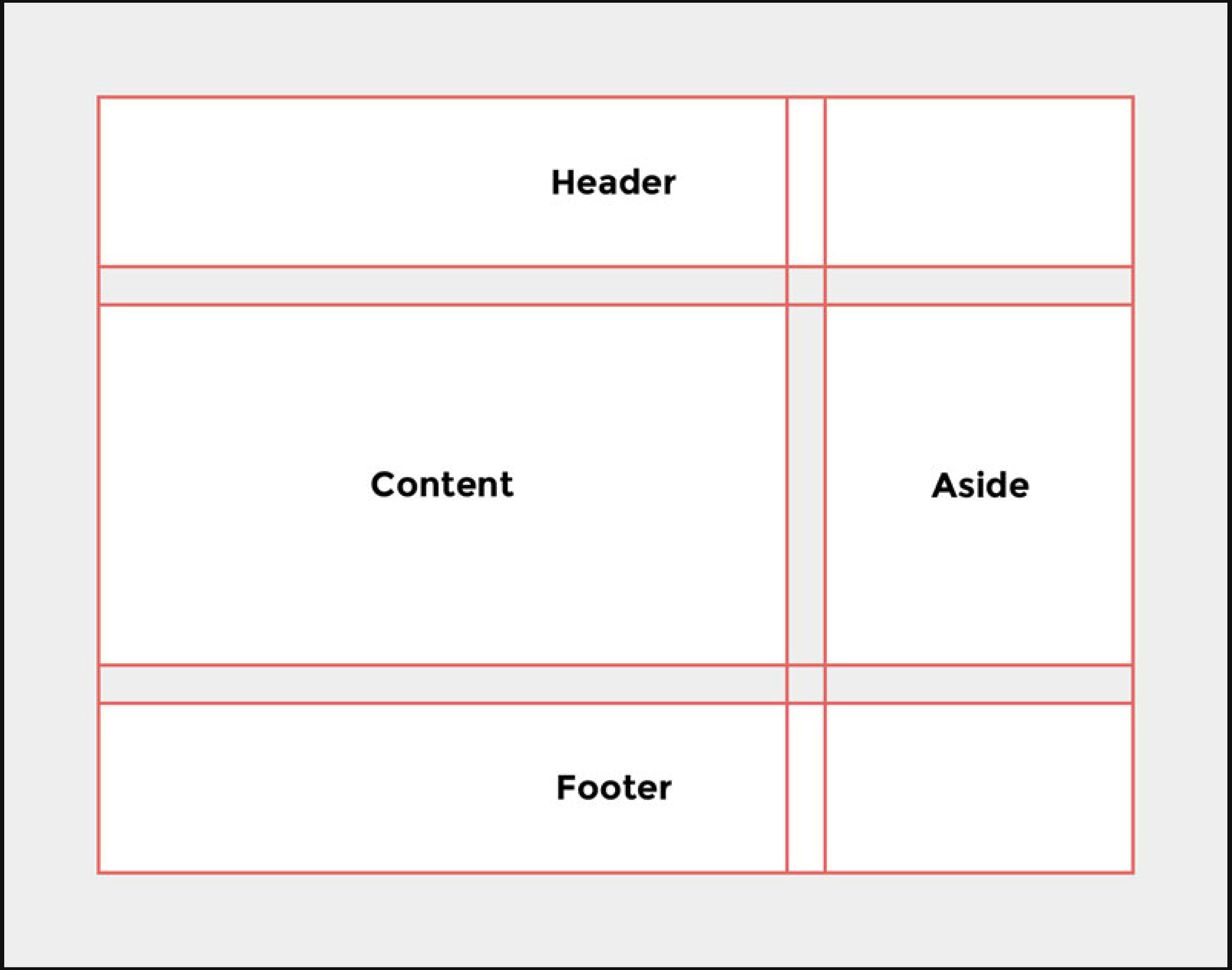
Header

Content

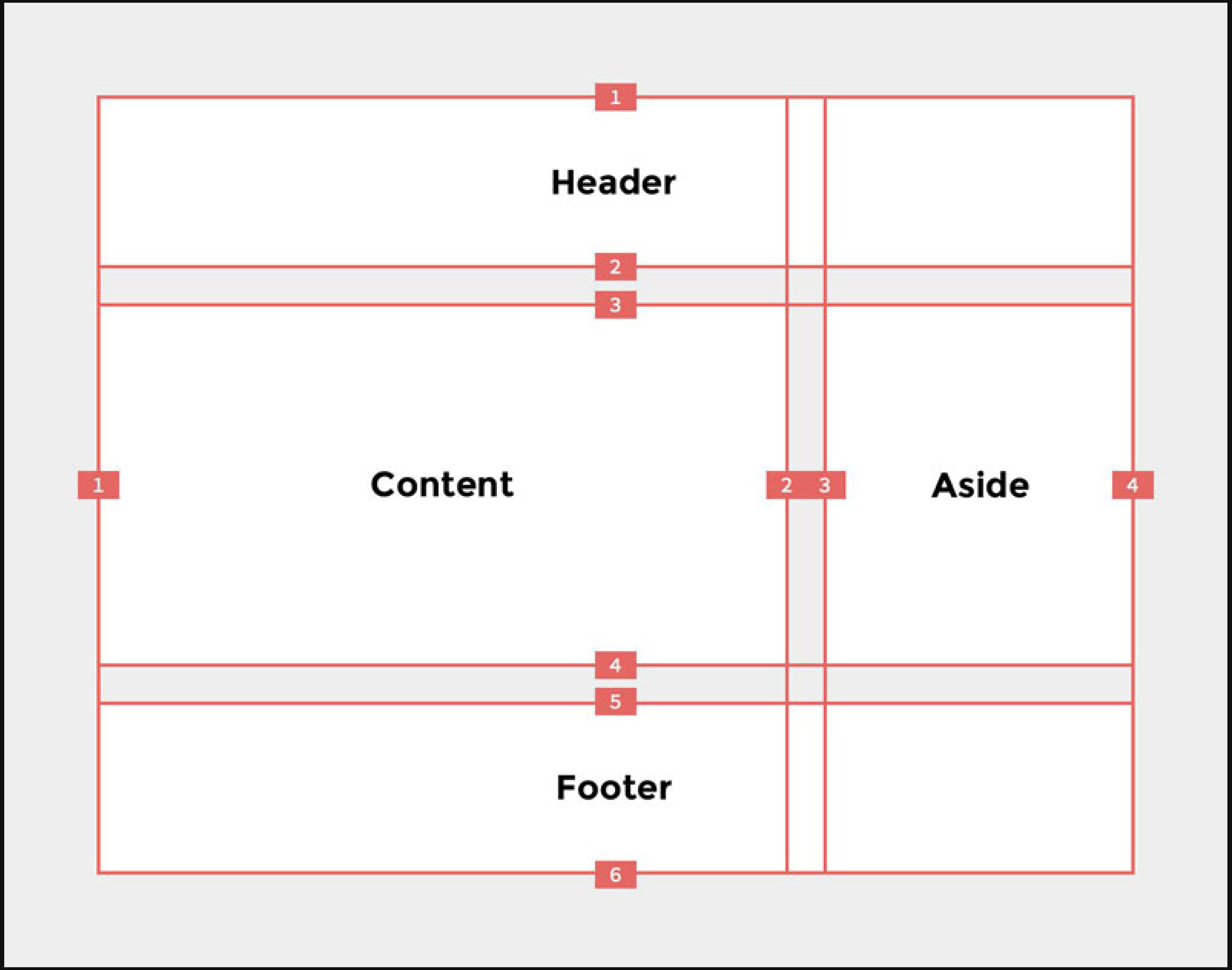
Aside

Footer

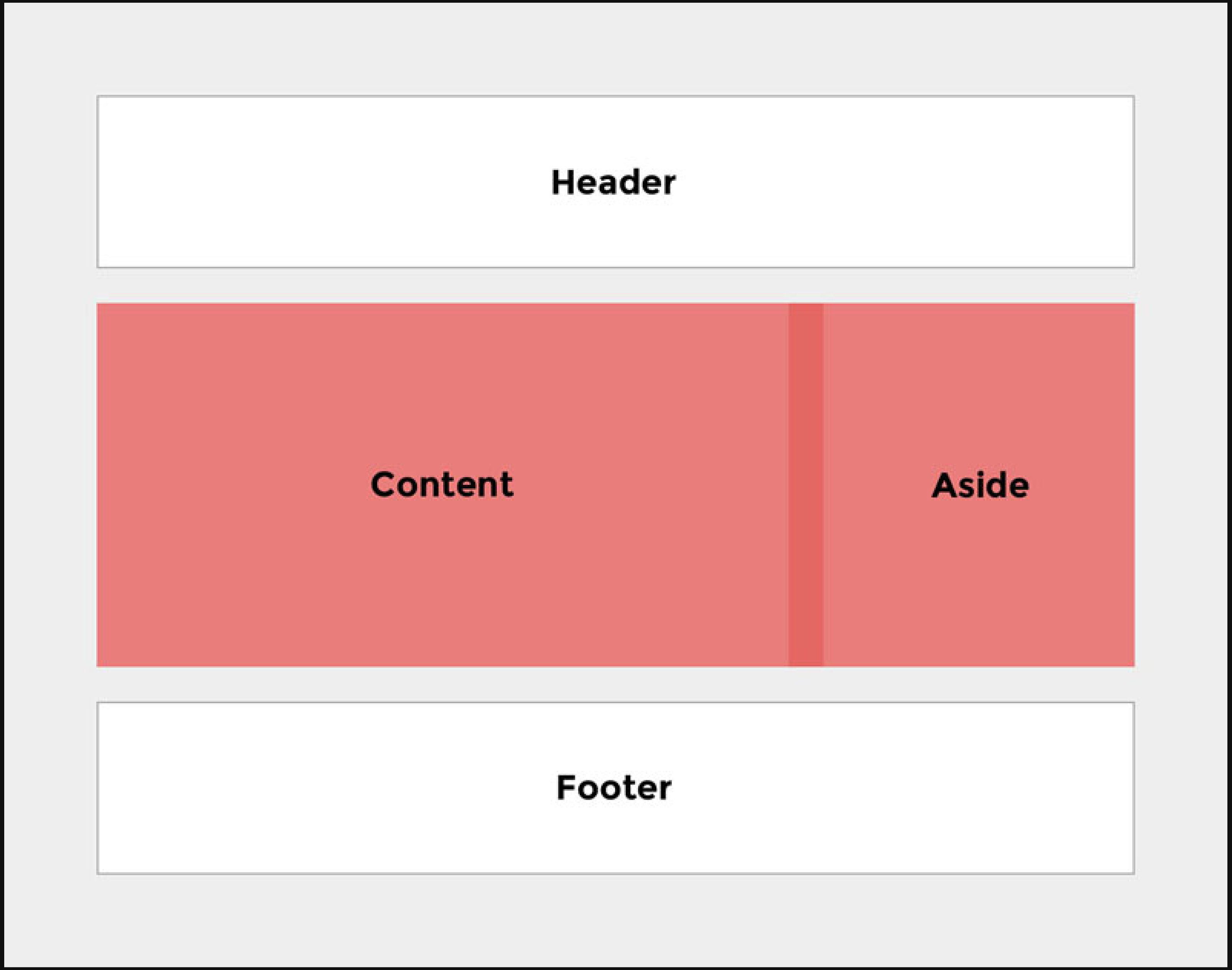
Grid lines



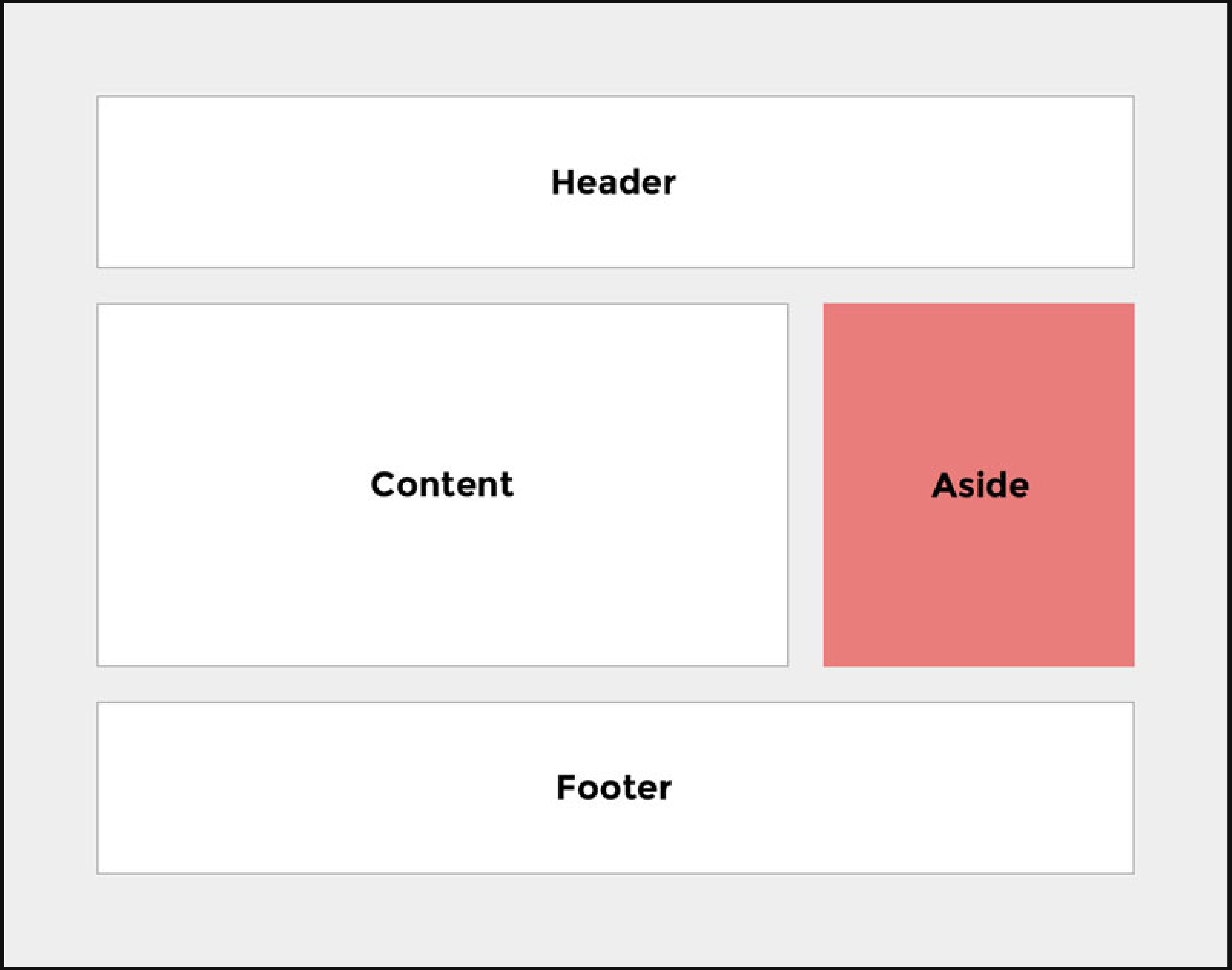
Grid lines (highlighted)



Grid track



Grid cell



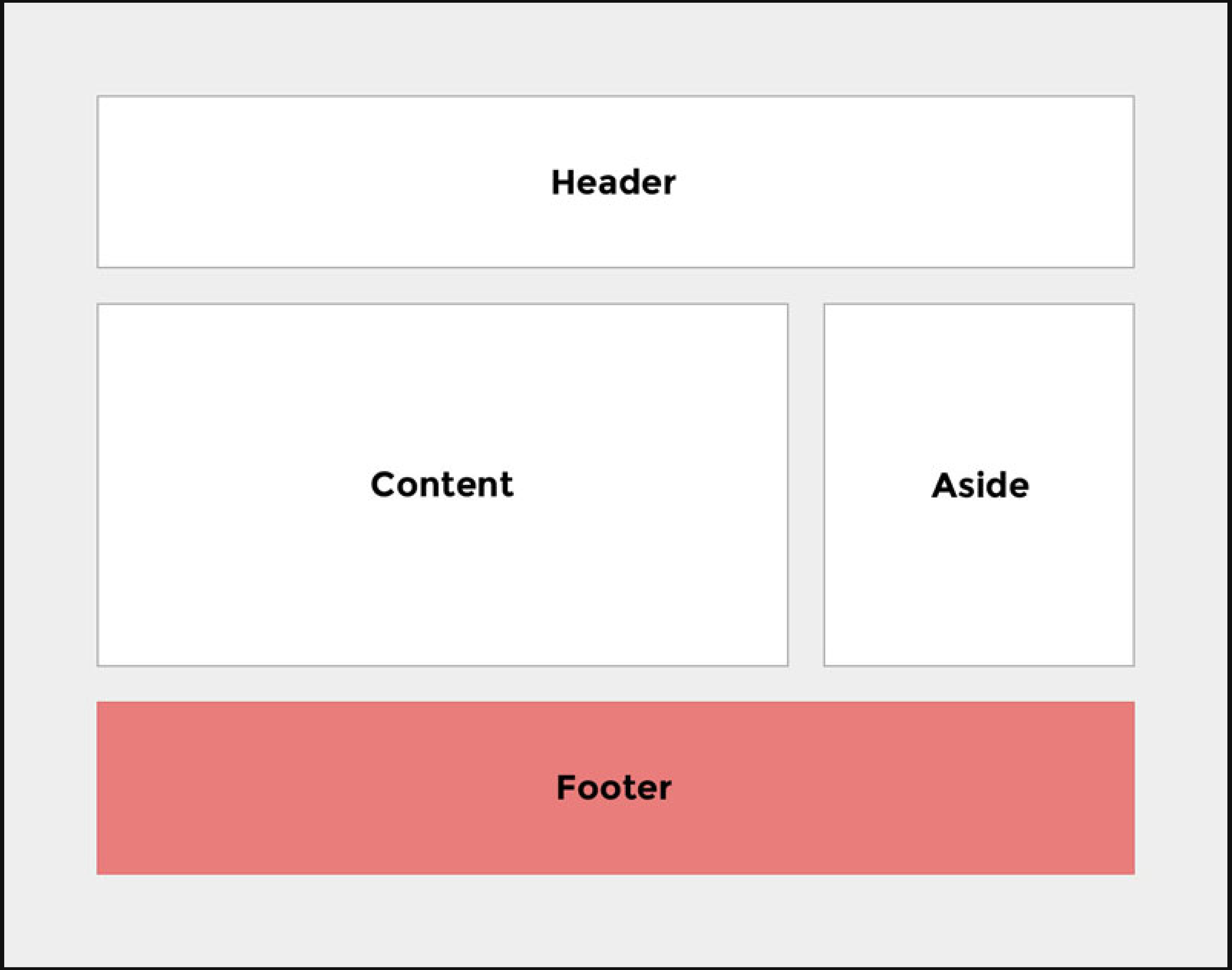
Header

Content

Aside

Footer

Grid area



Header

Content

Aside

Footer

Defining a grid

HTML:

```html
<div class="container">
    <div class="header">Header</div>
    <div class="main">Content</div>
    <div class="aside">Aside</div>
    <div class="footer">Footer</div>
</div>
```

