

GRID

LAYOUT

How to make Responsive Web design...?

The 3 key components:

- Media queries
- Flexible images
- Flexible grids

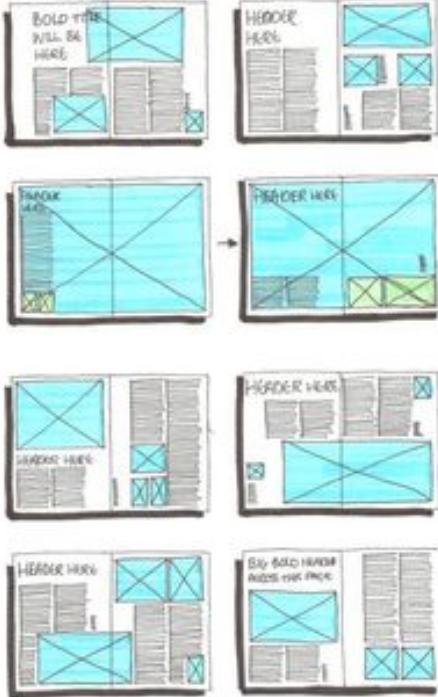
The tools of the trade:

HTML 5

CSS 3

JAVASCRIPT

Grid Layout in print design



DESIGNERY

HOME | ARTICLES | BROWSE | ABOUT | SEARCH | LOG IN

WELCOME TO DESIGNERY.us

Wednesday, December 22, 2010

DESIGNERY.us

The bigger they are... by Mark Weirath, Editor

Logos, the Next Wave by Chon Tatum, Editor

The Bigger They Are... by Mark Weirath, Editor

Will They Pay for It? Publishers Make a Last Stand by Rich Dawson

Web Design Techniques of the Highly Gifted by Mike Dugdale

Holding on to Offline Printing by CSS and the Web-on-the-Street

Licensing in Four Weeks by Dix Handley

More Articles...

Logos, the Next Wave by Chon Tatum, Editor

The Bigger They Are... by Mark Weirath, Editor

Will They Pay for It? Publishers Make a Last Stand by Rich Dawson

Web Design Techniques of the Highly Gifted by Mike Dugdale

Holding on to Offline Printing by CSS and the Web-on-the-Street

Licensing in Four Weeks by Dix Handley

More People...

Logos, the Next Wave by Chon Tatum, Editor

The Bigger They Are... by Mark Weirath, Editor

Will They Pay for It? Publishers Make a Last Stand by Rich Dawson

Web Design Techniques of the Highly Gifted by Mike Dugdale

Holding on to Offline Printing by CSS and the Web-on-the-Street

Licensing in Four Weeks by Dix Handley

More Projects...

SEARCH

Designery is a public forum for design practitioners, students and fans. LEARN MORE

