

# Design Burst

Week 3:

A taste of CSS grids



Mondrian cake. Source: [Smithsonian.com](https://www.smithsonian.com). Photo by Clay MacLachlan

# What is a grid?

- The grid is the most widely used system for arranging objects in patterns that make sense, whether by **semantic**, **practical** or **aesthetic** criteria.
- Grids are used in very diverse types of design.

For example ...

# Grids are used in ...

- print and graphic design



Meal voucher [designed](#) for the Munich Olympics (1972)

# Grids are used in ...

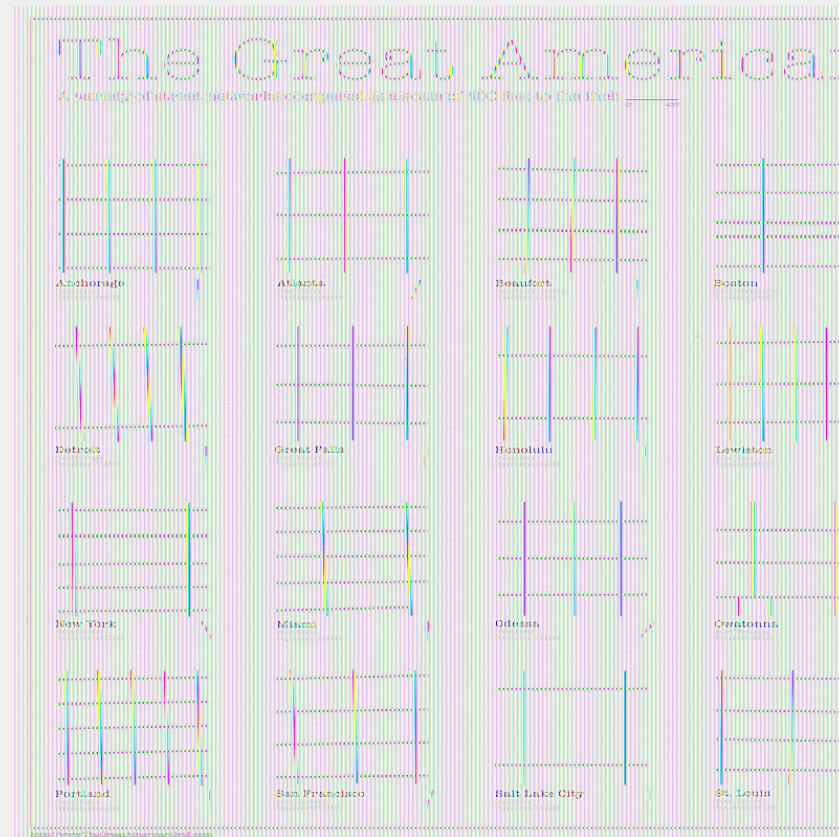
- furniture design
- interior design



Grids galore (furniture design: [de zeen](#))

# Grids are used in ...

- architecture
- urban planning



# What is the CSS grid?

In design, a grid is an often implicit framework for structuring the page into blocks defined by intersecting rows and columns.

In web design, you can use a grid to structure your entire webpage or specific container-elements on it. Here's an example of how you define a grid:

```
div {                                /*The commonest
    display: grid;                    specification*/
}
```

or

```
.wrapper {
    display: inline-grid;
}
```

# What's a grid good for?

Think of the grid as a 'creative constraint'

