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 R/GA, 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!

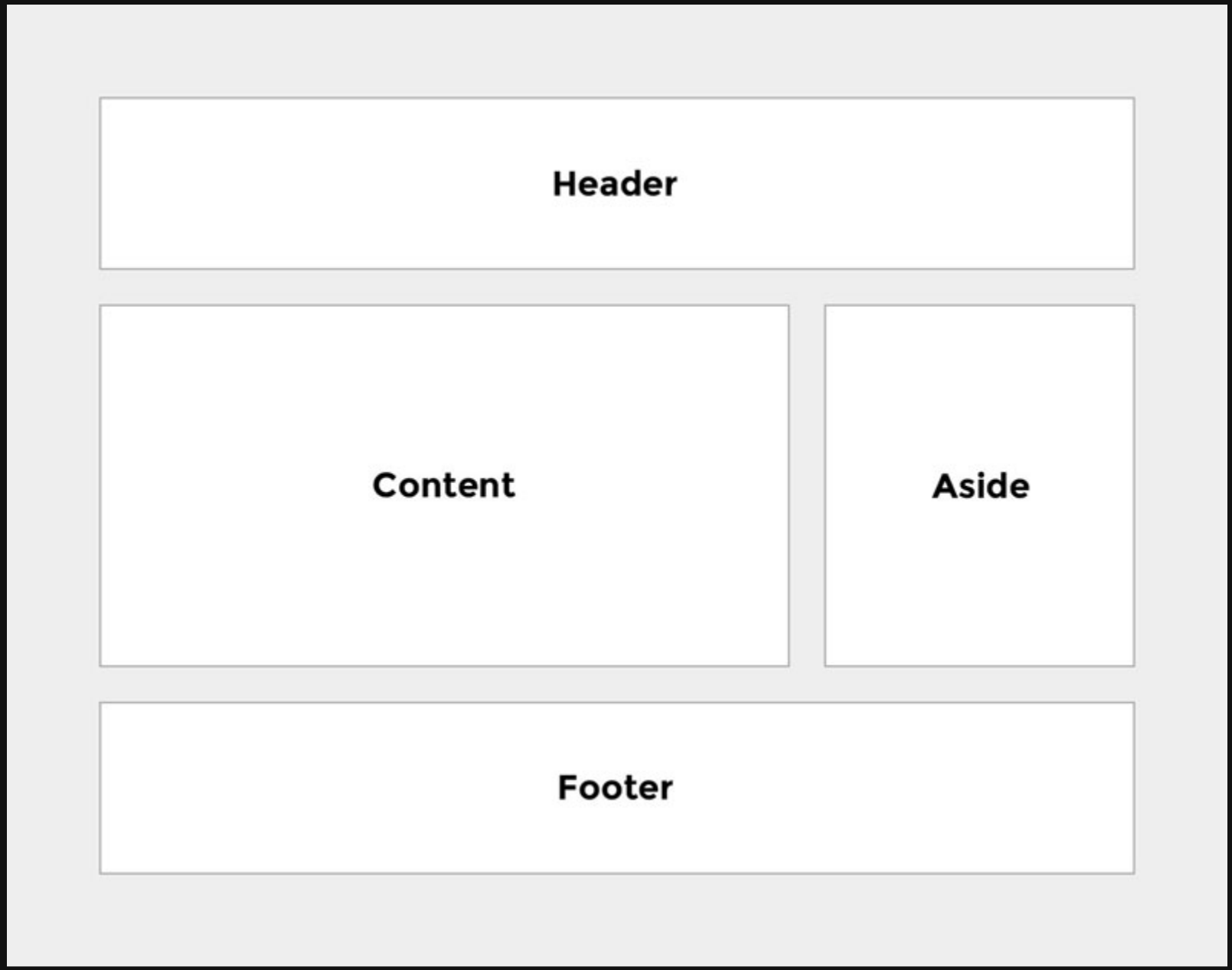
So...what's the future?

GRIDS!

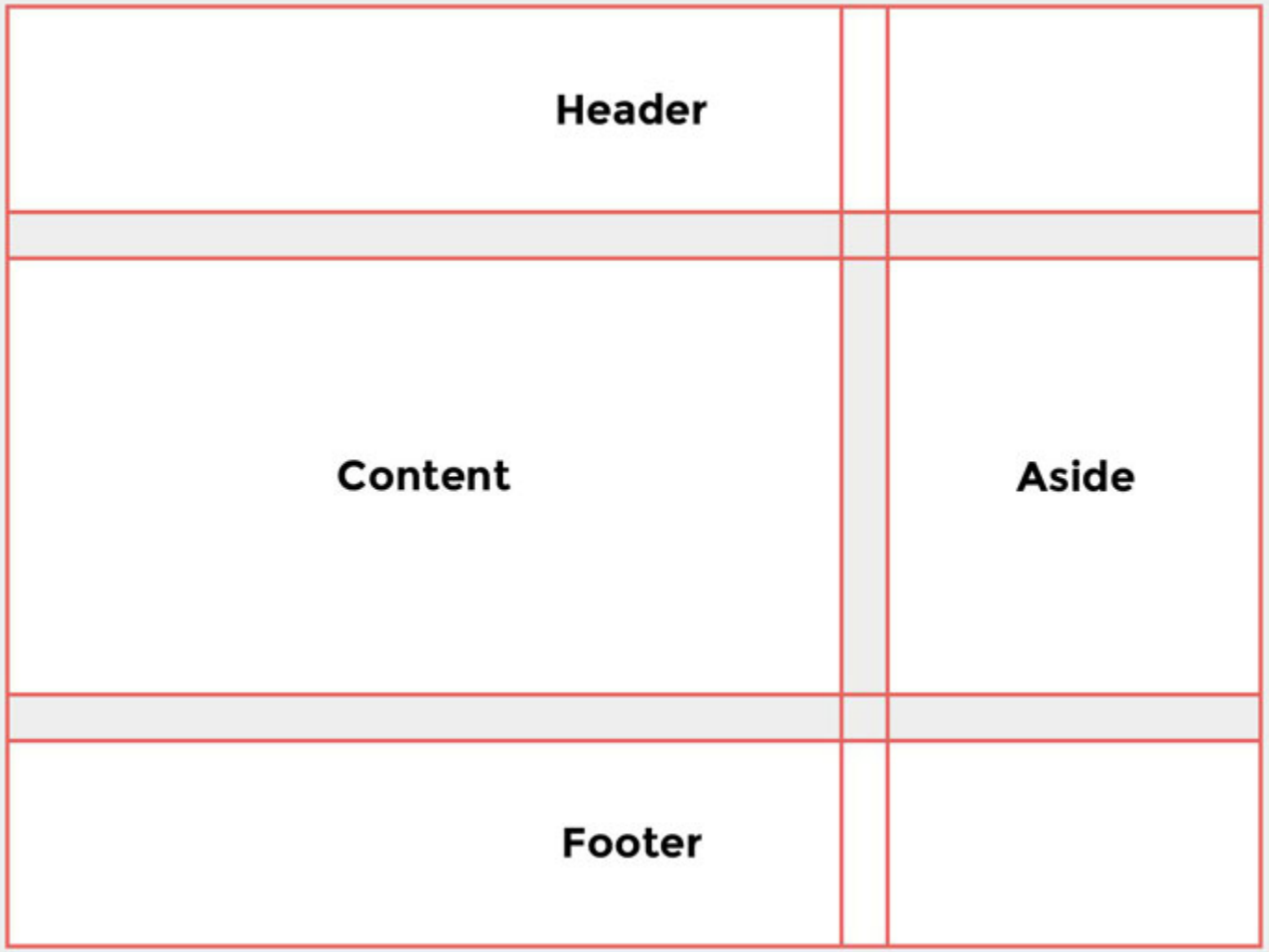
Grids are a new way of organizing a layout via CSS on a page