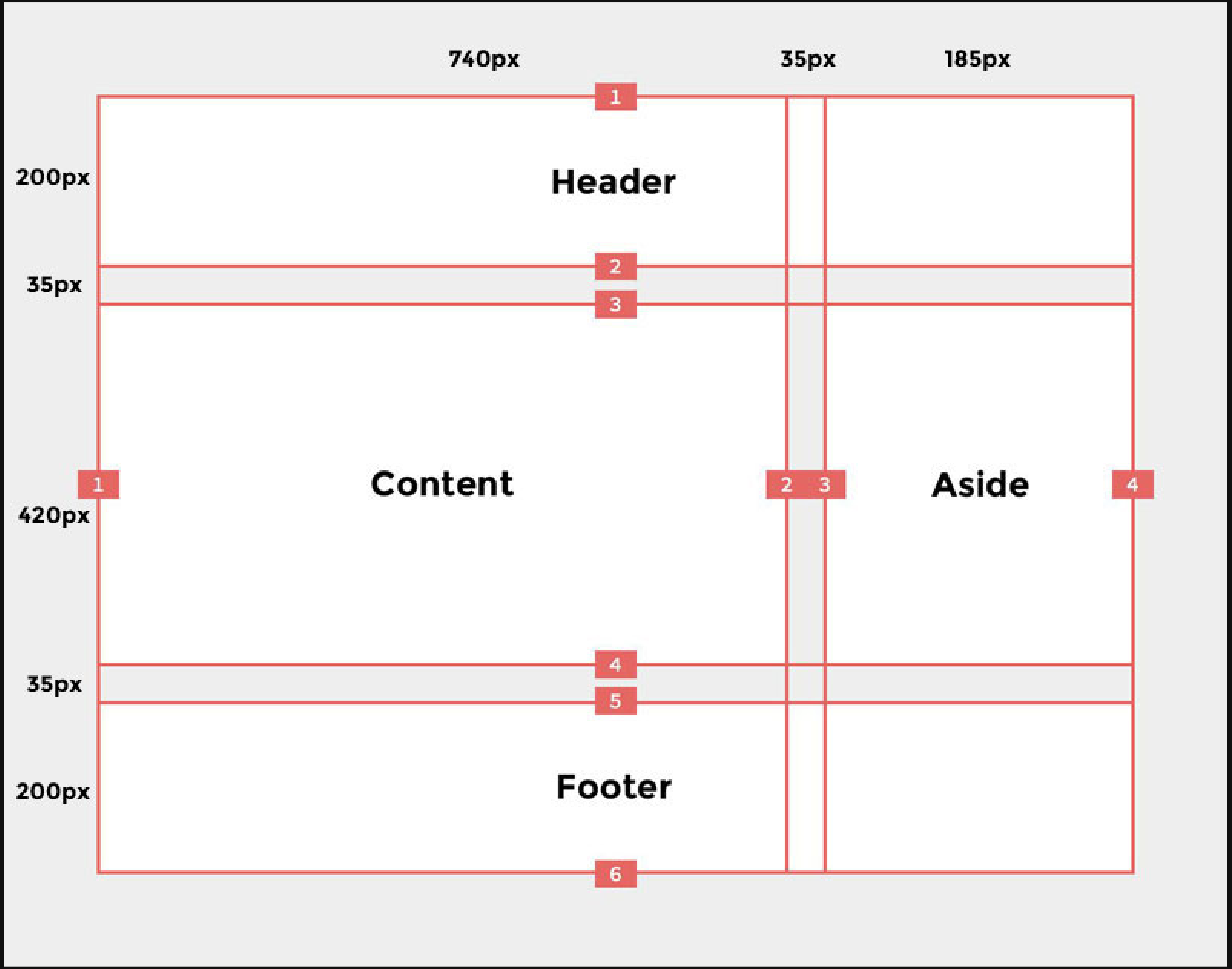
CSS:

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

Setting grid layout

CSS:

```
.container {
    display: grid;
    grid-template-rows: 200px 35px 420px 35px 200px;
    grid-template-columns: 740px 35px 185px;
    margin: auto;
    max-width: 960px;
}
.container div {
    border: 2px solid #c7c7c7;
    background-color: #fff;
    display: flex;
    align-items: center;
    justify-content: center;
    font: bold 30px/1 'Montserrat', Helvetica, sans-serif;
```


740px
35px
185px
200px
35px
420px
35px
200px
Header
Content
Aside
Footer
1
2
3
4
5
6
1
2
3
4

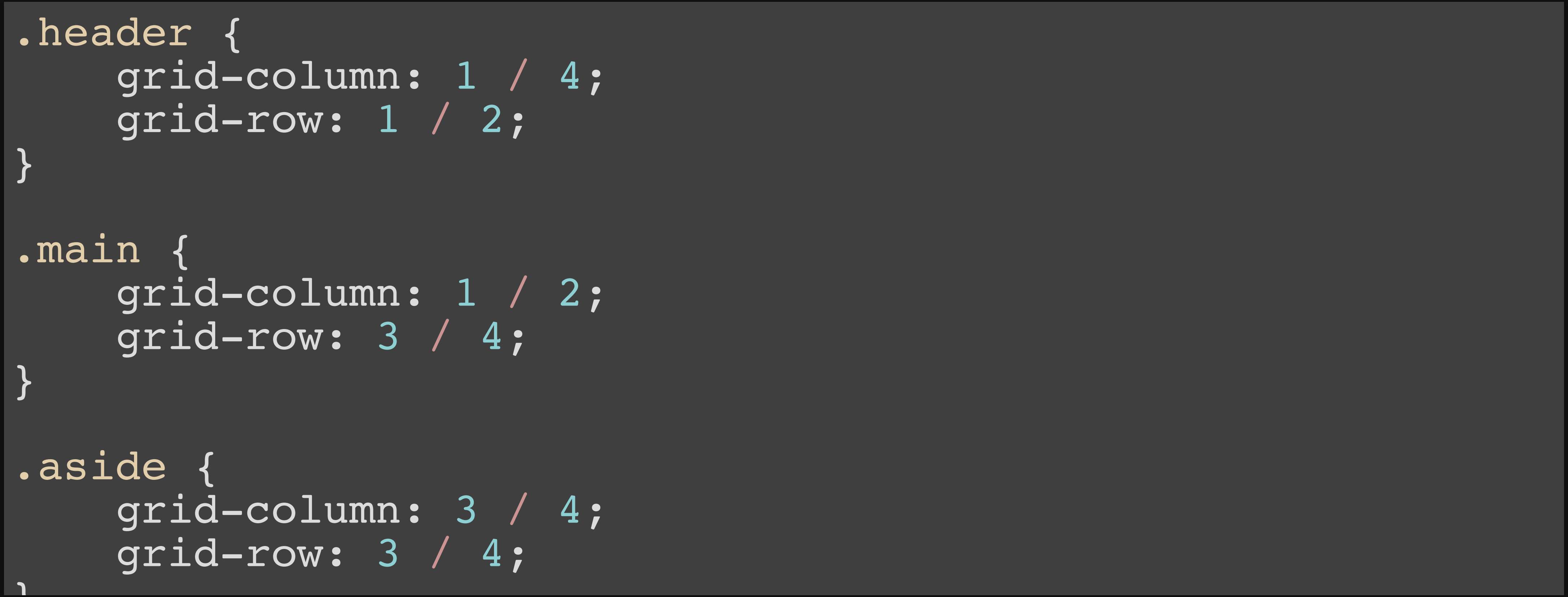
Defining where each content will be placed

CSS:

```css
.header {
    grid-column-start: 1;
    grid-column-end: 4;
    grid-row-start: 1;
    grid-row-end: 2;
}

.main {
    grid-column-start: 1;
    grid-column-end: 2;
    grid-row-start: 3;
    grid-row-end: 4;
}
```

View on CodePen

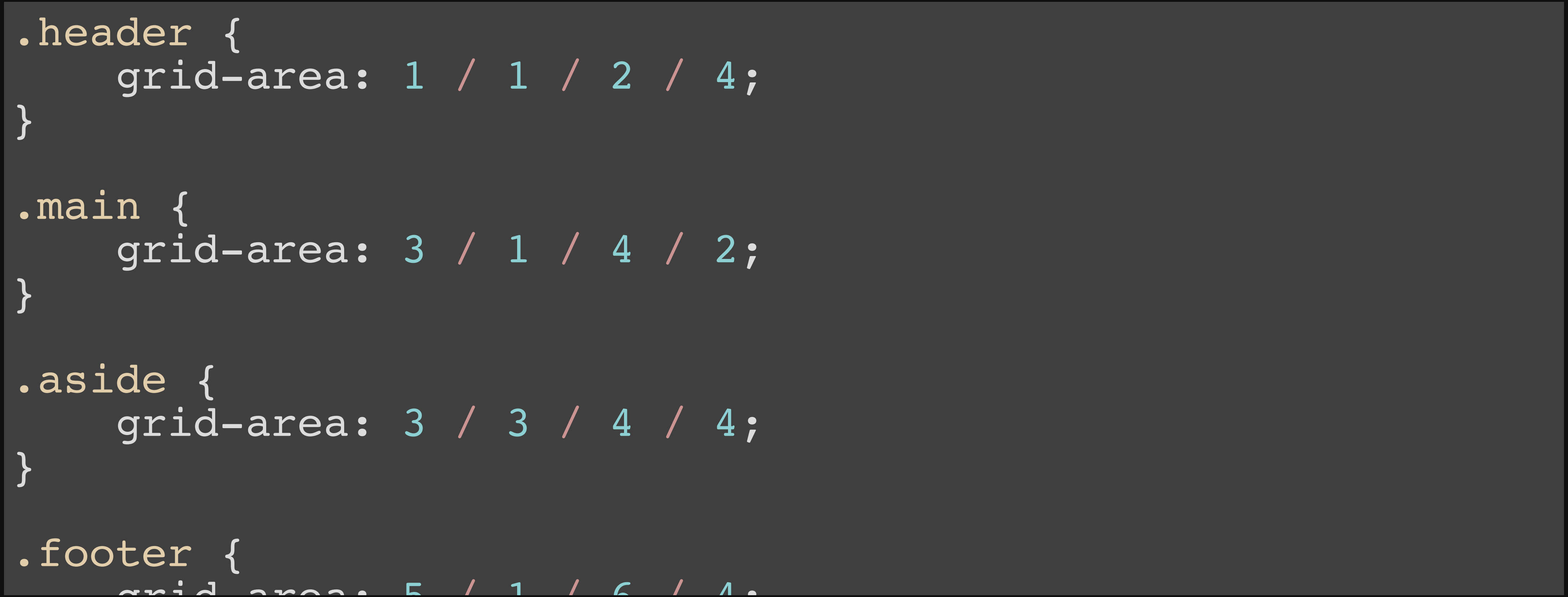
Grid column / grid row simplified:



```
.header {
    grid-column: 1 / 4;
    grid-row: 1 / 2;
}

.main {
    grid-column: 1 / 2;
    grid-row: 3 / 4;
}

.aside {
    grid-column: 3 / 4;
    grid-row: 3 / 4;
}
```