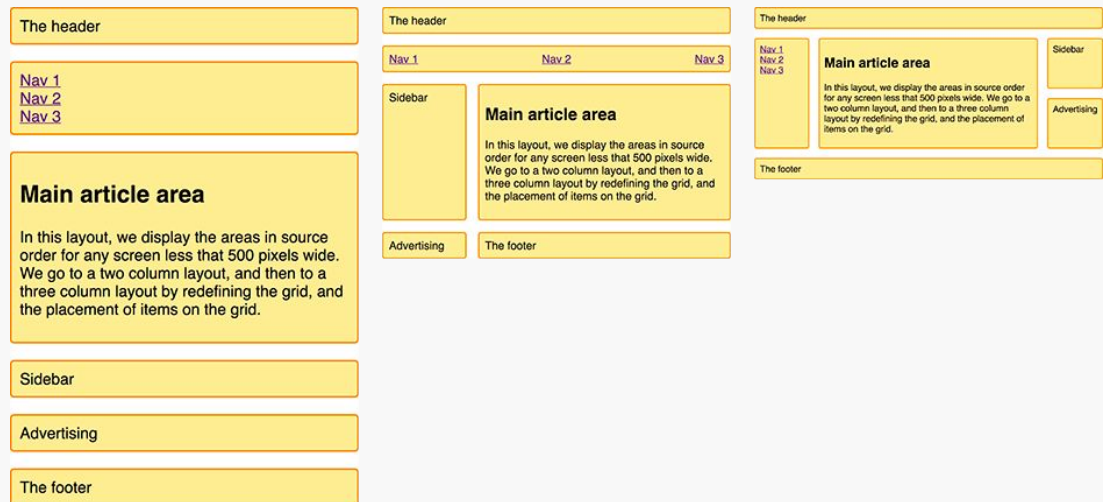
In **print and in web design**, grids

- allow us to position the elements of a page harmoniously and consistently regardless of page size or screen size
- help us structure the content of a page meaningfully
- make it easier for users to scan and navigate the content of the page.

# The advantages of the CSS grid

In **web design**, the CSS grid has

- made it easier to display content consistently across most browsers (apart from lesser browsers and the usual suspect - IE).
- removed the need for layout hacks - notably, tables
- revolutionised responsive design
- opened up possibilities for creative layouts.



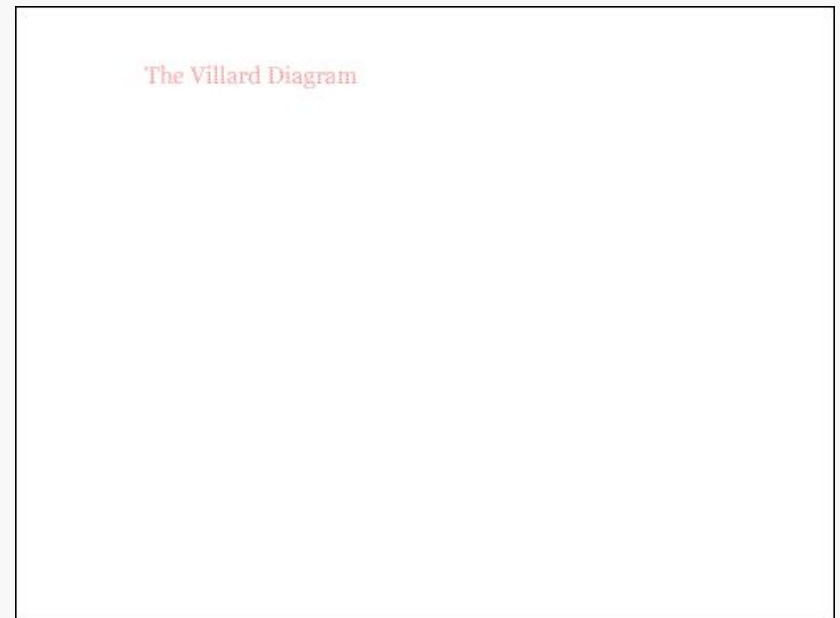
One grid, three layouts. Image credit: [MDN](#)



# A potted history of grids

## 1. Before print

- In the Renaissance, many artists and architects used various **ratios** – some studied in antiquity – to create frameworks out of intersecting lines. These frameworks helped them structure the layout of paintings and buildings.
- The proportions 13th-century French artist/architect **Villard de Honnecourt** applied in his diagrams influenced early printers and are still used today by designers.