Signed in as Chon Tatum You have 7 new messages Log Out

Ad

EXPLORE PROFILES ON DESIGNERY

- Most Popular
- Most Recently Updated
- Most Discussed

- Most Blocked
- Most Commented
- Random & Interesting

Troubles in Paradise

Chon Tatum Last Updated Dec 18, 2010

Holiday Fun

Gert Morgan Last Updated Nov 8, 2010

Chapter IX

Gert Morgan Last Updated Dec 18, 2010

Calculated Assumptions

James Gaskin Last Updated Nov 8, 2010

Big West ESG

Philip Dommerich Last Updated Dec 18, 2010

Murting Topeka

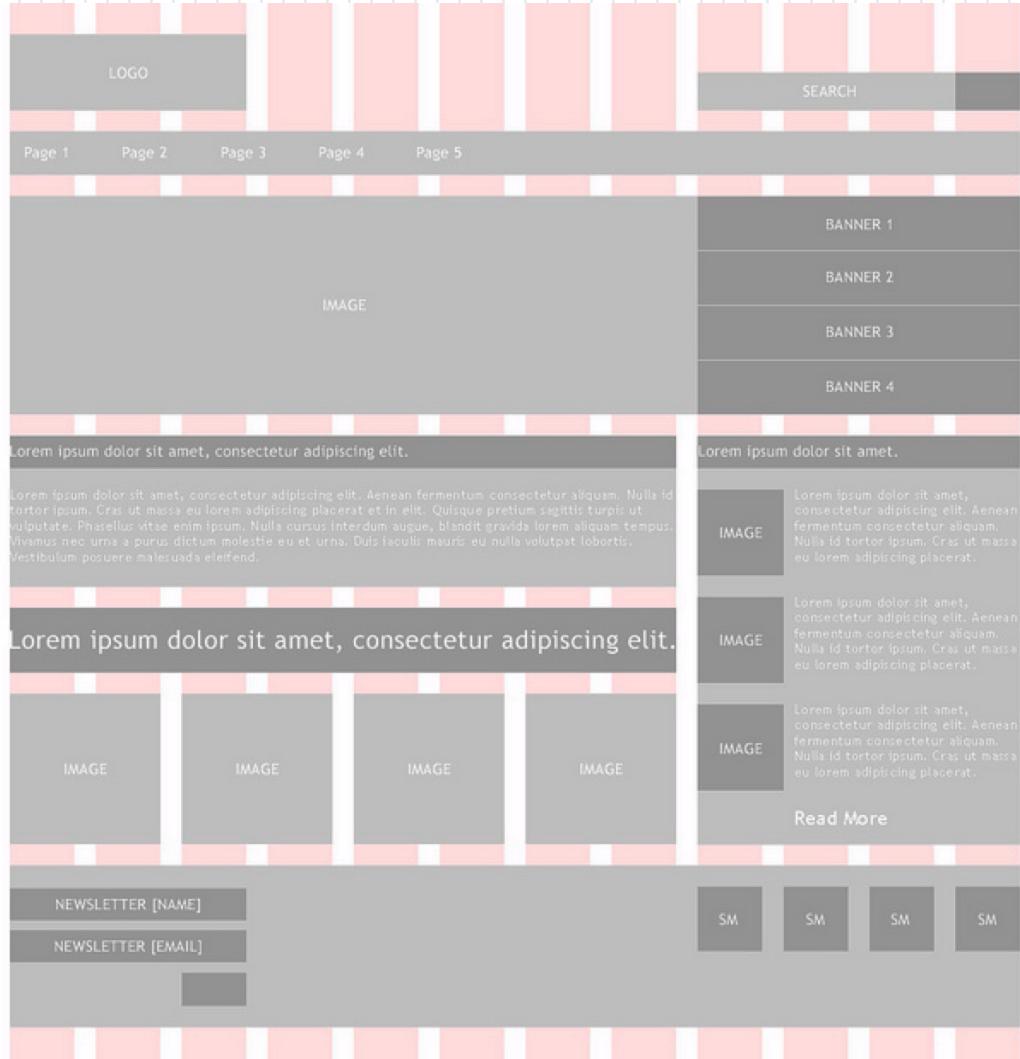
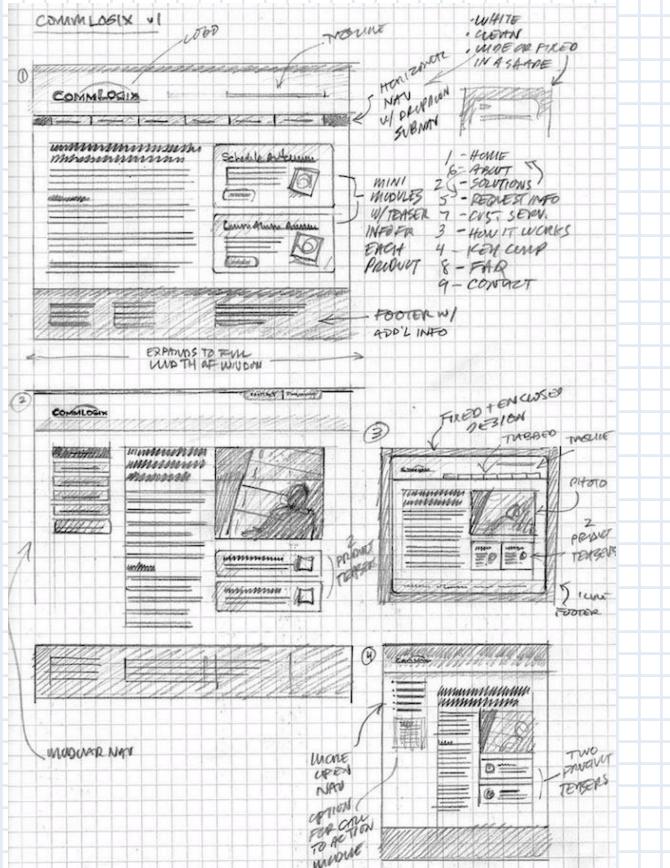
Hipsters

Water hole

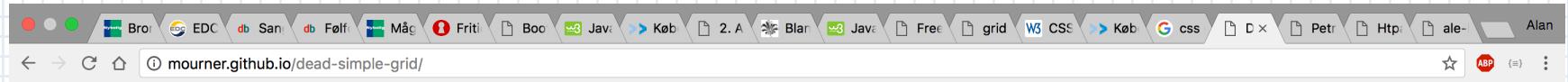
Last Updated Nov 8, 2010

DESIGNERY.us

Grid Layout in web design



Grid Layout in web design



Dead Simple Grid

As Simple As Responsive Grids Get

A demo of *Dead Simple Grid* by [Vladimir Agafonkin](#). View the source and [check it out on GitHub!](#)

Dead Simple Grid is a responsive CSS grid micro framework that is just that. Dead simple. It's the *Malevich's Black Square* of grid frameworks. It is *tiny* (about 250 bytes of CSS) and without dependencies, has only two classes (`row` and `col`), fluid columns with *fixed gutters*, supports *infinite nesting*, and doesn't constrain you to any column sets or media query breakpoints. It embraces concepts of progressive enhancement and mobile first, serving one-column mobile layout to older browsers (IE 6-7). IE 8 is supported if you use [Respond.js](#)

Feature Block One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et m. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Feature Block Two

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et m. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

And One Last Thing

Cum sociis natoque penatibus et m. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et m. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

A Simple Sidebar

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et m. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Info Block One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et m. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor.

Info Block Two

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et m. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor.

Star 529 Fork 69 Tweet

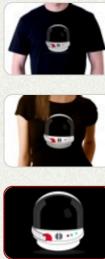
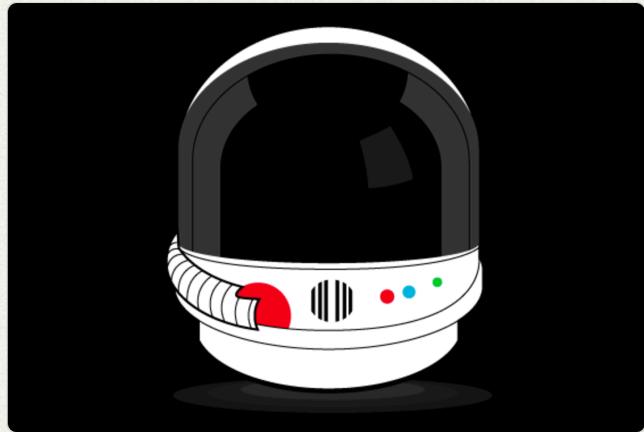
Grid Layout in web design

Chrome File Edit View History Bookmarks People Window Help

Br ED Sa db Føl Må Frit Bo Jav Køl Bla gra Aut W3 CS FX 13 Ix Lu ala Re gric gric Gri gric Alan

bradfrost.com/demo/mobile-first/#

FFLY T-SHIRTS HOODIES PANTS Search Store

Super Ffly T-shirt
\$23 ★★★★☆ 8 Reviews

QTY 1 SIZE S ▾
ADD TO CART

SHARE FIND NEARBY

Behold, your new favorite shirt! This built-to-last tee adapts to your size and capabilities, starting off XS then stretches and scales to keep you comfortable as you grow taller and wider. The future is here, so look sharp.