View on CodePen

Grid column / grid row even more simplified:

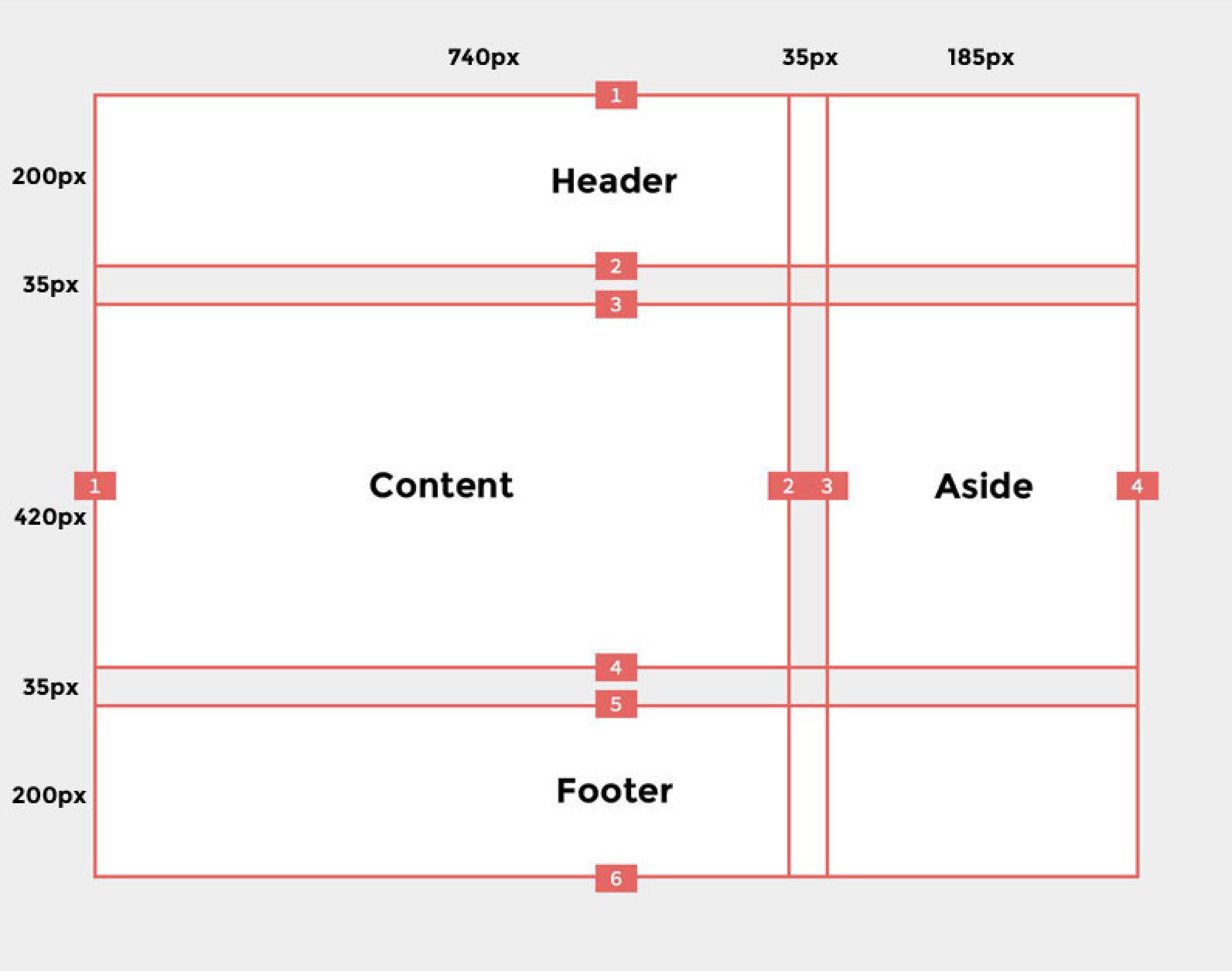


```css
.header {
    grid-area: 1 / 1 / 2 / 4;
}

.main {
    grid-area: 3 / 1 / 4 / 2;
}

.aside {
    grid-area: 3 / 3 / 4 / 4;
}

.footer {
    grid-area: 5 / 1 / 6 / 4;
```

View on CodePen

Grid layout without gaps:

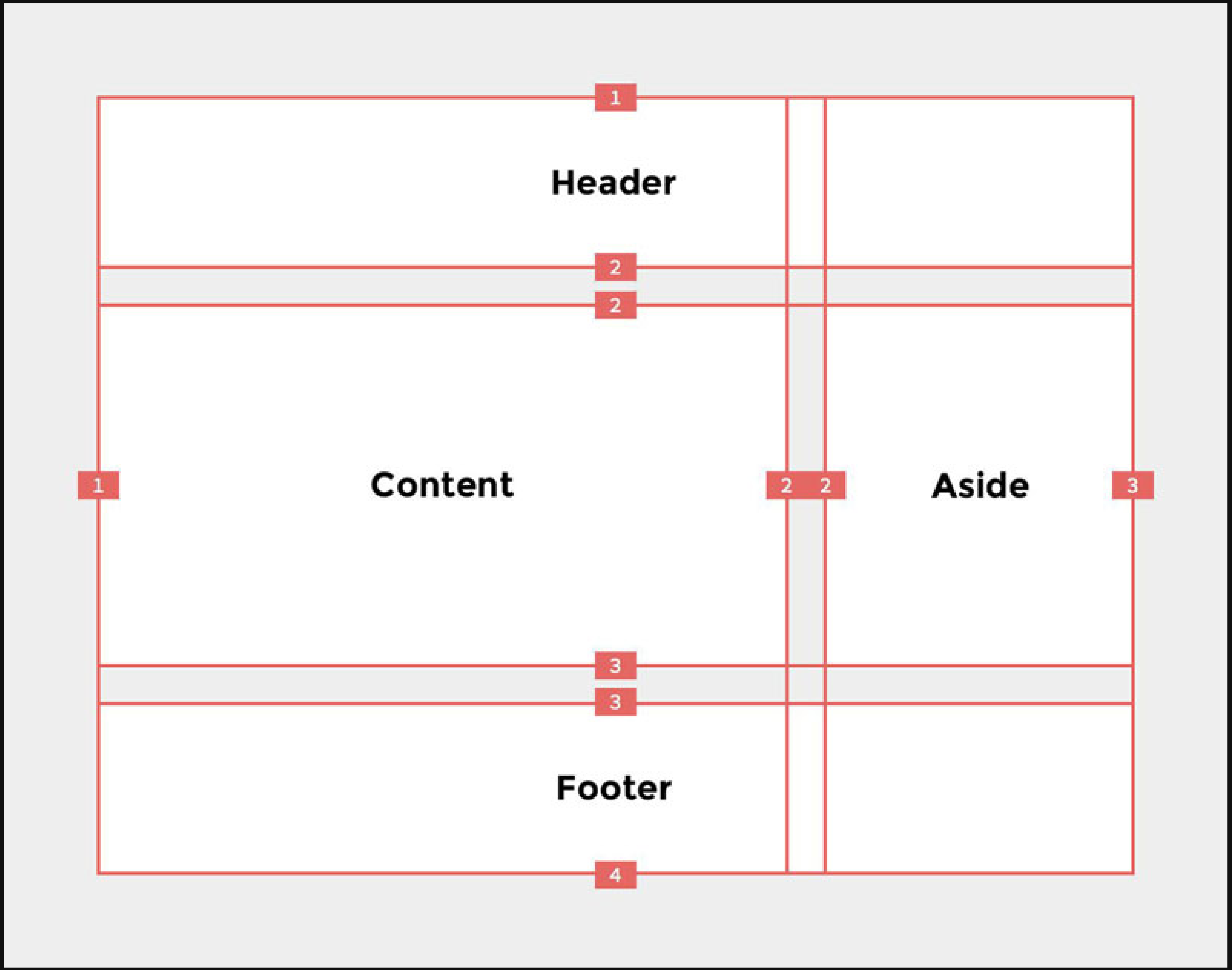
```
.container {
    display: grid;
    grid-template-rows: 200px 35px 420px 35px 200px;
    grid-template-columns: 740px 35px 185px;
    margin: auto;
    max-width: 960px;
}
```


740px
35px
185px
200px
35px
420px
35px
200px
Header
Content
Aside
Footer
1
2
3
4
5
6
1
2
3
4