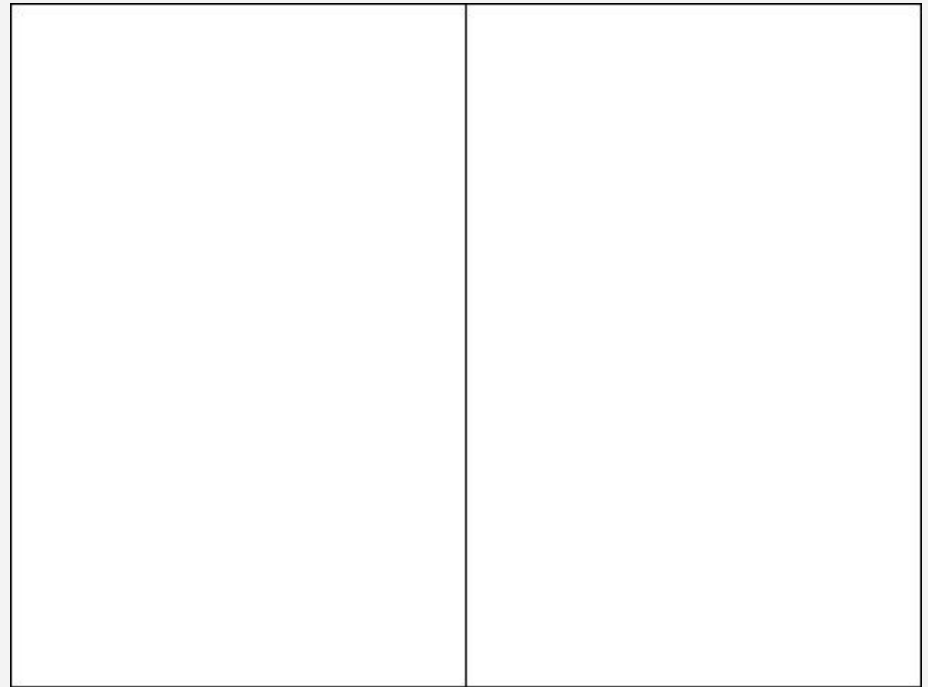


Villard's Diagram. Image credit: [Retinart](#)

# A potted history of grids:

## 2. Gutenberg and print

- Calligraphers, scribes and, from the 15th century onwards, printers, all applied canons based on such ratios – mainly (variants of) the **golden ratio** (1.618) or **Rosarivo's** 'golden number' (2:3).



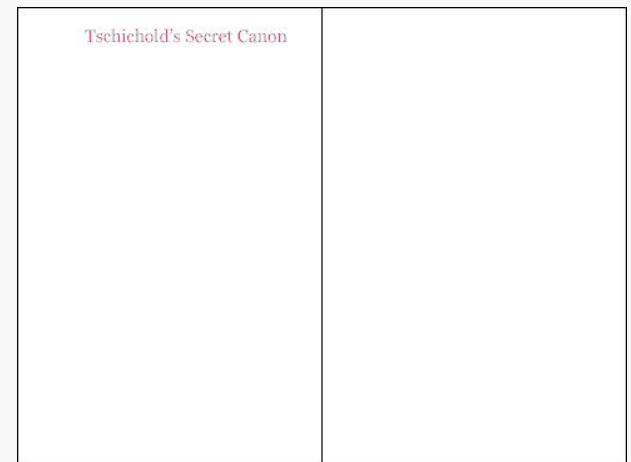
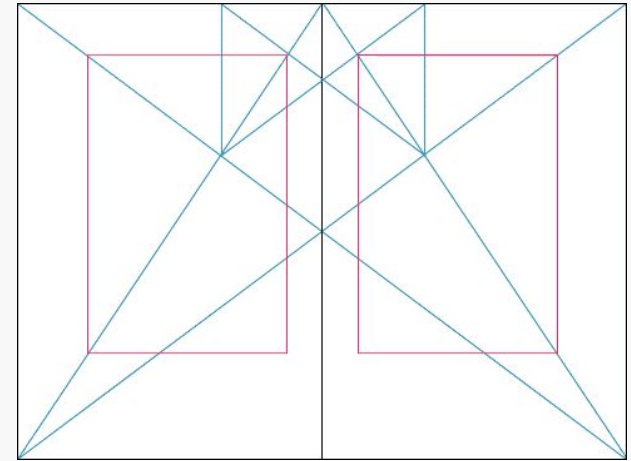
Rosarivo's Canon. Image credit: [Retinart](#)

# A potted history of grids:

## 3. The modern grid

In the 1940s, Dutch book designer **J. A. Van de Graaf** derived from medieval canons his own canon for page layout, which is still used today.