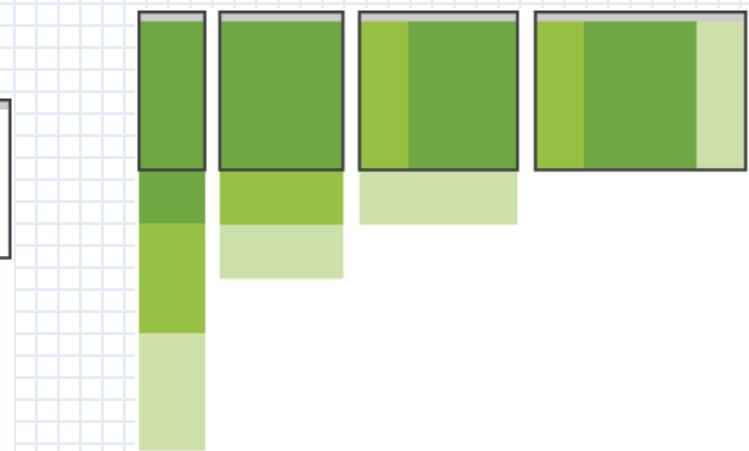
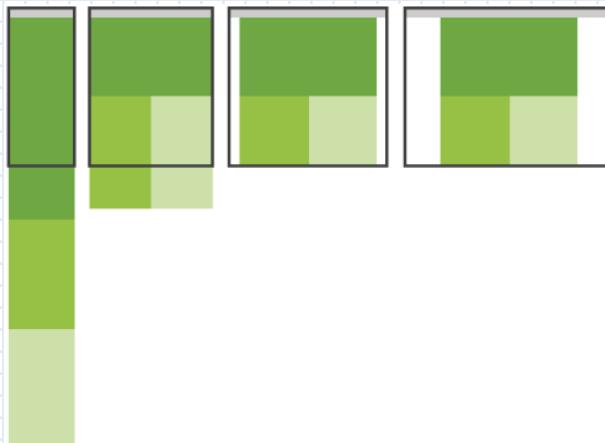
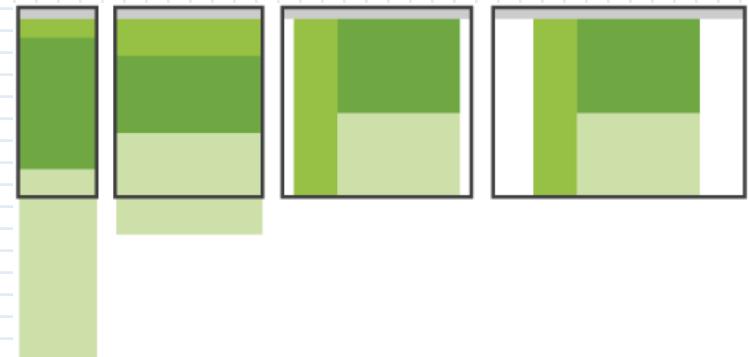
Similar T-shirts

Annotations: OFF

Responsive Web design

The 3 key components:

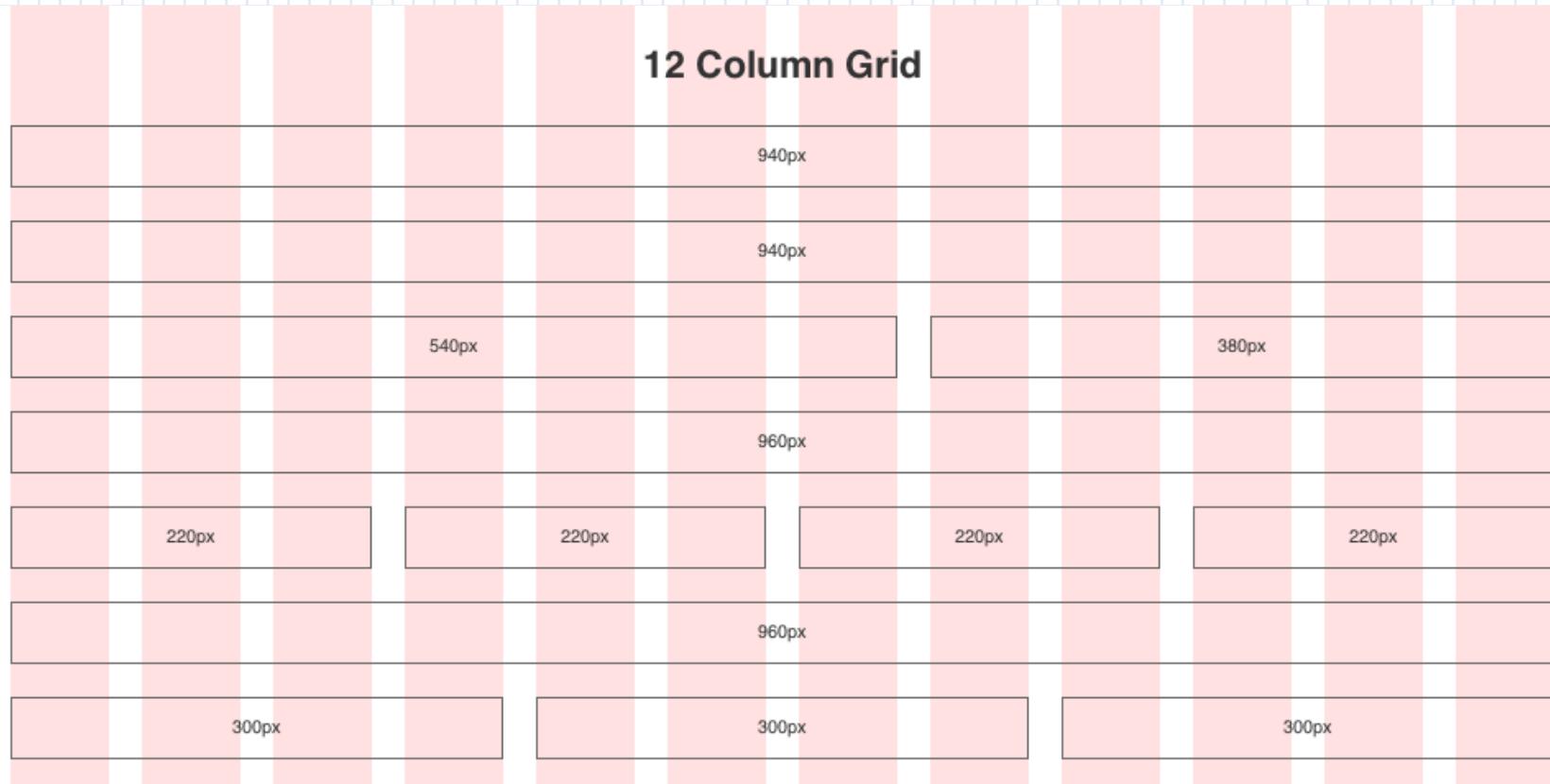
- Flexible grids
- Flexible images
- Media queries