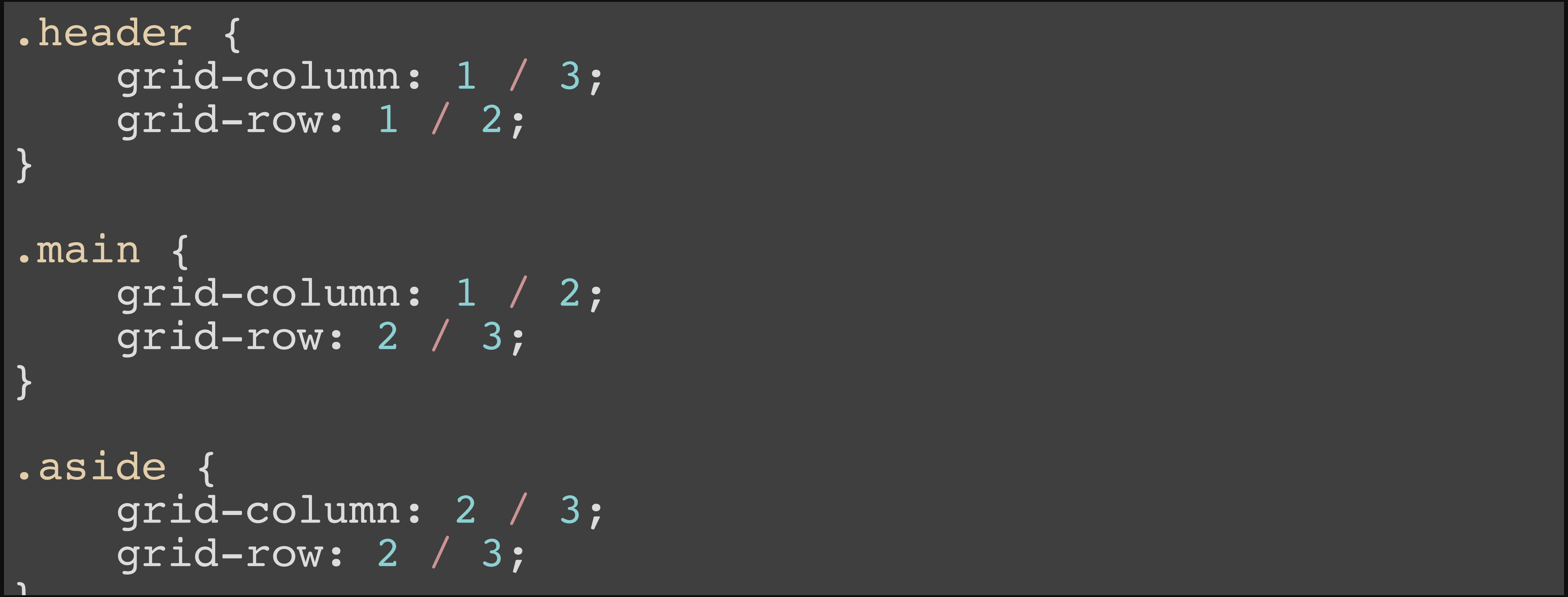
Setting grid layout (with gaps!)

CSS:

```css
.container {
    display: grid;
    grid-template-rows: 200px 420px 200px;
    grid-template-columns: 740px 185px;
    grid-gap: 35px;
    margin: auto;
    max-width: 960px;
}
```


1
Header
2
2
1
Content
2
2
Aside
3
3
3
Footer
4

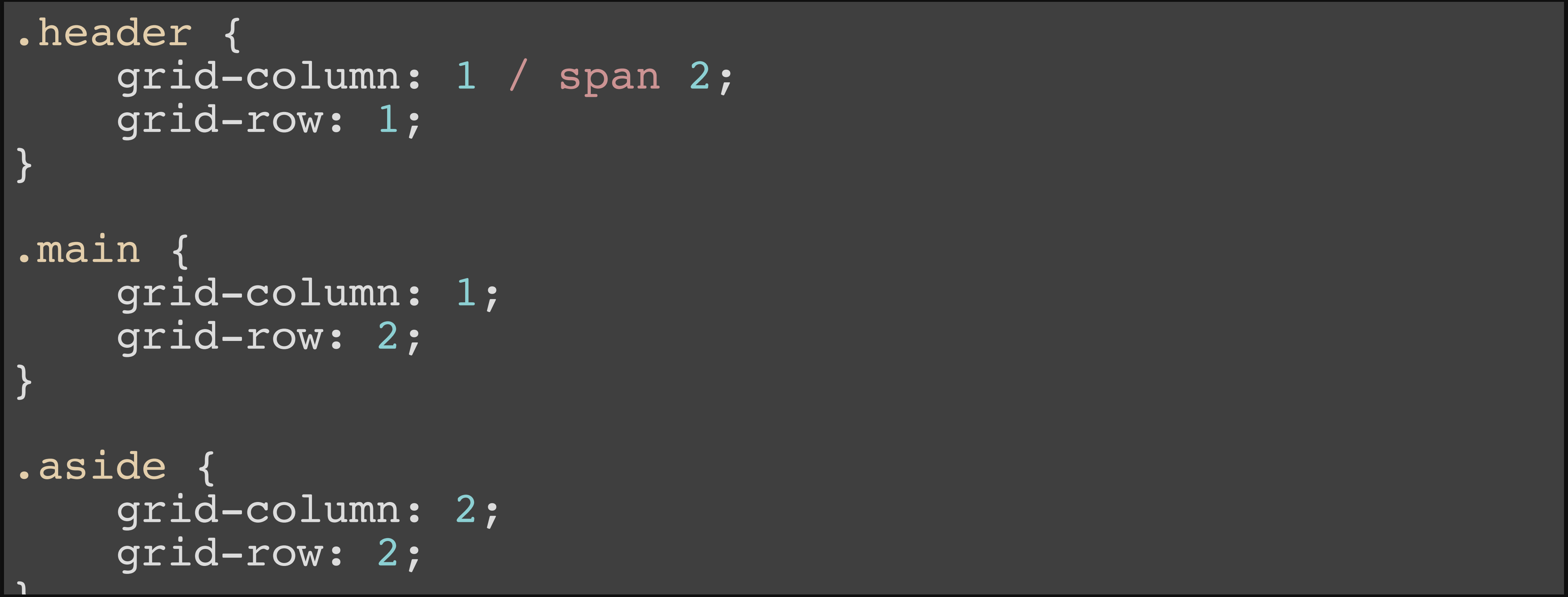
CSS:



```css
.header {
    grid-column: 1 / 3;
    grid-row: 1 / 2;
}

.main {
    grid-column: 1 / 2;
    grid-row: 2 / 3;
}

.aside {
    grid-column: 2 / 3;
    grid-row: 2 / 3;
}
```

View on CodePen

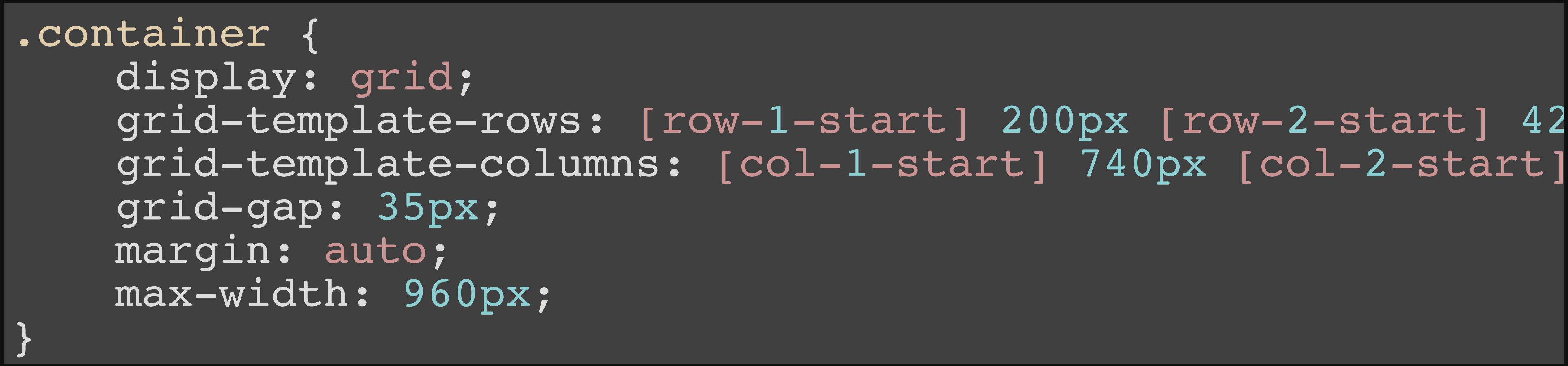
Adding the "span" keyword:



```css
.header {
    grid-column: 1 / span 2;
    grid-row: 1;
}

.main {
    grid-column: 1;
    grid-row: 2;
}

.aside {
    grid-column: 2;
    grid-row: 2;
}
```

View on CodePen

Adding name based grid lines:



```
.container {
    display: grid;
    grid-template-rows: [row-1-start] 200px [row-2-start] 42
    grid-template-columns: [col-1-start] 740px [col-2-start]
    grid-gap: 35px;
    margin: auto;
    max-width: 960px;
}
```

Adding name based grid lines (extended):

```
.header {
    grid-column: col-1-start / col-2-end;
    grid-row: row-1-start;
}

.main {
    grid-column: col-1-start;
    grid-row: row-2-start;
}

.aside {
    grid-column: col-2-start;
    grid-row: row-2-start;
}
```

View on CodePen

Defining the same name for every row/col:

```
.container {
    display: grid;
    grid-template-rows: [row] 200px [row] 420px  [row] 200px
    grid-template-columns: [col] 740px [col] 185px [col];
    grid-gap: 35px;
    margin: auto;
    max-width: 960px;
}
```

Specifying each row/col:

```css
.header {
    grid-column: col 1 / 3;
    grid-row: row;
}

.main {
    grid-column: col 1;
    grid-row: row 2;
}

.aside {
    grid-column: col 2;
    grid-row: row 2;
}
```

View on CodePen

Explicit vs Implicit grid

Grid auto rows/columns

```
.container {
    display: grid;
    grid-template-rows: [row] 200px [row] 420px [row] 200px
    grid-template-columns: [col] 740px [col] 185px;
    grid-gap: 35px;
    grid-auto-rows: 500px;
    grid-auto-columns: 200px;
    margin: auto;
    max-width: 960px;
}
```