At around the same time, German designer and typographer **Jan Tschichold** distilled his enduring 'secret canon' from Van de Graaf's rules and from the layouts applied in illuminated manuscripts and early books.



Top: Van de Graaf's canon; Bottom: Tschichold's canon  
Image credits: [Retinart](#)

# A potted history of grids:

## 4. The CSS grid

1940s

Rosarivo's and Tschichold's layout rules evolved into the grids that graphic designers and printers came to use.

1990s

In the late 1990s, these grids inspired the creators of CSS, Dr. Bert Bos and Håkon Wium Lie, to develop for the web the framework that became the CSS grid.

2017

All major browsers shipped the first stable version of the CSS grid within a year of its launch in March 2017.

The grid [now has](#) over 90% unprefixed support across different devices.

# Anatomy of the CSS grid

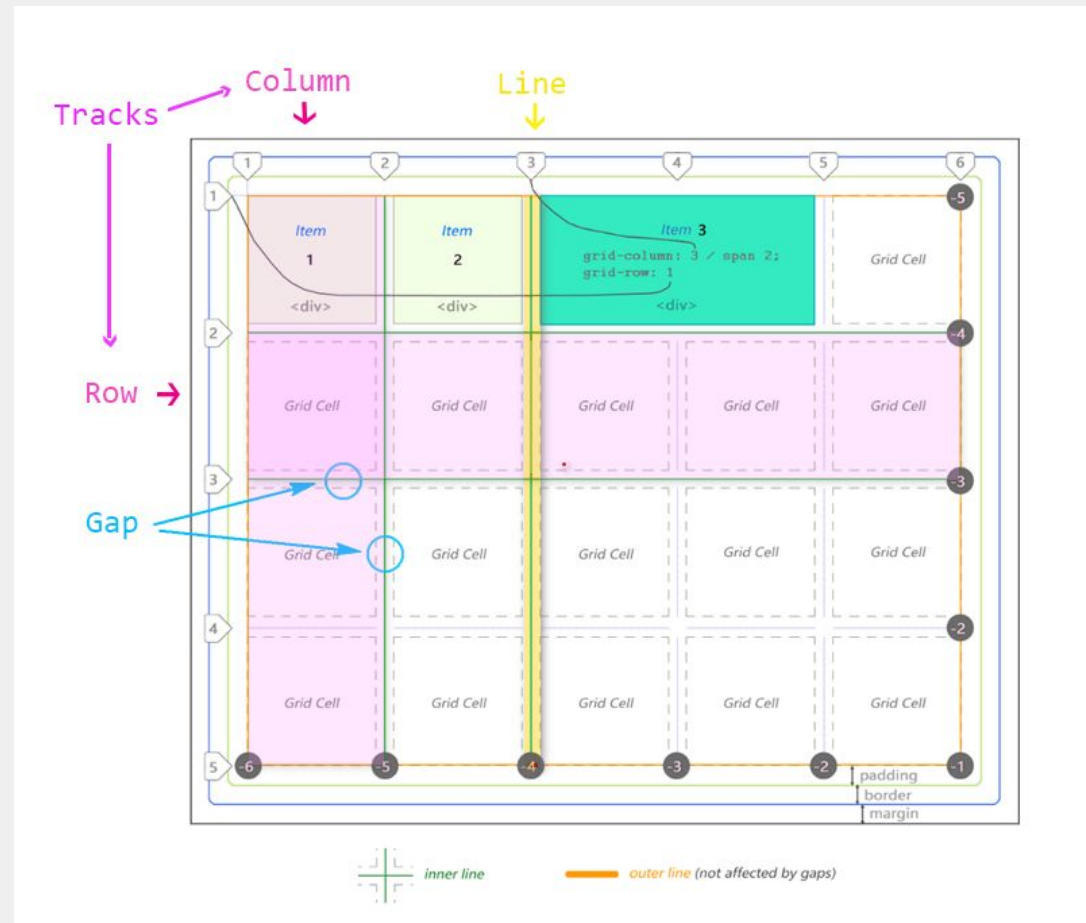
**Track:** the rectangular area defined by grid lines