Flexible Grids

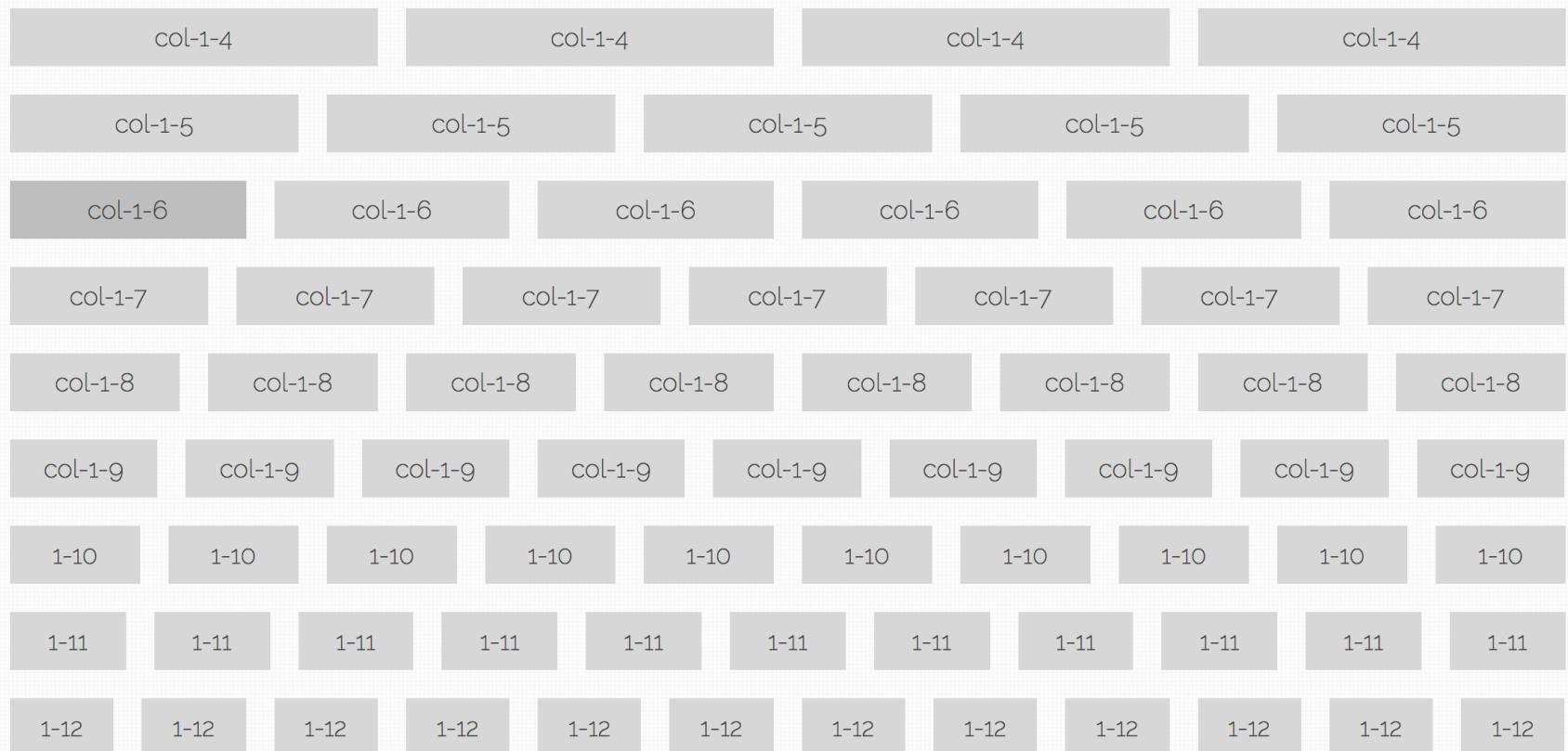


Grid systems



<https://www.webpagefx.com/blog/web-design/responsive-css-grid/>

Grid systems



<https://www.webpagefx.com/blog/web-design/responsive-css-grid/>

Standard grid layouts

The 'Intro Page'

Has a header row with a logo and some nav, some featured content (maybe a carousel or some headline text), a row for the smaller features and a footer.

The logo, spanning 1 of 3

The navigation, spanning 2 of 3

The carousel, spanning 3 of 3

Feature, spanning 1 of 3

Feature, spanning 1 of 3

Feature, spanning 1 of 3

The footer, spanning 3 of 3

The 'Blog'

Has a header row, a row with the content and the sidebar and a footer.