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

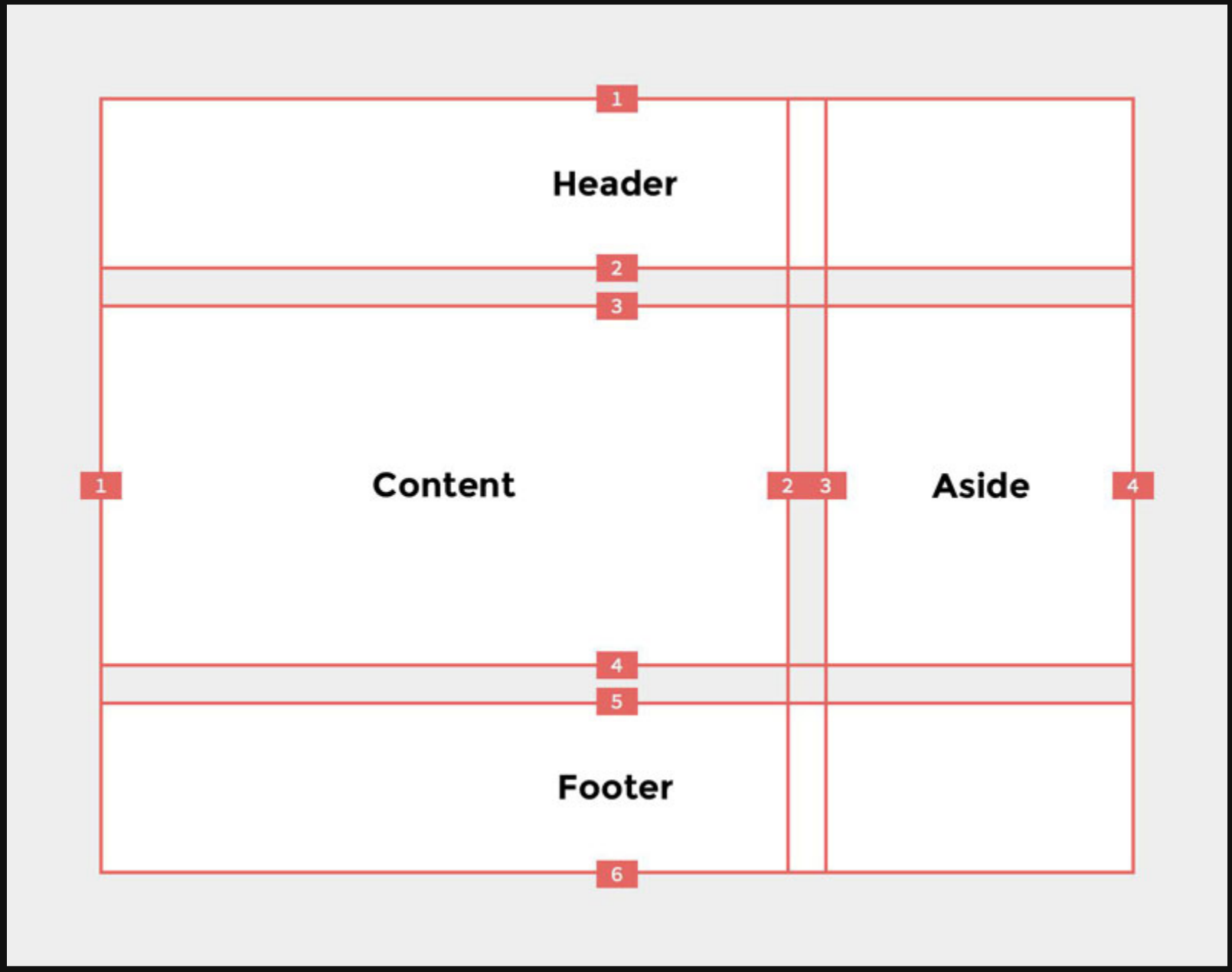
Standard grid