**Column:** a vertical track

**Row:** a horizontal track

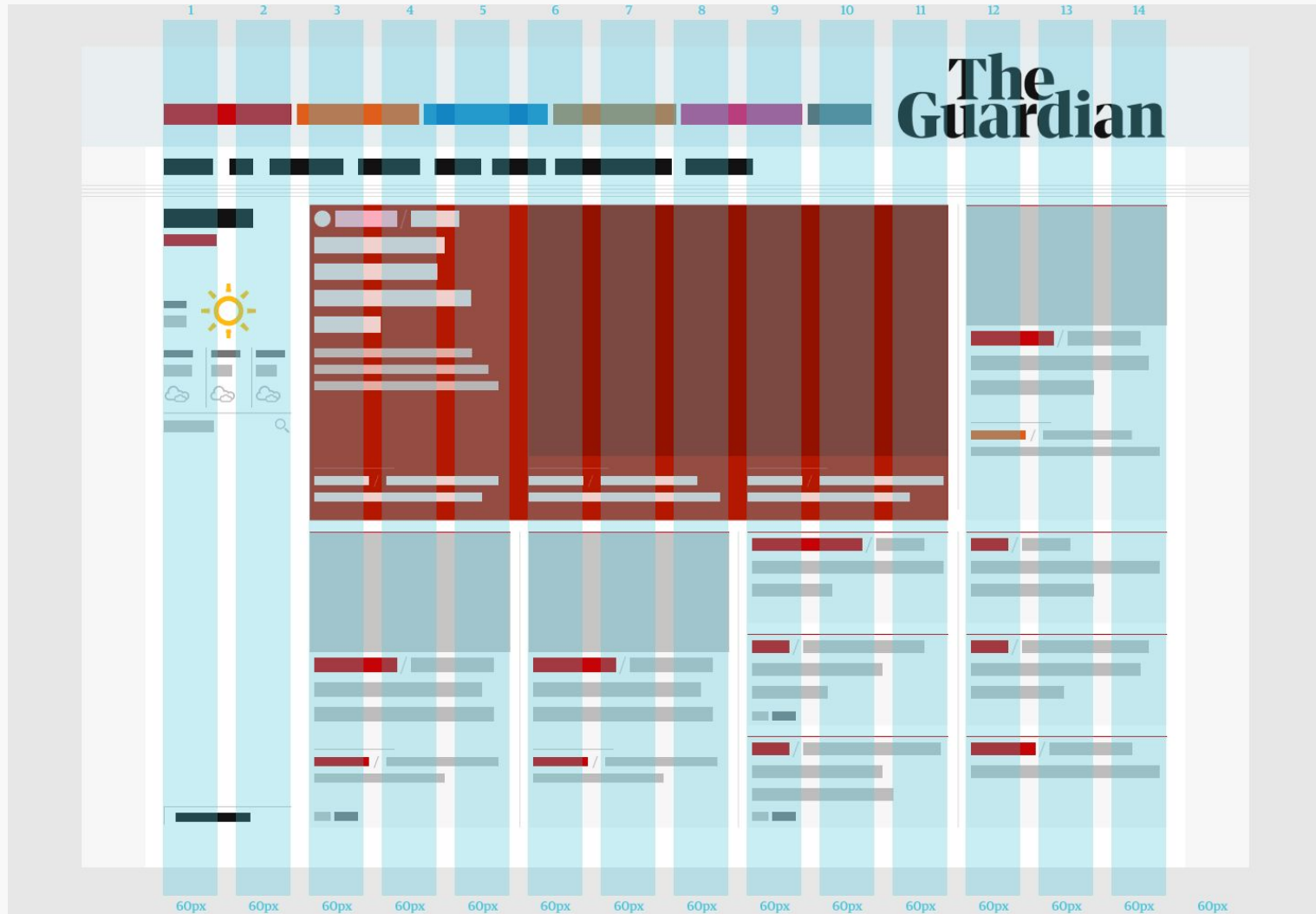
**Gap/Gutter:** the space between tracks

**NB:** the numbers you see correspond to lines, *not* tracks.



Based on [this image](#) from freeCodeCamp

# Example of responsive grid



# Types of CSS grid layouts

1. **The single-column grid:** popular in print. When might a web designer want to use a single-column grid?
2. **The multi-column grid:** it divides the page vertically into multiple columns. The versatile 12-column grid allows you to divide the page into 2, 3, 4, 6 or 12 vertical tracks.
3. **The modular grid:** very popular and [here's why](#):  
'While a multicolumn grid splits a page vertically into a number of columns, a modular grid subdivides a page both vertically and horizontally into [...] a matrix of cells, or modules.'