The header, spanning 3 of 3

The content, spanning 2 of 3

The sidebar, spanning 1 of 3

The footer, spanning 3 of 3

Standard grid layouts

The 'Portfolio'

Has a header row with a logo and some nav, a couple of rows of portfolio items and a footer.

The logo, spanning 1 of 3

The navigation, spanning 2 of 3

Portfolio item, spanning 1 of 4

The footer, spanning 3 of 3

The 'Golden Ratio'

According to [wikipedia](#), two quantities are in the golden ratio if the ratio of the sum of the quantities to the larger quantity is equal to the ratio of the larger quantity to the smaller one. It has a value of 1.618. If you use a grid of eight, and split into five columns and three columns the ratio is 1.66, which is pretty close.

This column spans 5 of 8

This column spans 3 of 8

Who's using grid layout...

www.responsivagridsystem.com

Intro Generator Layouts Go Further Playground Download

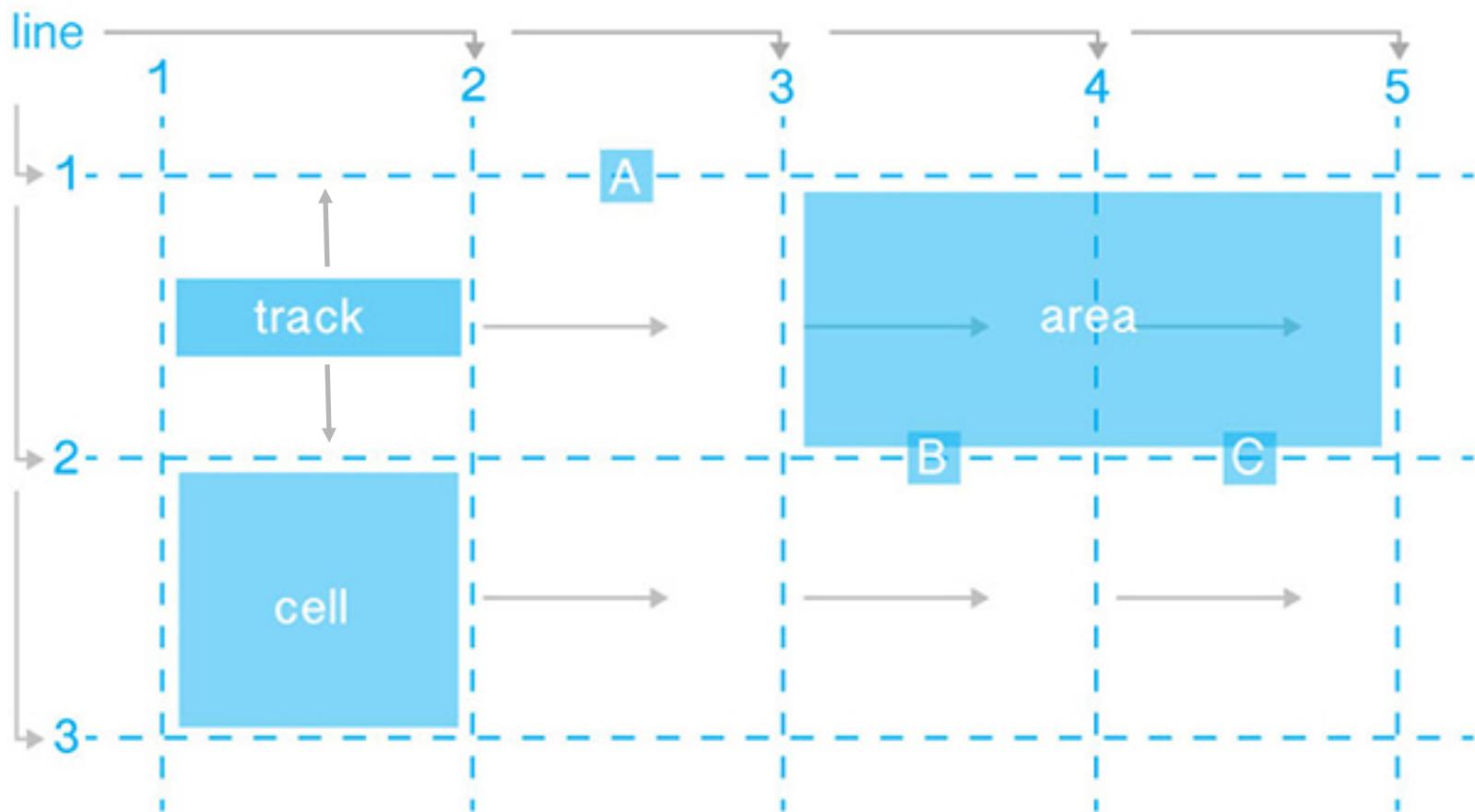
Who's Using It?

Apart from this site? Some of the other websites out there using the responsive grid system:

The screenshot displays a 5x6 grid of website thumbnails, each illustrating a different application of a responsive grid layout. The websites cover a wide range of topics, including personal profiles, news, e-commerce, travel, and industrial services. The layout is organized into five rows and six columns, with each website example enclosed in a light gray box.