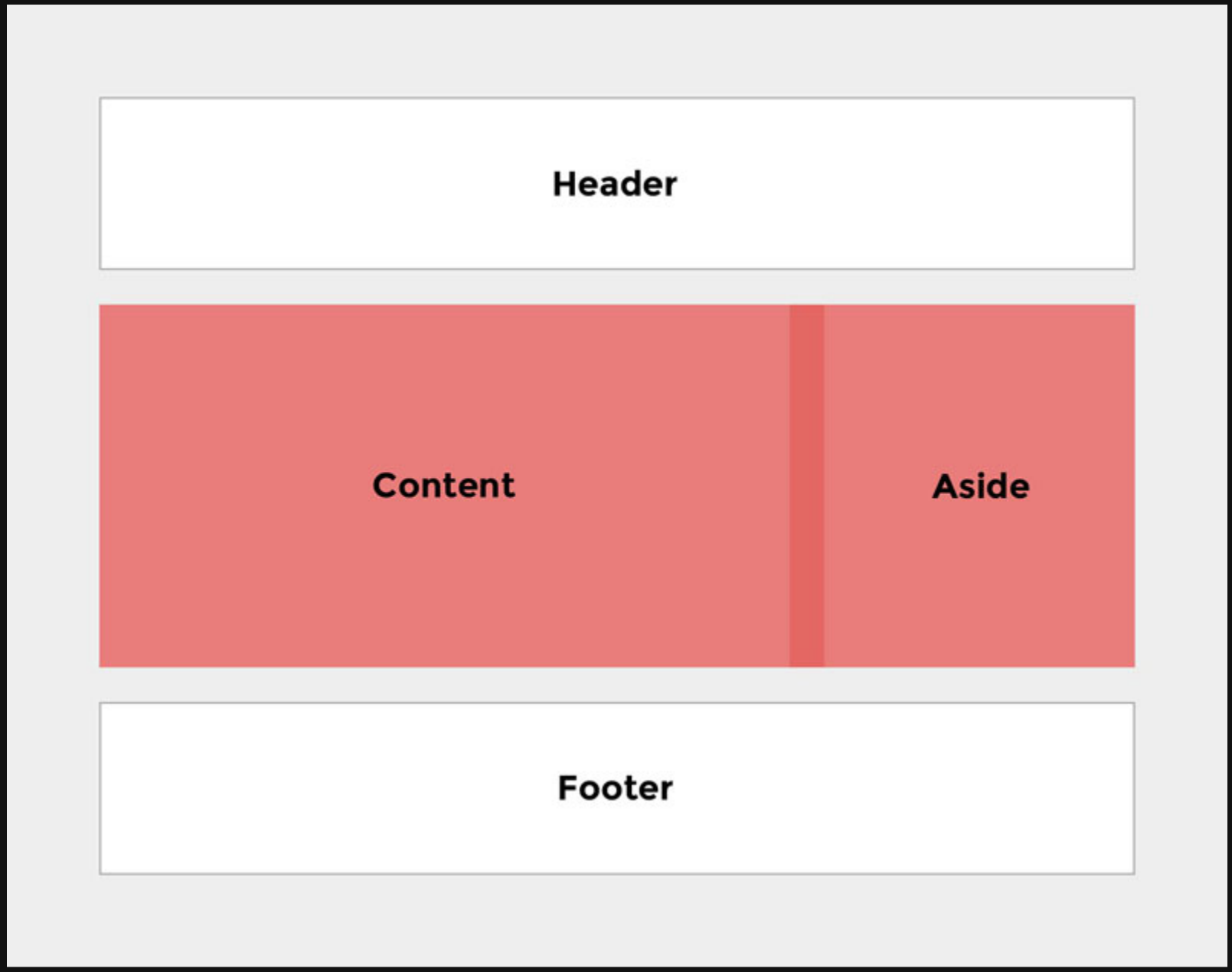
Grid lines



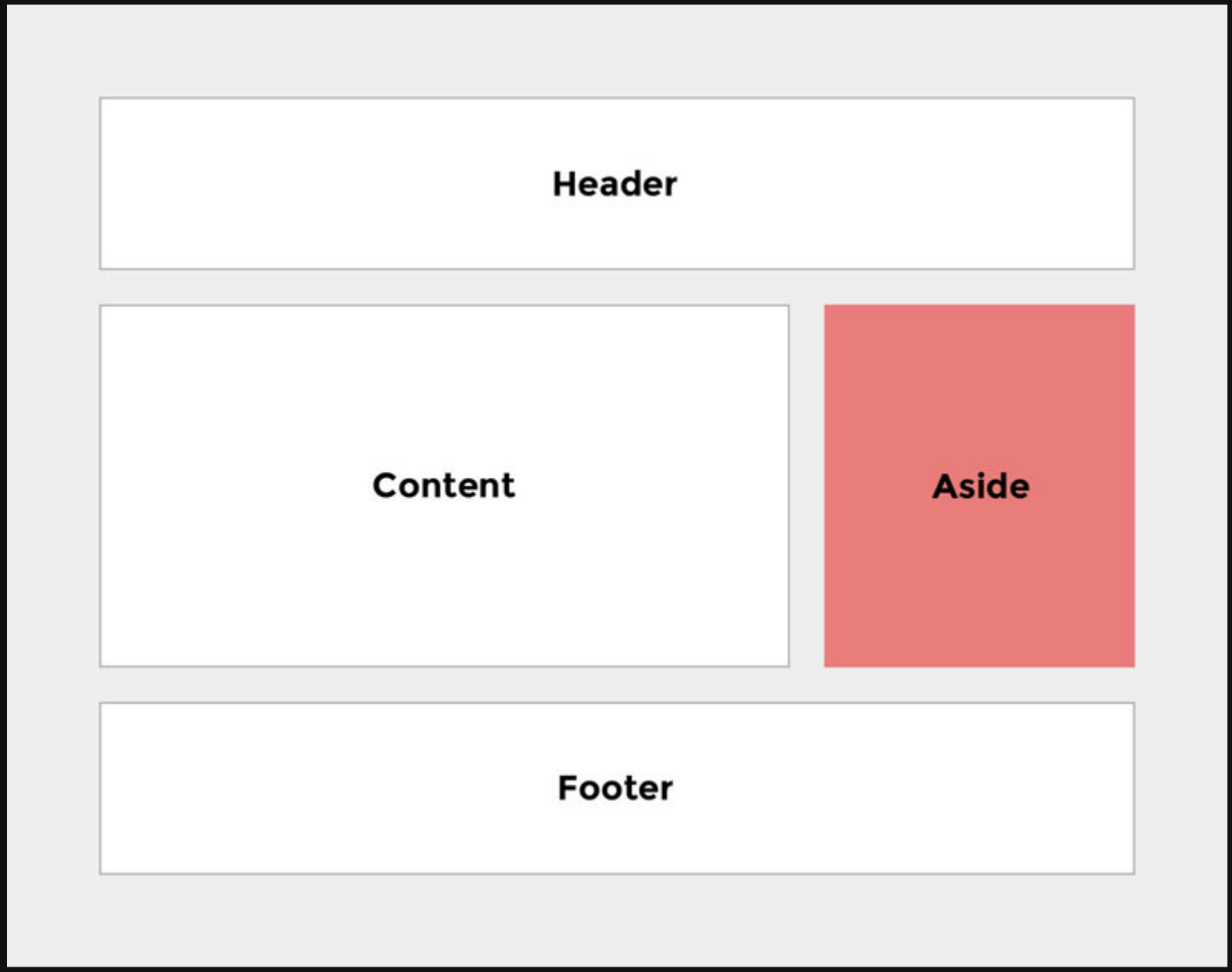
Grid lines (highlighted)