# The original single-column grid

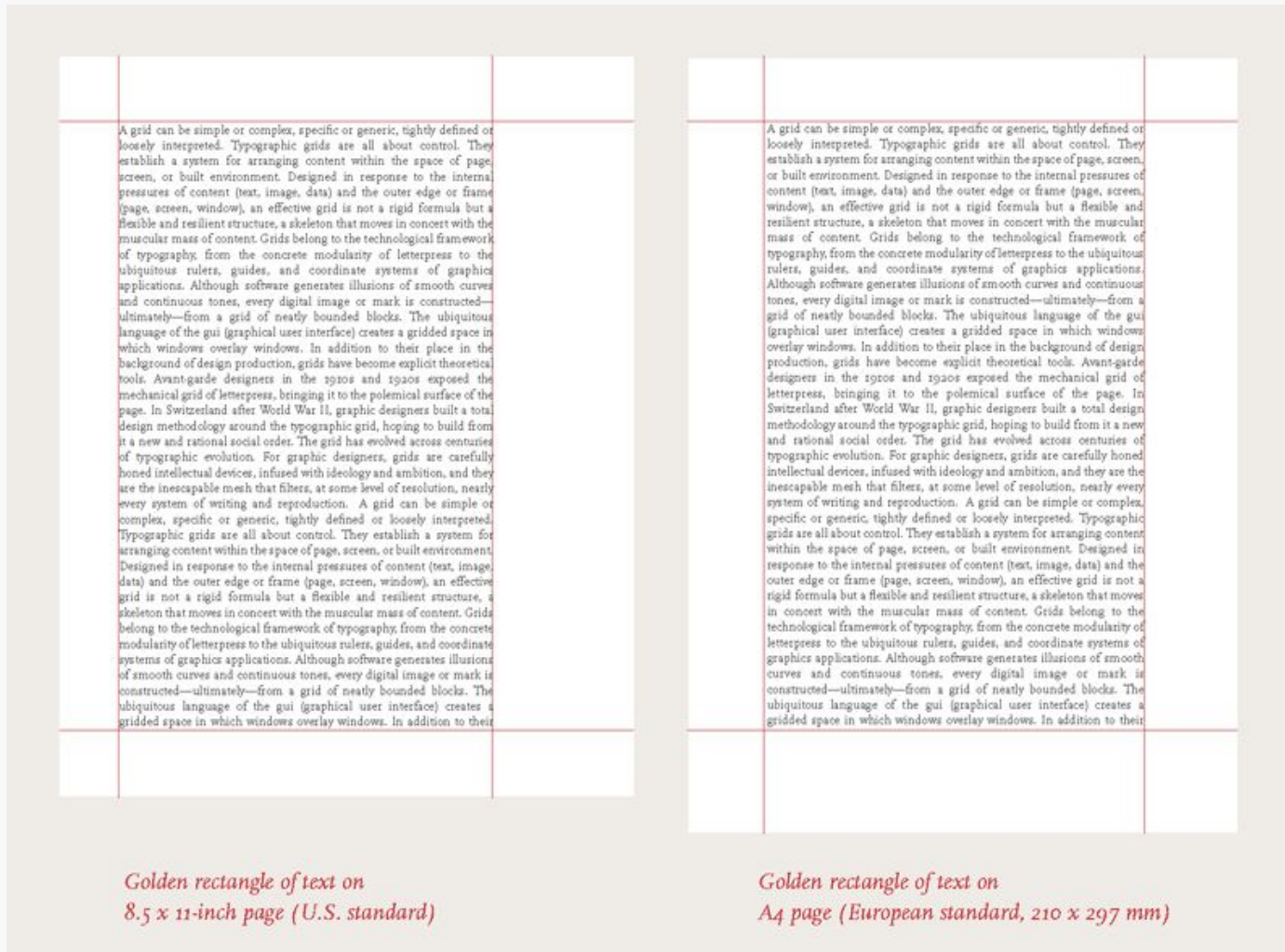


Image source: [Thinking with Type](#)