View on CodePen

The magic fr unit

```
.container {
    display: grid;
    grid-template-rows: 200px;
    grid-template-columns: 1fr 2fr 100px;
    grid-gap: 10px;
    margin: auto;
    max-width: 1000px;
}
```

View on CodePen

The repeat() function

```css
.container {
    display: grid;
    grid-template-rows: 200px;
    grid-template-columns: repeat(5, 1fr);
    grid-gap: 10px;
    margin: auto;
    max-width: 1000px;
}
```

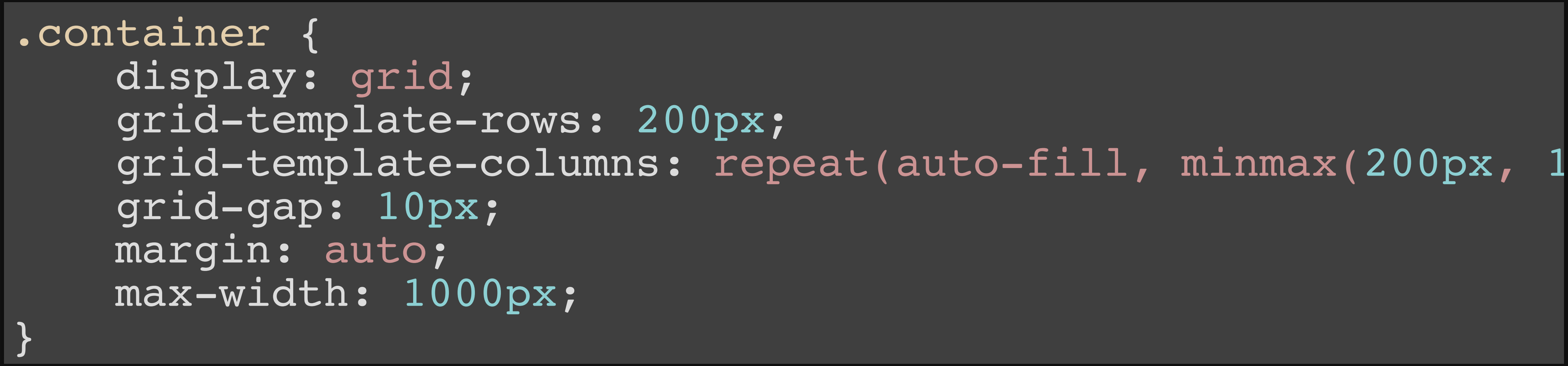
View on CodePen

Grid auto-flow

```css
.container {
    display: grid;
    grid-template-rows: 200px;
    grid-template-columns: repeat(5, 1fr);
    grid-auto-flow: dense;
    grid-gap: 10px;
    margin: auto;
    max-width: 1000px;
}
```

View on CodePen

Auto-fit, auto-fill and the minmax() function



```
.container {
    display: grid;
    grid-template-rows: 200px;
    grid-template-columns: repeat(auto-fill, minmax(200px, 1
    grid-gap: 10px;
    margin: auto;
    max-width: 1000px;
}
```

View on CodePen

Nested Grids

HTML:

```
<div class="container">
    <div class="header">Header</div>
    <div class="main">
        <div class="subcontent subcontent-1">Subcontent 1</d
        <div class="subcontent subcontent-2">Subcontent 2</d
        <div class="subcontent subcontent-3">Subcontent 3</d
    </div>
    <div class="aside">Aside</div>
    <div class="footer">Footer</div>
</div>
```

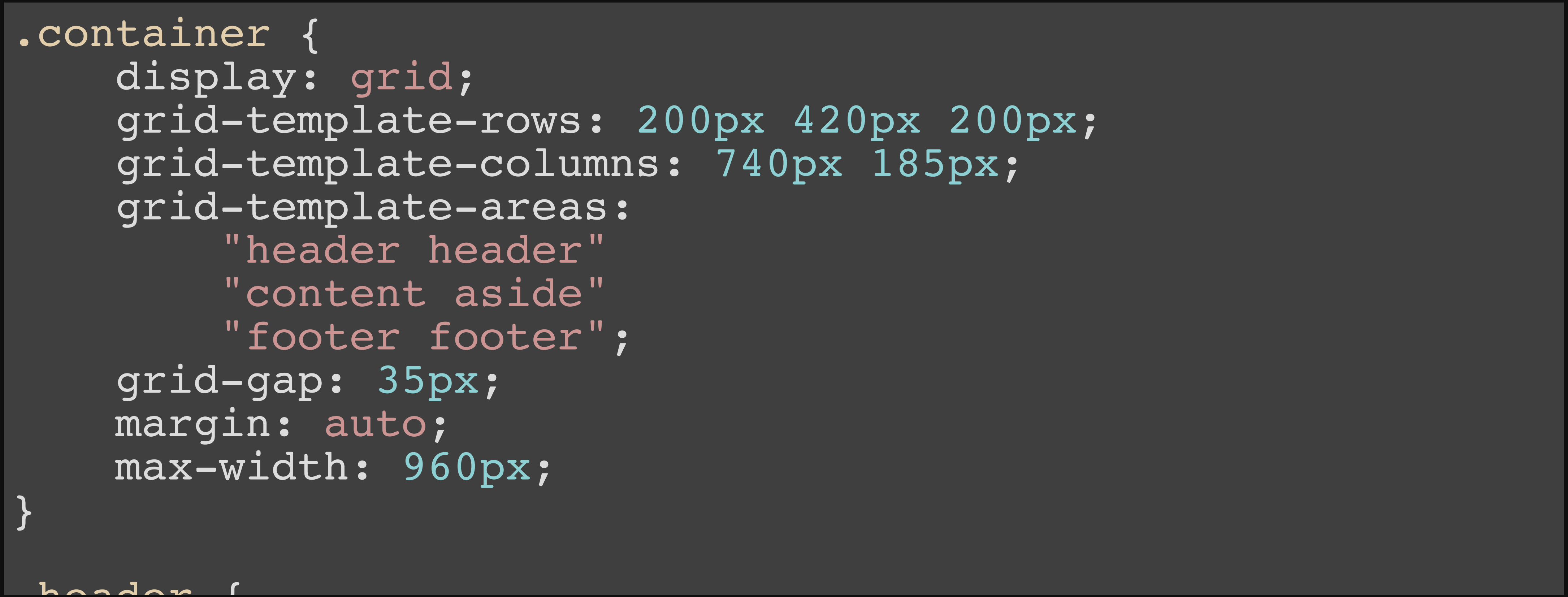
CSS:

```
.main {
    grid-column: col 1;
    grid-row: row 2;
    display: grid;
    padding: 10px;
    grid-gap: 10px;
    grid-template-columns: 1fr 1fr;
}

.subcontent-1 {
    grid-column: 1 / 3;
    grid-row: 1;
}
```

View on CodePen

Grid areas

css:



```
.container {
    display: grid;
    grid-template-rows: 200px 420px 200px;
    grid-template-columns: 740px 185px;
    grid-template-areas:
        "header header"
        "content aside"
        "footer footer";
    grid-gap: 35px;
    margin: auto;
    max-width: 960px;
}

.header {
```

View on CodePen

Defining grid placement based on grid area name

HTML:

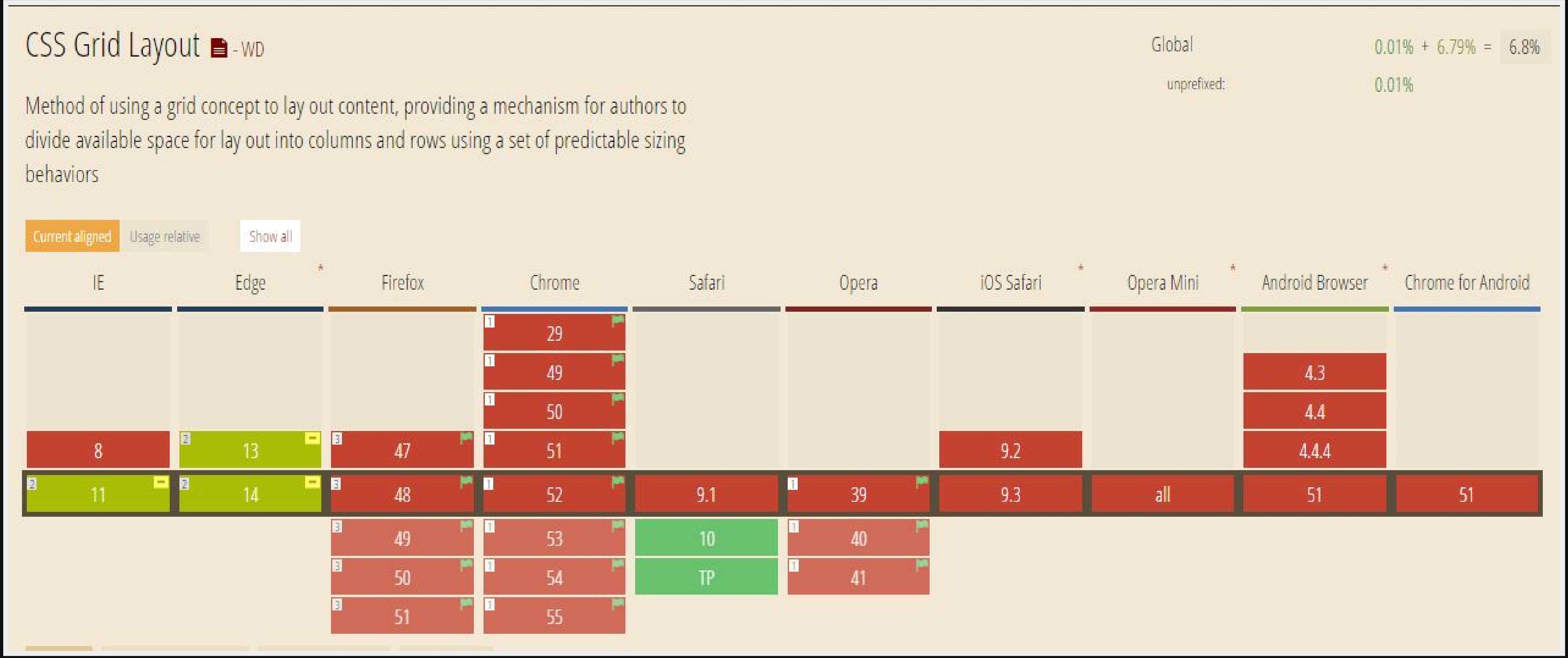
```html
<div class="container">
  <div class="header">Header</div>
  <div class="main">Content</div>
  <div class="aside">Aside</div>
  <div class="footer">Footer</div>
  <div class="overlay">Overlay</div>
</div>
```

CSS:

```css
.container .overlay {
    background-color: #ff0000;
    grid-column: content-start / content-end;
    grid-row: header-start / content-end;
}
```

View on CodePen

Grid support: not much...for now!