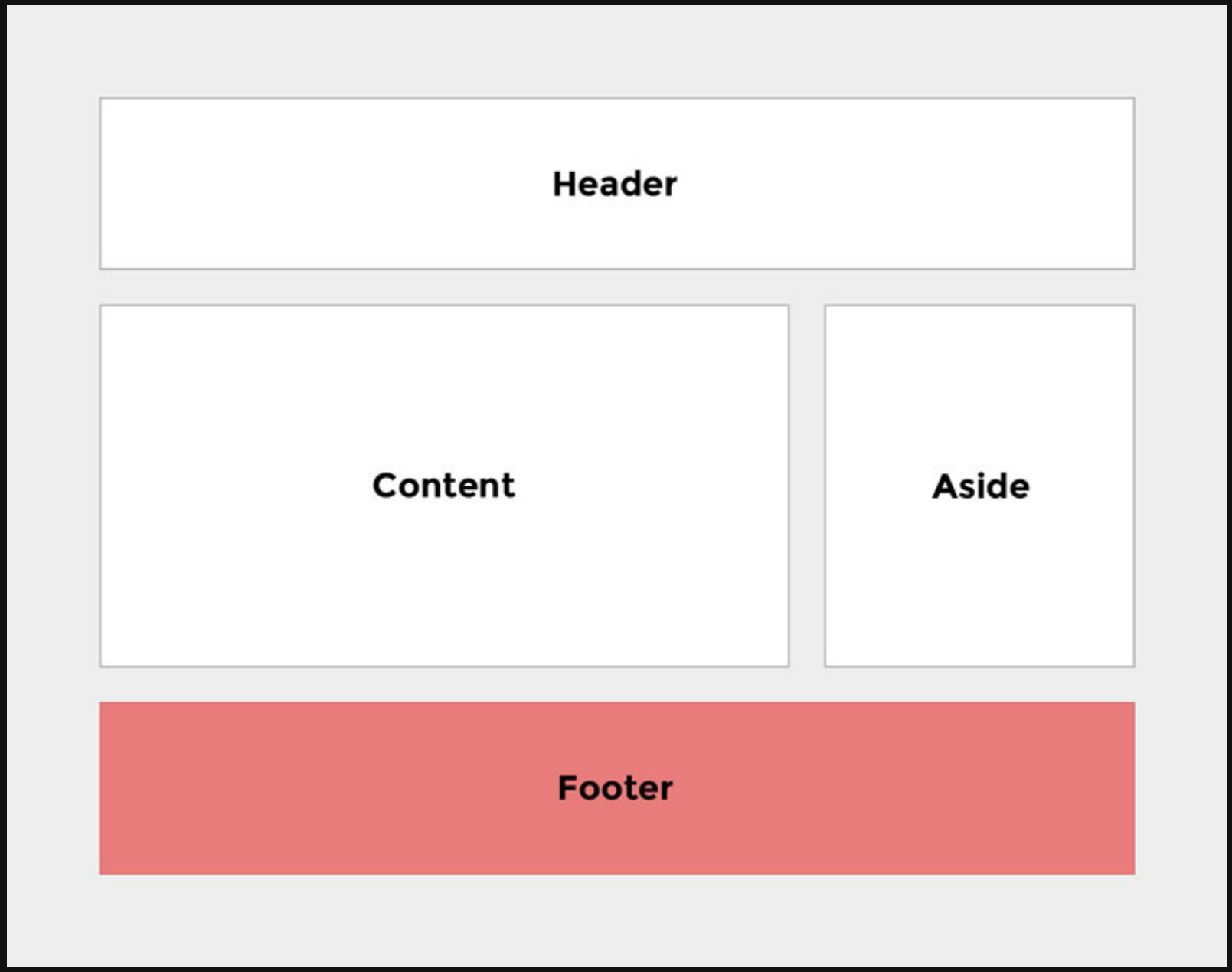
Grid track



Grid cell



Grid area



Defining a grid

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:

```
.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;
}
```

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;
}

.aside {
    grid-column-start: 3;
    grid-column-end: 4;
    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;
}