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

The CSS Grid



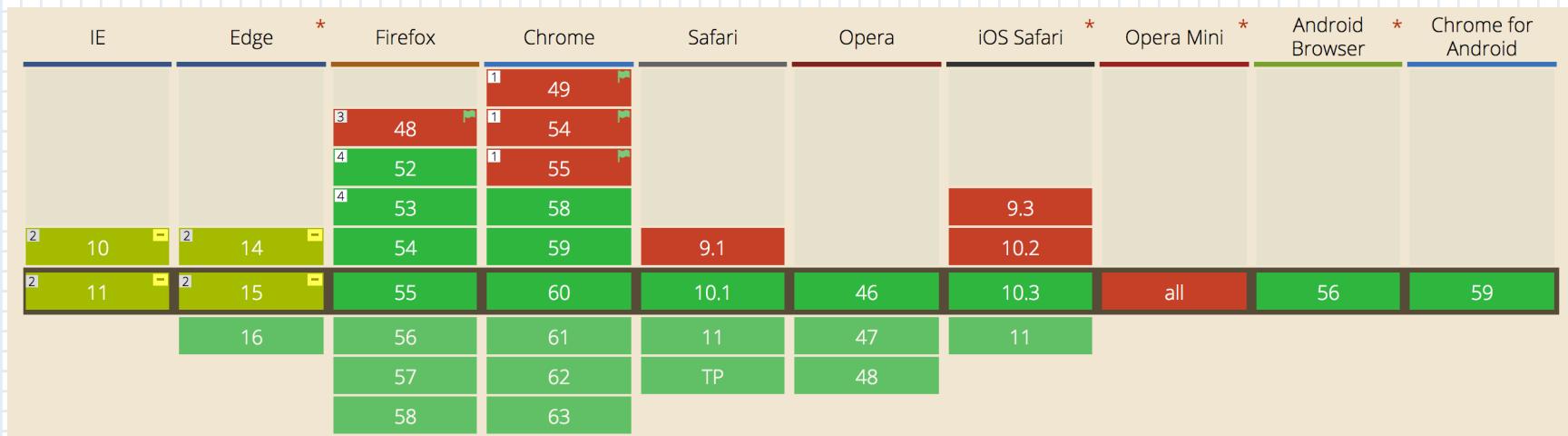
The CSS Grid

- Controls the layout of its content through the use of a grid which **create a sizing and positioning coordinate system for the grid container's contents.**
- Grid Layout features:
 - **fixed, flexible, and content-based track sizing functions**
 - explicit item placement via forwards (positive) and backwards (negative) **numerical grid coordinates, named grid lines, and named grid areas**
 - automatic item placement into empty slots, **including reordering with order**
 - space-sensitive track repetition and **automatic addition of rows or columns to accommodate additional content**
 - **control over alignment and spacing** with margins, gutters, and the alignment properties
 - the ability to overlap content and control layering with z-index
- Grid containers **can be nested** (or mixed with flex containers) as necessary to create more complex layouts.

<https://www.w3.org/TR/css-grid-1/>

In other words...

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



<https://caniuse.com/#search=grid>

The CSS Grid terminology

Grid lines

- are the horizontal and vertical dividing lines of the grid. A grid line exists on either side of a column or row.

A grid cell

- is the intersection of a grid row and a grid column.
- the smallest unit of the grid that can be referenced when positioning grid items.

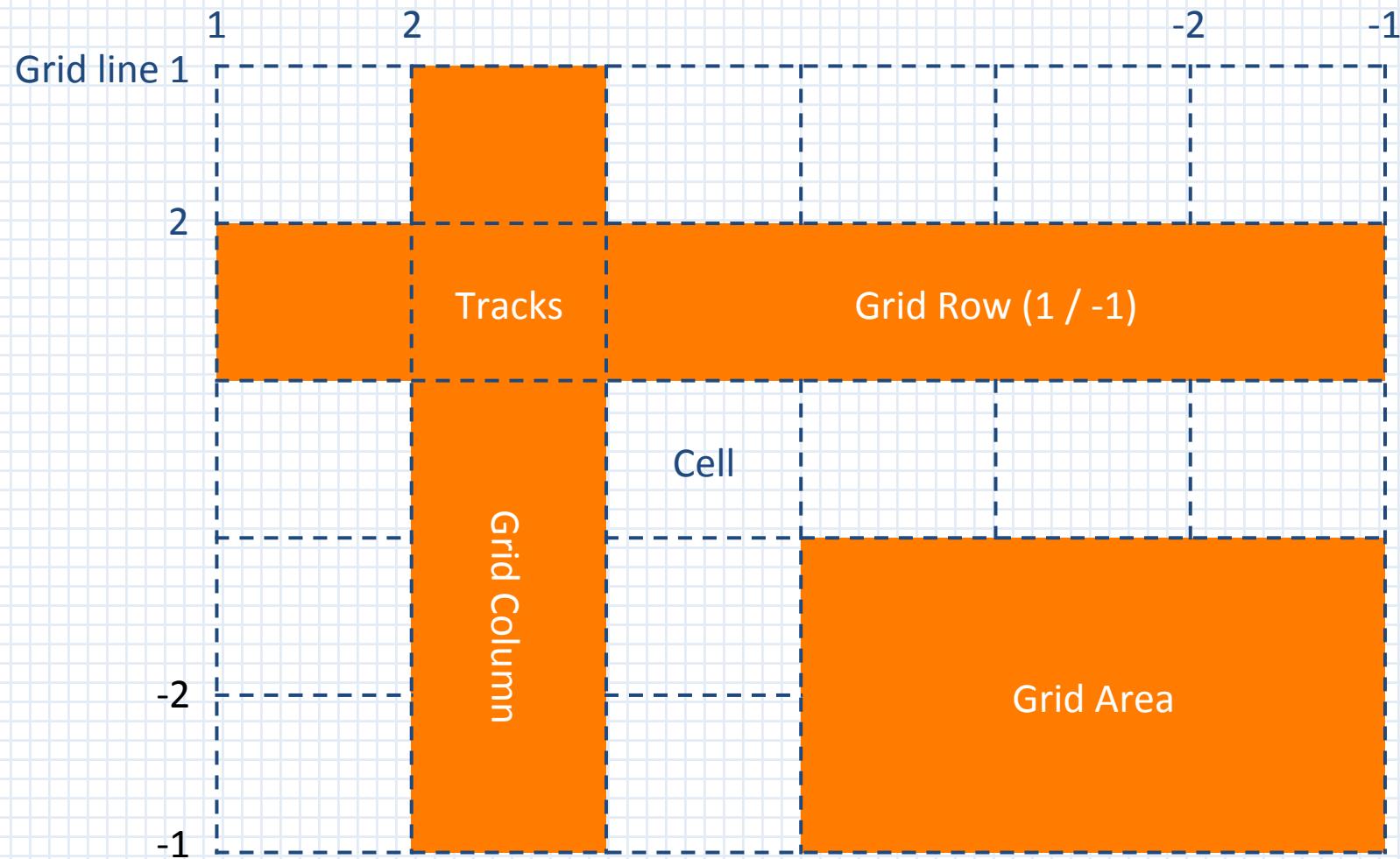
Grid track

- a grid column or grid row
- it is the space between two adjacent grid lines.
- Each grid track is assigned a sizing function, which controls how wide or tall the column or row may grow