CSS Grid Layout - WD

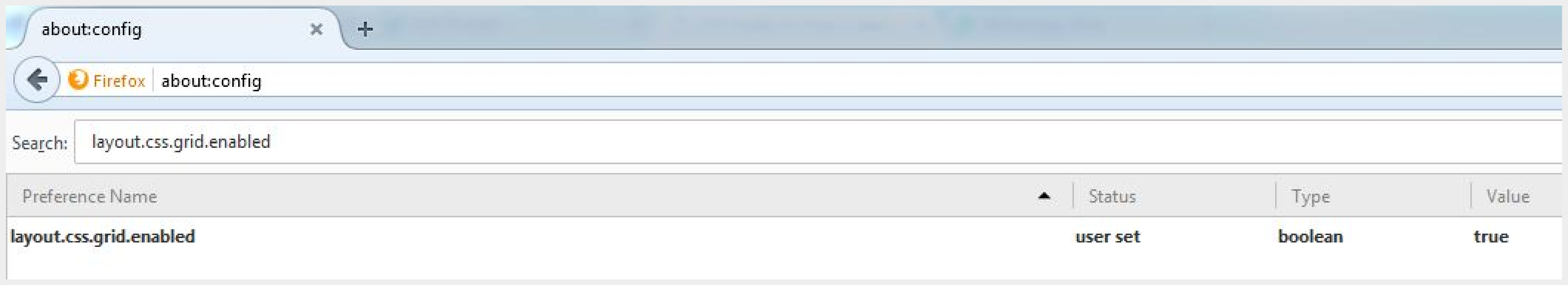
Global 0.01% + 6.79% = 6.8%

unprefixed: 0.01%

Method of using a grid concept to lay out content, providing a mechanism for authors to divide available space for lay out into columns and rows using a set of predictable sizing behaviors

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			[1] 29						
			[1] 49					4.3	
			[1] 50					4.4	
8	[2] 13	[3] 47	[1] 51			9.2		4.4.4	
[2] 11	[2] 14	[3] 48	[1] 52	9.1	[1] 39	9.3	all	51	51
		[3] 49	[1] 53	10	[1] 40				
		[3] 50	[1] 54	TP	[1] 41				
		[3] 51	[1] 55						

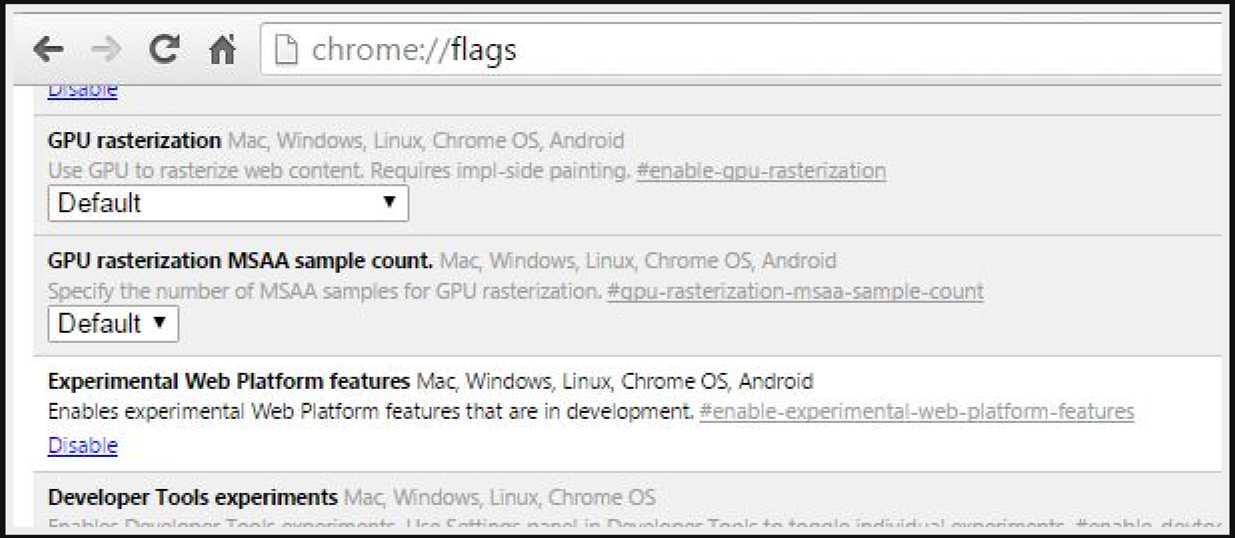
Enabling grid support on the browser: Firefox



Search: layout.css.grid.enabled

Preference Name	Status	Type	Value
layout.css.grid.enabled	user set	boolean	true

Enabling grid support on the browser: Chrome



chrome://flags

Disable

GPU rasterization Mac, Windows, Linux, Chrome OS, Android

Use GPU to rasterize web content. Requires impl-side painting. #enable-gpu-rasterization

Default ▼

GPU rasterization MSAA sample count. Mac, Windows, Linux, Chrome OS, Android

Specify the number of MSAA samples for GPU rasterization. #gpu-rasterization-msaa-sample-count

Default ▼

Experimental Web Platform features Mac, Windows, Linux, Chrome OS, Android

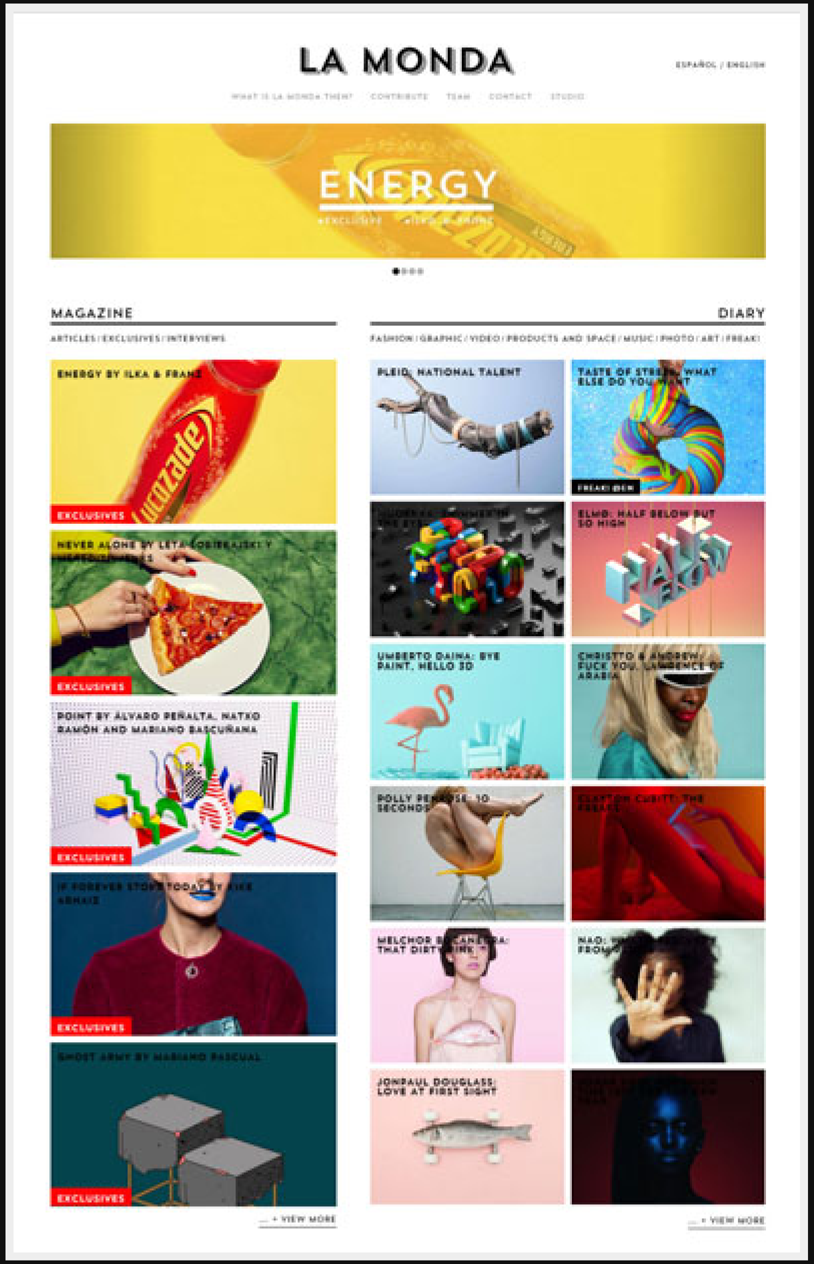
Enables experimental Web Platform features that are in development. #enable-experimental-web-platform-features