.footer {
    grid-column: 1 / 4;
    grid-row: 5 / 6;
}
```

View on CodePen

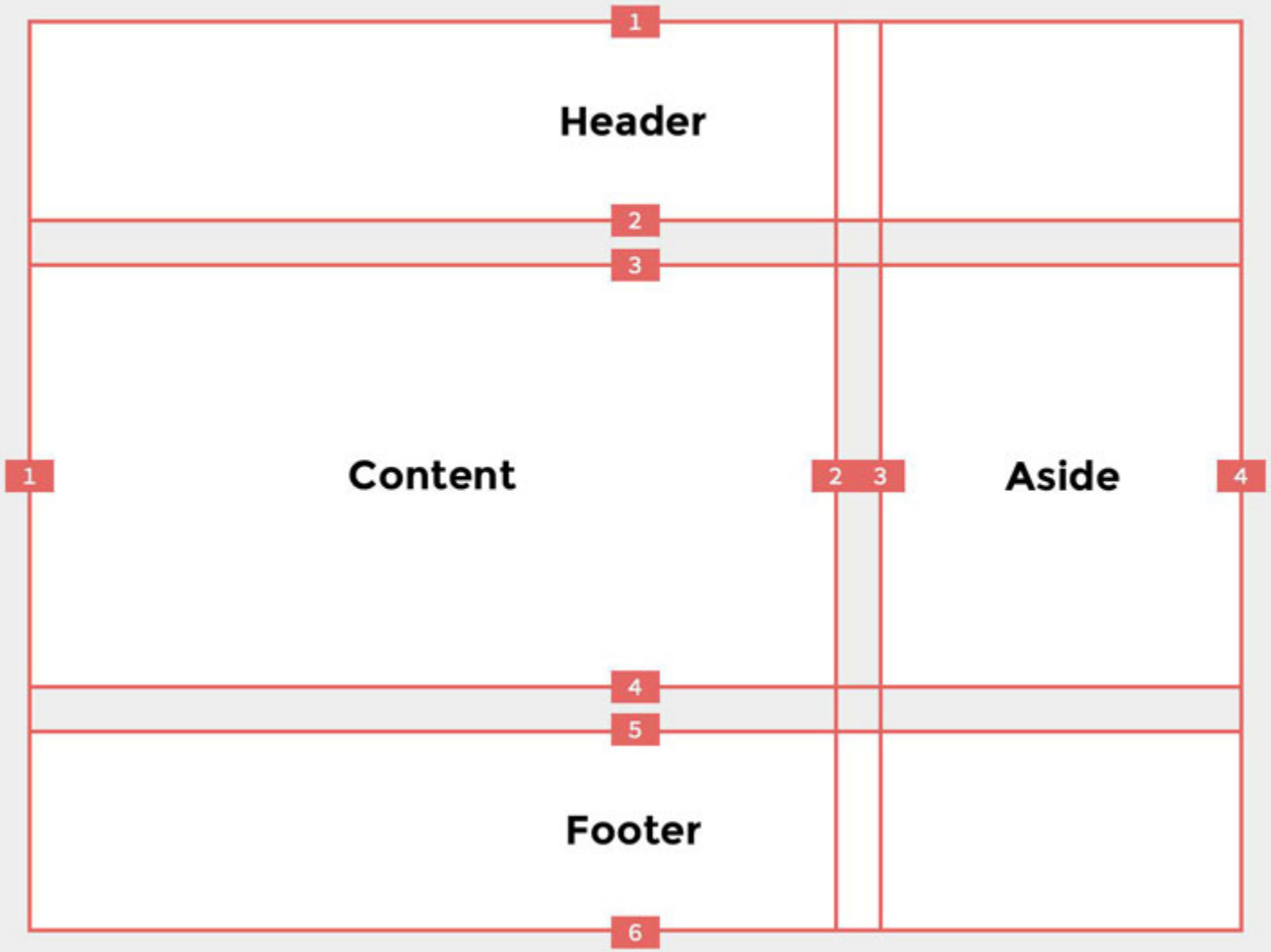
Grid column / grid row even more simplified:

```
.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:

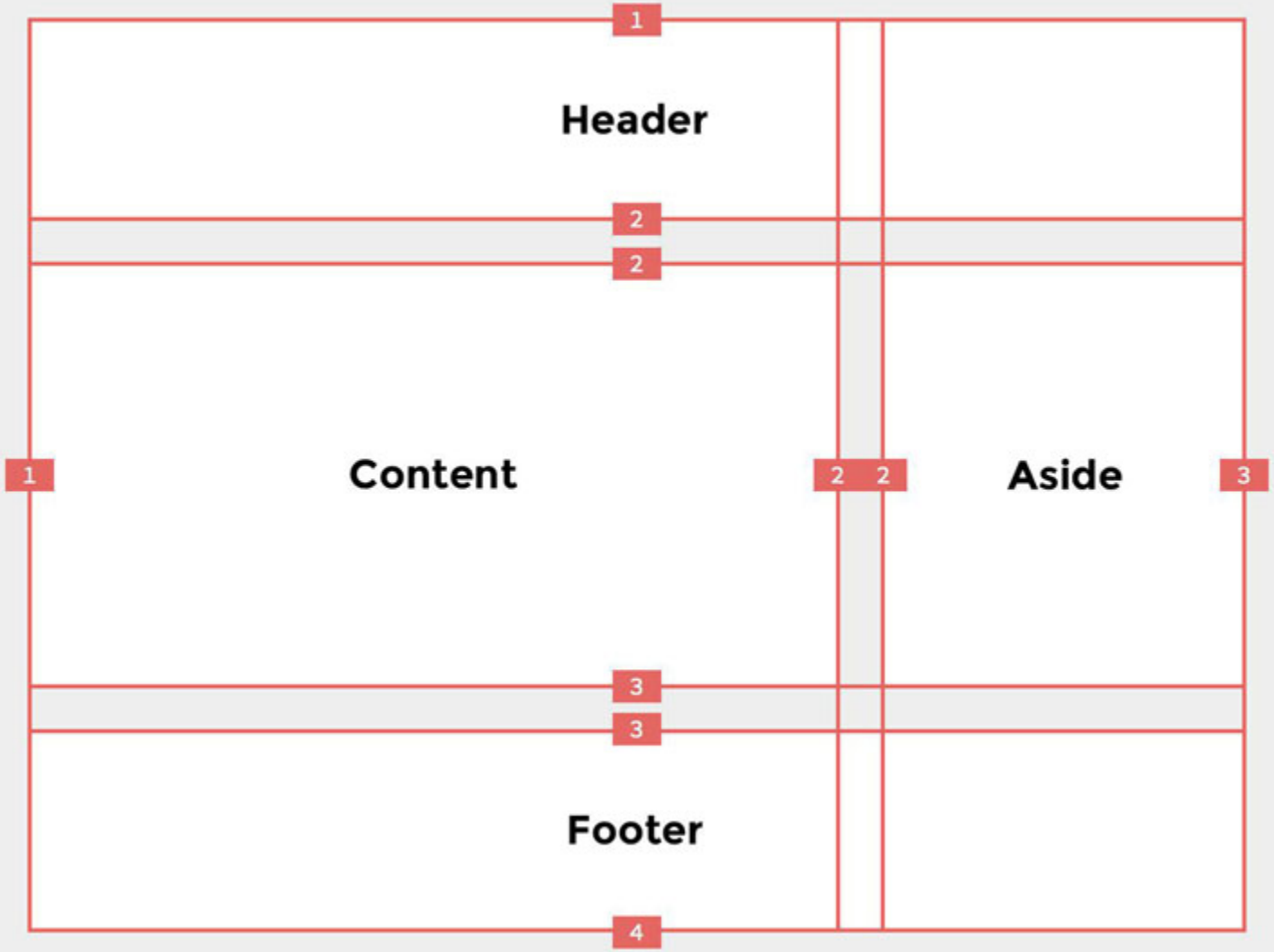
```
.container {
    display: grid;
    grid-template-rows: 200px 35px 420px 35px 200px;
    grid-template-columns: 740px 35px 185px;
    margin: auto;
    max-width: 960px;
}
```