A grid area

- A grid area consists of one or more adjacent grid cells.
- It is bound by four grid lines, one on each side of the grid area
- can be named explicitly using the grid-template-areas property

The CSS Grid terminology



The Explicit Grid

- The **three properties**:
 - grid-template-rows,
 - grid-template-columns, and
 - grid-template-areastogether **define the explicit grid** of a **grid container**.
- The final grid **may** end up larger due to grid items placed outside the explicit grid; in this case **implicit tracks will be created** (see later slide).
- Numeric indexes in the grid-placement properties count from the **edges** of the explicit grid.
- Positive indexes count from the start side (starting from 1 for the start-most explicit **line**), while negative indexes count from the end side (starting from -1 for the end-most explicit line).

<https://www.w3.org/TR/css-grid-1/>

CSS Grid-template: Two approaches

```
@media screen and (min-width: 600px){  
    main{  
        margin: auto;  
        display: grid;  
        grid-gap: 1px;  
        grid-template-columns: 2fr 4fr 2fr;  
        grid-template-areas:  
            "header header header"  
            "nav nav nav"  
            "article article aside"  
            "footer footer footer"  
    }  
    header{  
        grid-area: header;  
    }  
    nav{  
        grid-area: nav;  
    }  
    article{  
        grid-area: article;  
    }  
    aside{  
        grid-area: aside;  
    }  
    footer{  
        grid-area: footer;  
    }  
}
```

```
@media (min-width: 600px){  
    main{  
        grid-template-columns: 4fr 1fr;  
    }  
    header, nav, footer{  
        grid-column: 1/-1;  
    }  
}
```

Site Header

Home About Products Contact

Main content header

Section header

Body text Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce quis lectus quis sem lacinia nonummy. Proin mollis lorem non dolor. In hac habitasse platea dictumst. Nulla ultrices odio. Donec augue. Phasellus dui. Maecenas facilisis nisl vitae nibh. Proin vel seo est vitae eros pretium dignissim. Aliquam aliquam sodales orci. Suspendisse potenti. Nunc adipiscing euismod arcu. Quisque facilisis mattis lacus. Fusce bibendum, velit in venenatis viverra, tellus ligula dignissim felis, quis euismod mauris tellus ut urna. Proin scelerisque. Nulla in mi. Integer ac leo.

Aside Nunc urna ligula, gravida a, pretium vitae, bibendum nec, ante. Aliquam ullamcorper iaculis lectus. Sed vel dui. Etiam lacinia risus vitae lacus. Aliquam elementum imperdiet turpis. In id metus. Mauris eu nisl. Nam pharetra nisi nec enim.

Page footer

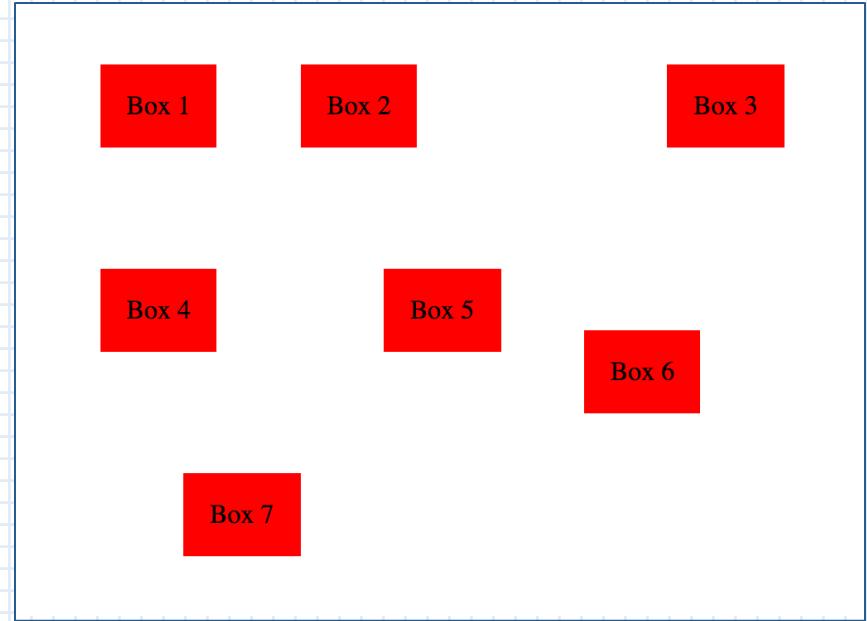
The Implicit Grid

- When grid items are positioned **outside the explicit grid**, the grid container generates **implicit grid tracks** by adding implicit grid **lines** to the grid.
- These lines **together with the explicit grid** form the implicit grid.
- **The grid-auto-rows and grid-auto-columns properties size these implicit grid tracks.**
- The grid-auto-flow property controls auto-placement of grid items without an explicit position.
- Once the explicit grid is filled (or if there is no explicit grid) **auto-placement will also cause the generation of implicit grid tracks.**