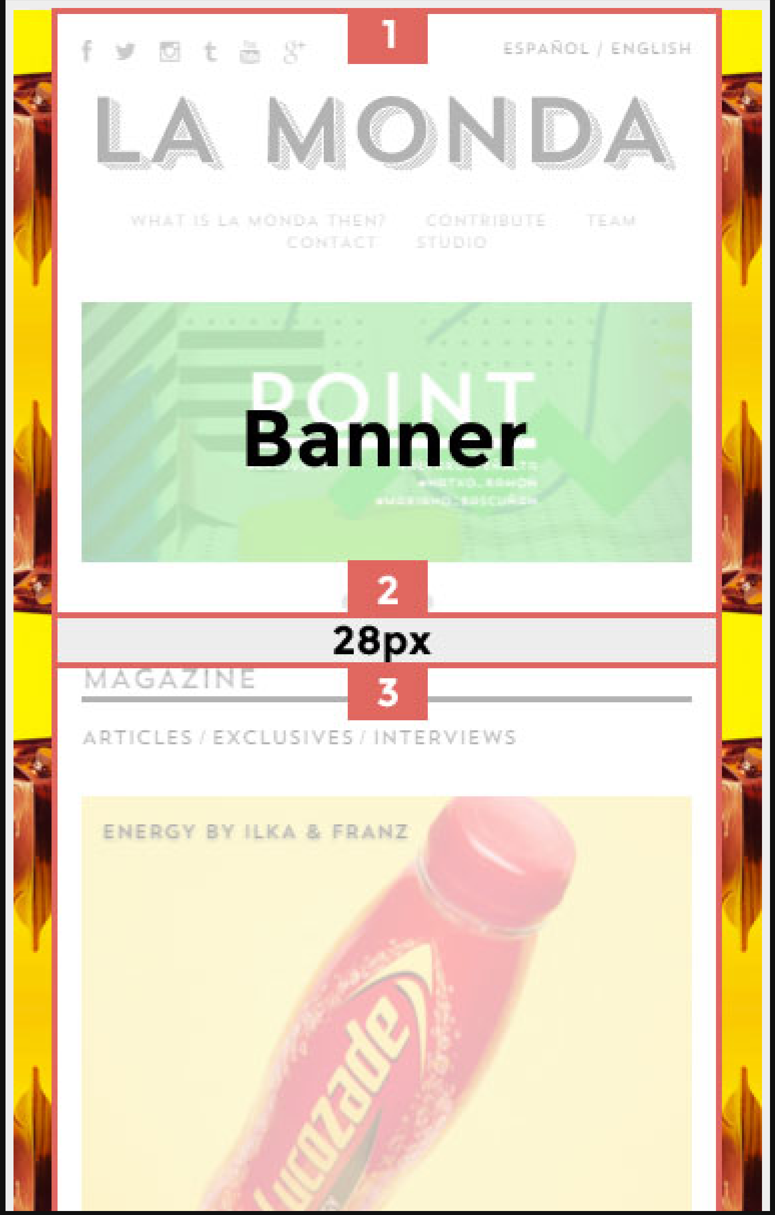
Disable

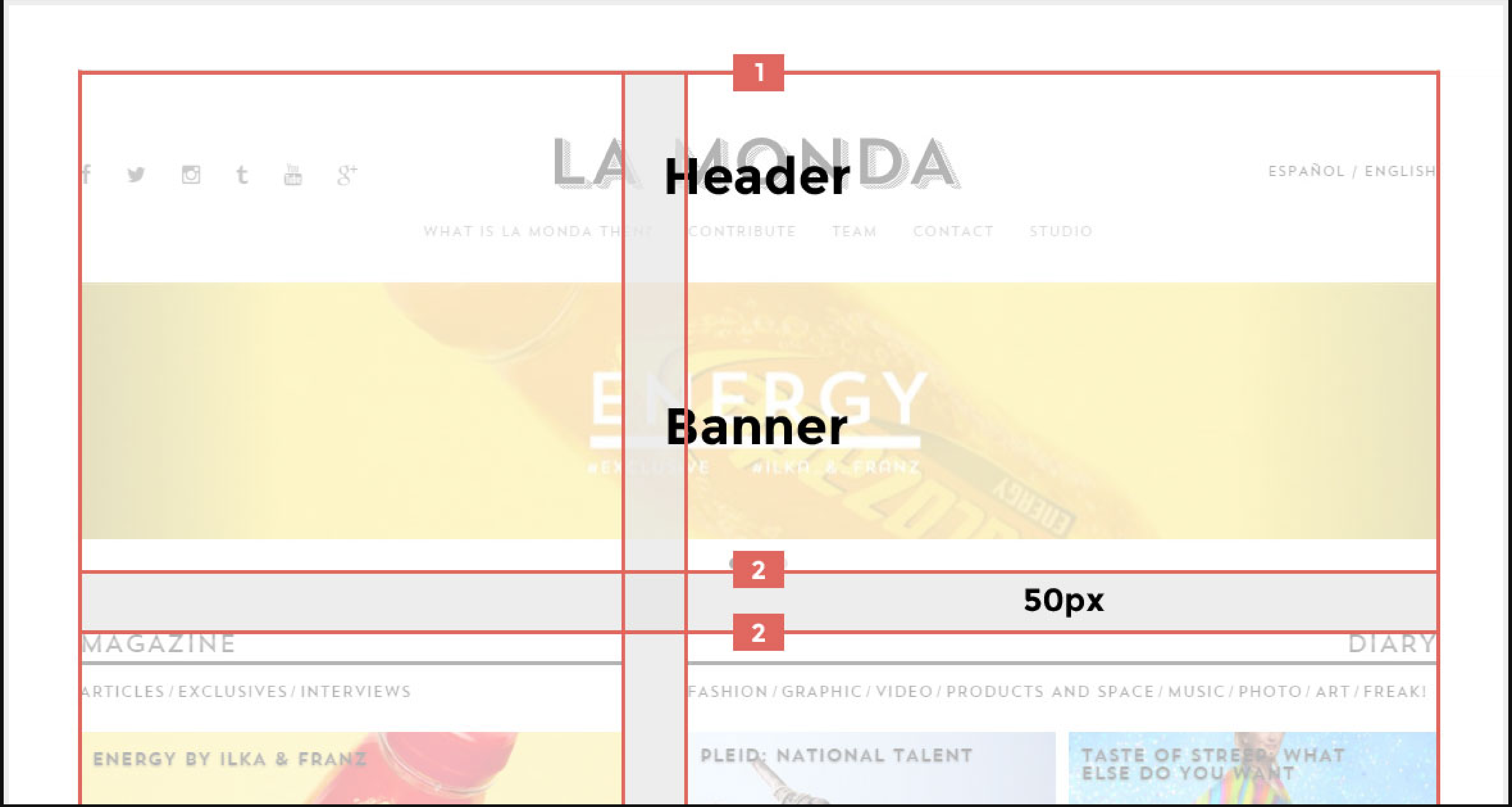
Developer Tools experiments Mac, Windows, Linux, Chrome OS

Enables Developer Tools experiments. Use Settings panel in Developer Tools to toggle individual experiments. #enable-devtoo

So..what about the real world?


LA MONDA
ESPAÑOL / ENGLISH
WHAT IS LA MONDA THEN?
CONTRIBUTE
TEAM
CONTACT
STUDIO
ENERGY
MAGAZINE
ARTICLES / EXCLUSIVES / INTERVIEWS
DIARY
FASHION / GRAPHIC / VIDEO / PRODUCTS AND SPACE / MUSIC / PHOTO / ART / FREAK!
ENERGY BY ILKA & FRANZ
EXCLUSIVES
NEVER ALONE BY LETA SOBIERAJSKI Y
EXCLUSIVES
POINT BY ÁLVARO PEÑALTA, NATXO RAMÓN AND MARIANO BASCUÑANA
EXCLUSIVES
IF FOREVER STOPS TODAY BY KIKE ARNAIZ
EXCLUSIVES
GHOST ARMY BY MARIANO PASCUAL
EXCLUSIVES
PLEID: NATIONAL TALENT
TASTE OF STREET, WHAT ELSE DO YOU WANT
FREAK! @EM
ELMO: HALF BELOW BUT SO HIGH
UMBERTO DAINA: BYE PAINT, HELLO 3D
CHRISTTO & ANDREW: FUCK YOU, LAWRENCE OF ARABIA
POLLY PENROSE: 10 SECONDS
CLAYTON CUBITT: THE FREAKS
MELCHOR BACANEGRA: THAT DIRTY LINE
NAO:
JONPAUL DOUGLASS: LOVE AT FIRST SIGHT
+ VIEW MORE
+ VIEW MORE


1
ESPAÑOL / ENGLISH
LA MONDA
WHAT IS LA MONDA THEN?
CONTRIBUTE
TEAM
CONTACT
STUDIO
Banner
2
28px
MAGAZINE
3
ARTICLES / EXCLUSIVES / INTERVIEWS
ENERGY BY ILKA & FRANZ


1
LA MONDA
Header
ESPAÑOL / ENGLISH
WHAT IS LA MONDA THEN?
CONTRIBUTE
TEAM
CONTACT
STUDIO
ENERGY
Banner
#EXCLUSIVE
#ILKA_&_FRANZ
2
50px
2
MAGAZINE
DIARY
ARTICLES / EXCLUSIVES / INTERVIEWS
FASHION / GRAPHIC / VIDEO / PRODUCTS AND SPACE / MUSIC / PHOTO / ART / FREAK!
ENERGY BY ILKA & FRANZ
PLEID: NATIONAL TALENT
TASTE OF STREET: WHAT ELSE DO YOU WANT

Files on https://github.com/wiznia/grids

Download talk on PDF at: https://raw.githubusercontent.com/wiznia/grids/master/css-grids.pdf

Codepen examples from this talk at: http://codepen.io/wiznia/

My Twitter: @wiznia

Thank you!