1
Header
2
3
1
Content
2
3
Aside
4
4
5
Footer
6

Setting grid layout (with gaps!)

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;
}

.footer {
    grid-column: 1 / 3;
    grid-row: 3 / 4;
}
```

View on CodePen

Adding the "span" keyword:

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

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

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

.footer {
    grid-column: 1 / span 2;
    grid-row: 3;
}
```

View on CodePen

Adding name based grid lines:

```
.container {
    display: grid;
    grid-template-rows: [row-1-start] 200px [row-2-start] 420px  [row-3-start] 200px [r
    grid-template-columns: [col-1-start] 740px [col-2-start] 185px [col-2-end];
    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;
}

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

View on CodePen

Defining the same name for every row/col:

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

Specifying each row/col:

```
.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;
}

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

View on CodePen

Explicit vs Implicit grid

Nested Grids

HTML:

```
<div class="container">
    <div class="header">Header</div>
    <div class="main">
        <div class="subcontent subcontent-1">Subcontent 1</div>
        <div class="subcontent subcontent-2">Subcontent 2</div>
        <div class="subcontent subcontent-3">Subcontent 3</div>
    </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: 360px 360px;
}

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

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

.subcontent-3 {
    grid-column: 2;
```

View on CodePen

Grid auto rows/columns

```
.container {
    display: grid;
    grid-template-rows: [row] 200px [row] 420px [row] 200px [row];
    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

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 {
    grid-area: header;
}

.main {
    grid-area: content;
}
```

View on CodePen

Defining grid placement based on grid area name

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:

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

View on CodePen

Align items / Justify items

There are 4 possible values for this properties:

- start
- end
- stretch
- center

```
.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;
    align-items: center;
}
```

View on CodePen

```
.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;
    align-items: start;
}
```

View on CodePen

```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;
    align-items: end;
}
```

View on CodePen

```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;
    align-items: stretch;
}
```

View on CodePen

Align items / Justify items

```
.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;
    justify-items: center;
}
```

View on CodePen

Align items / Justify items

```
.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;
    justify-items: start;
}
```

View on CodePen

Align items / Justify items

```
.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;
    justify-items: end;
}
```