<https://www.w3.org/TR/css-grid-1/>

Alignment and Spacing

- Gutters:
 - grid-column-gap
 - grid-row-gap
 - grid-gap
- Aligning with auto margins
- Row-axis Alignment:
 - justify-self
 - justify-items
- Column-axis Alignment:
 - align-self
 - align-items
- Aligning the Grid
 - justify-content
 - align-content
- Grid Container Baselines



<https://www.w3.org/TR/css-grid-1/>

CSS Grid Links

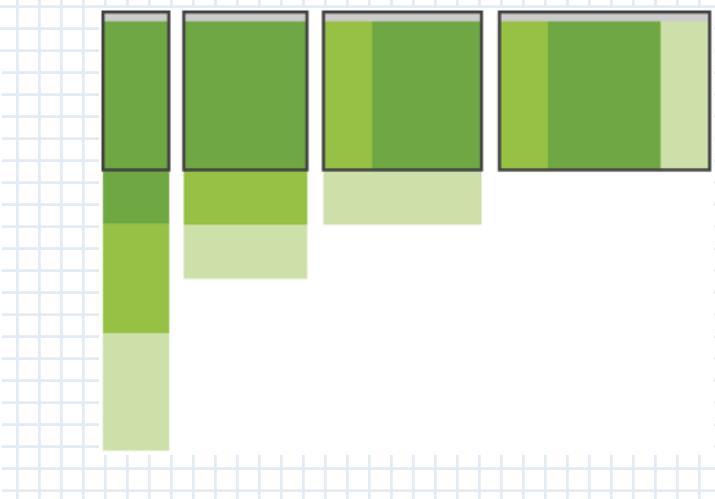
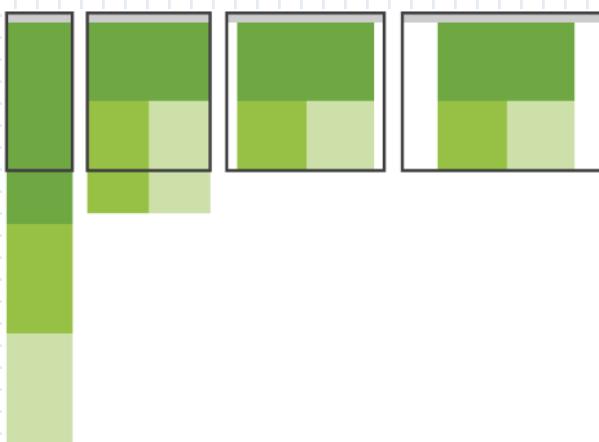
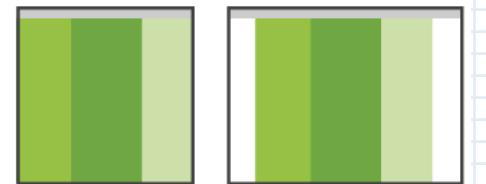
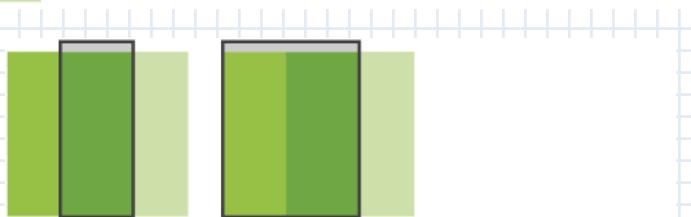
- <https://gridbyexample.com/video/>
- <https://www.w3.org/TR/css-grid-1/#implicit-named-areas>

For inspiration (examples of older approaches):

- <https://www.responsivegridsystem.com/layouts/>
- <https://www.webpagefx.com/blog/web-design/responsive-css-grid/>

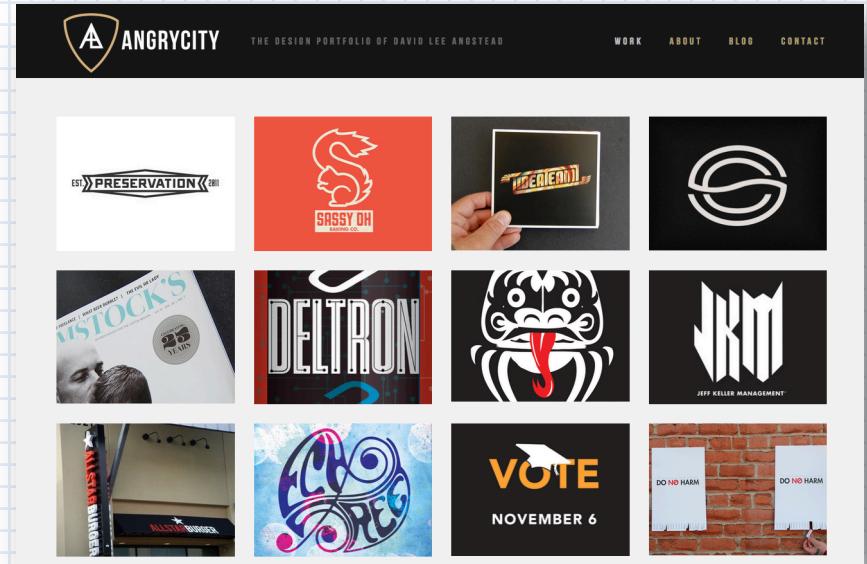
Exercise 1

- Replicate all of Luke Wroblewski's RWD layout patterns using CSS Grid.
- Try both approaches: grid-template-area and grid-column span.
- <https://www.lukew.com/ff/entry.asp?1514>



Exercise 2

- Replicate the 4 “standard” grid layouts (slide 10 + 11):
 - Intro page
 - Blog
 - Portfolio
 - The Golden Ratio
- Use CSS grid
- Try both approaches: grid-template-area and grid-column span



Exercise 3 (next week)

- Deconstruct and rebuild FFLY with CSS Grid
- [http://bradfrost.com/
demo/mobile-first/#](http://bradfrost.com/demo/mobile-first/#)
- The page should be fully responsive.
- Focus on the grid structure - no need for fancy JS animations...