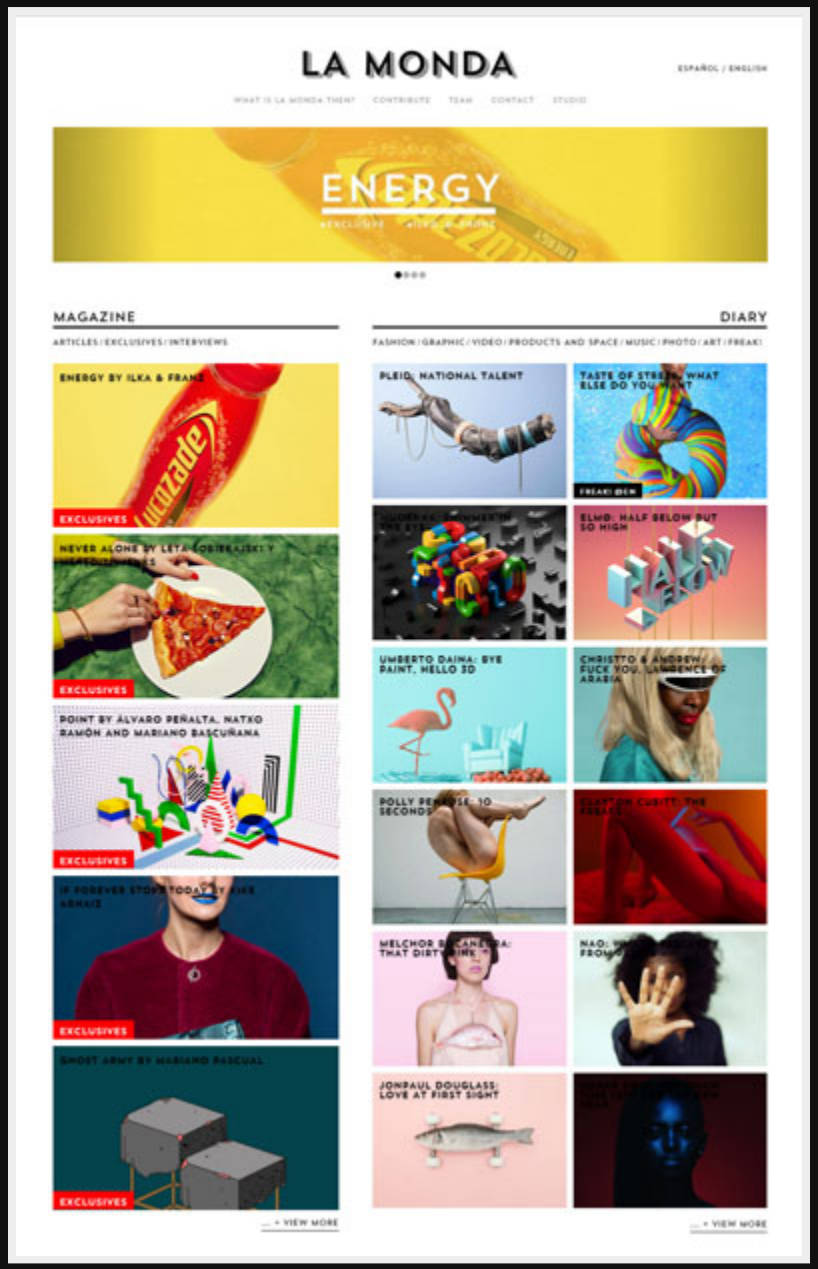
View on CodePen

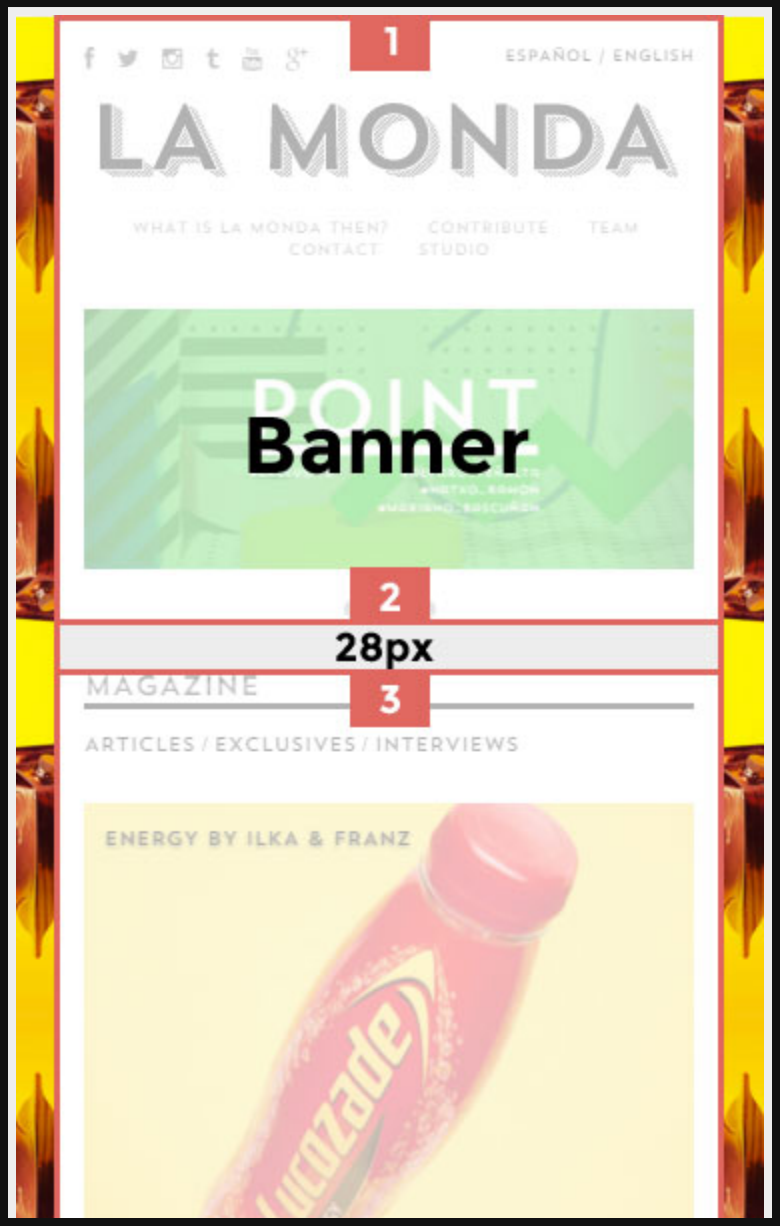
Align items / Justify items

```
.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;
    justify-items: stretch;
}
```

View on CodePen

So..what about the real world?




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

Header

LA MONDA

ESPAÑOL / ENGLISH

WHAT IS LA MONDA THEN? CONTRIBUTE TEAM CONTACT STUDIO

Banner

2

50px

2

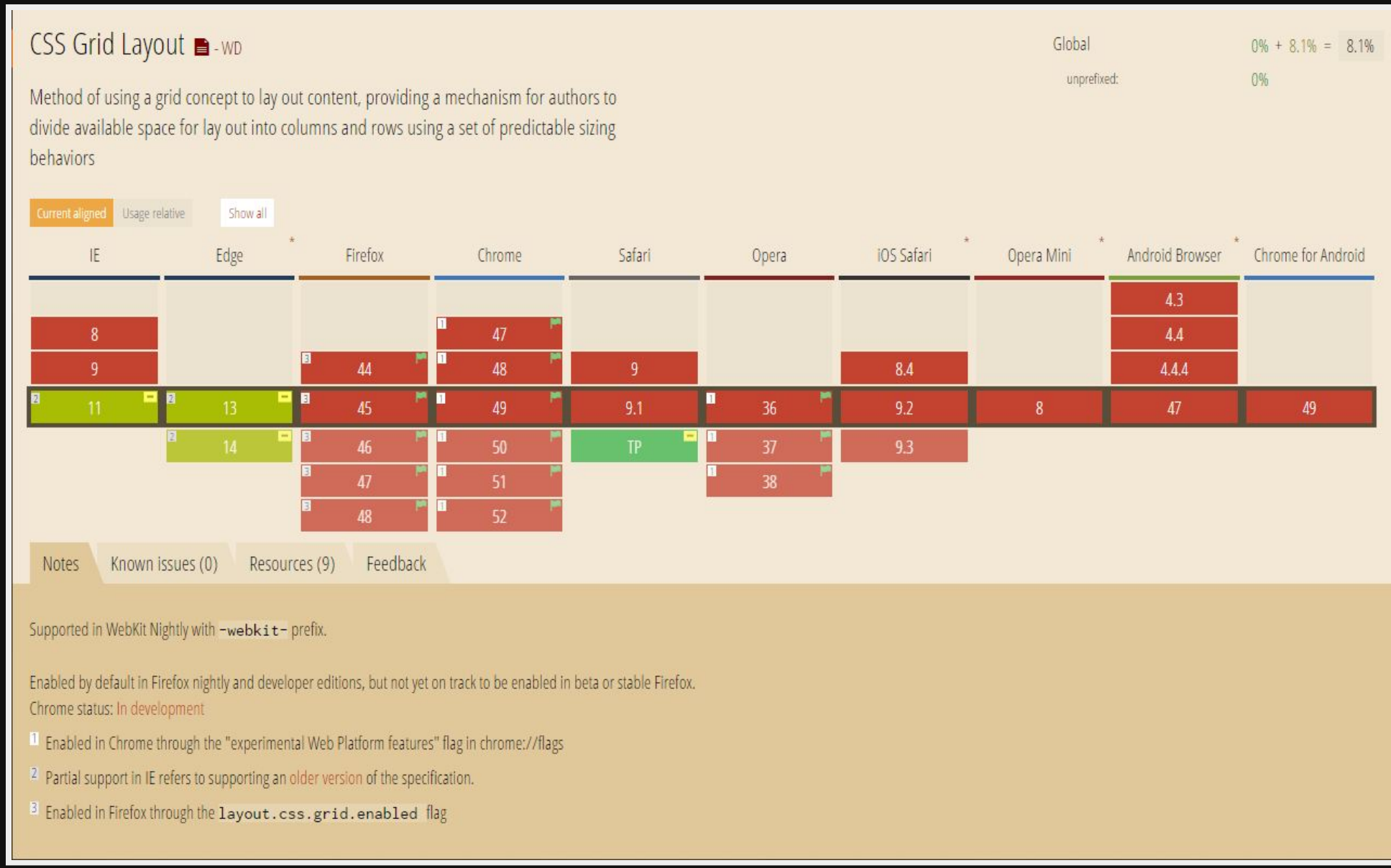
MAGAZINE

DIARY

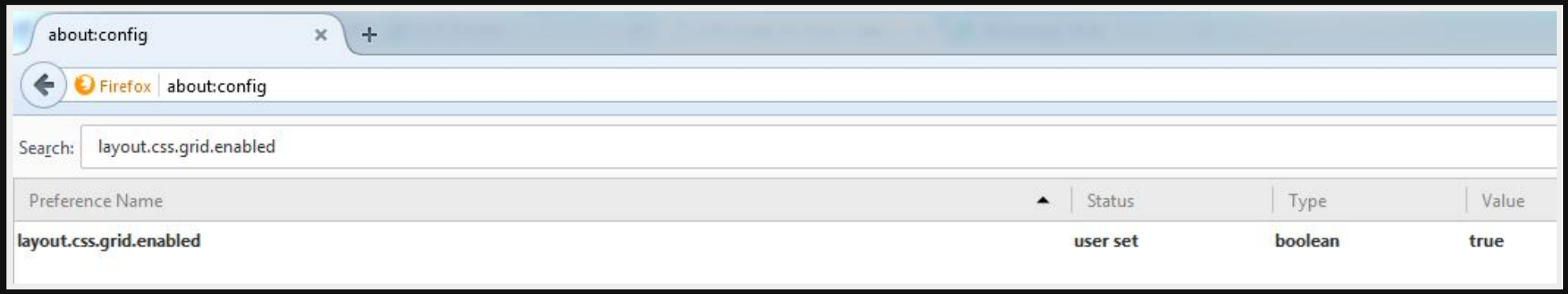
ARTICLES / EXCLUSIVES / INTERVIEWS

FASHION / GRAPHIC / VIDEO / PRODUCTS AND SPACE / MUSIC / PHOTO / ART / FREAK!

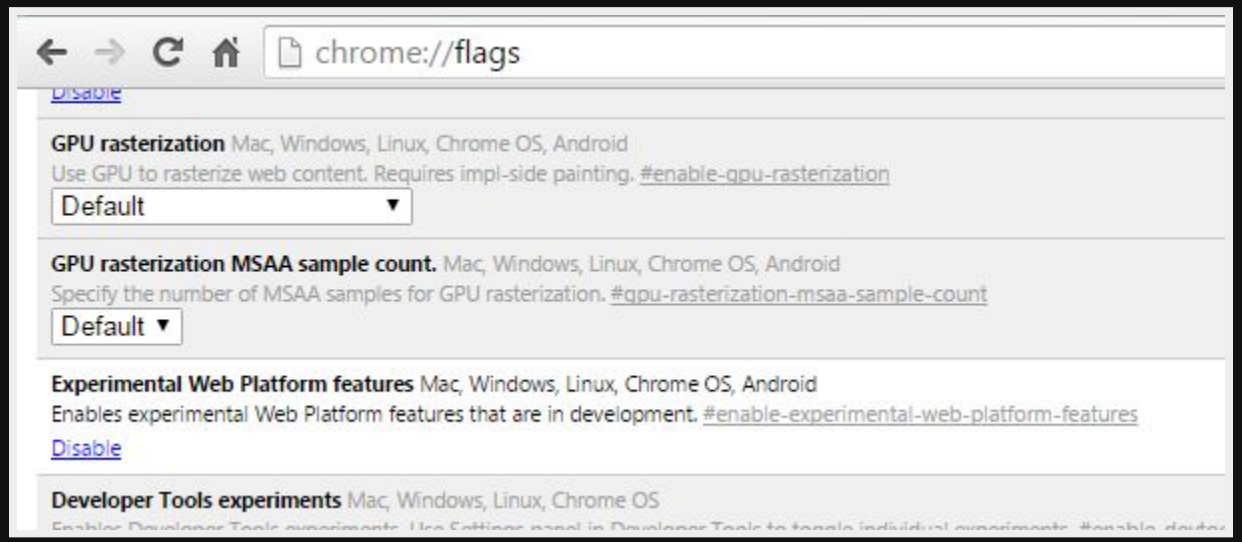
Grid support: not much...for now!



Enabling grid support on the browser: Firefox



Enabling grid support on the browser: Chrome



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!