# The single column grid in web design

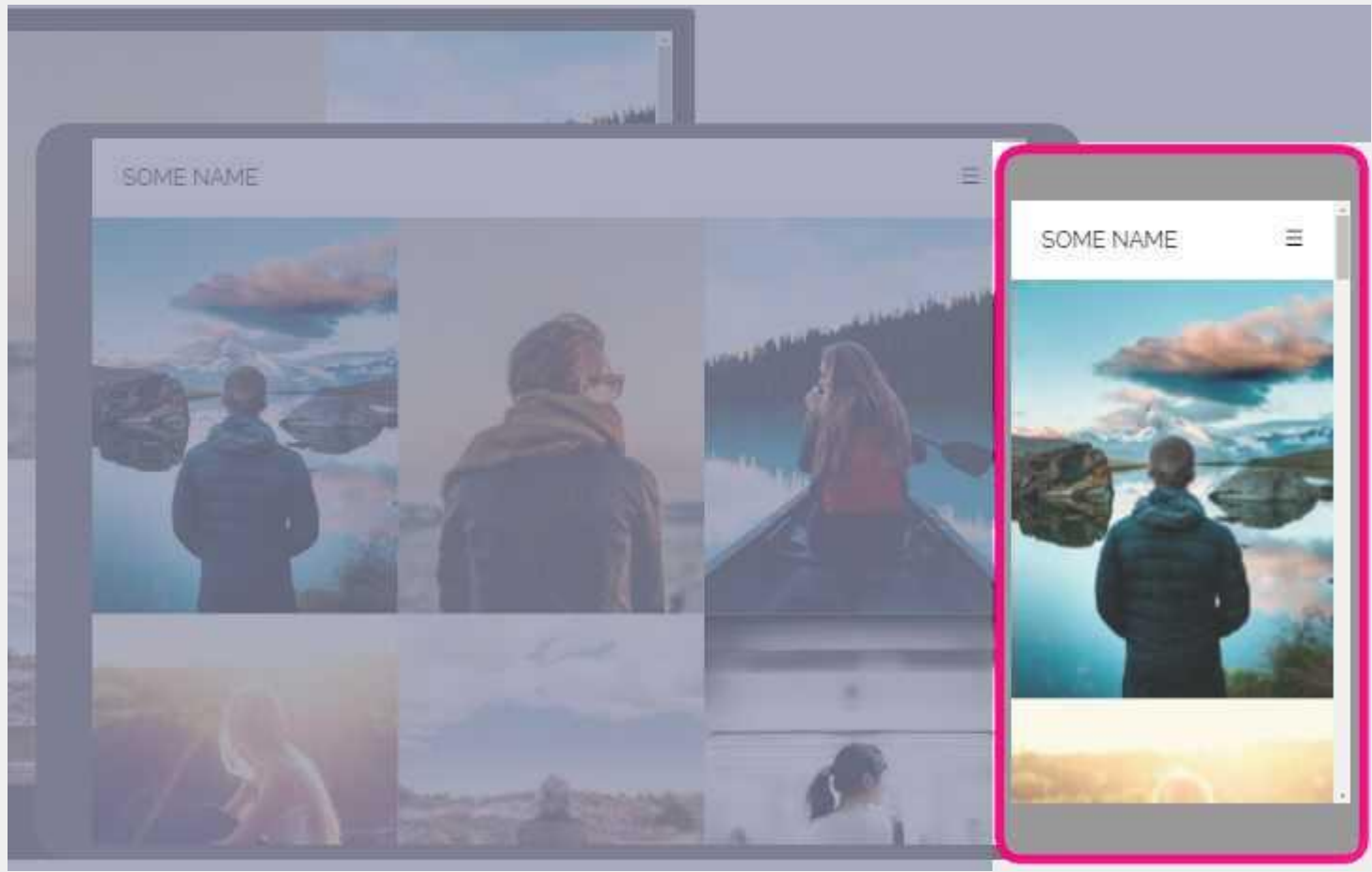
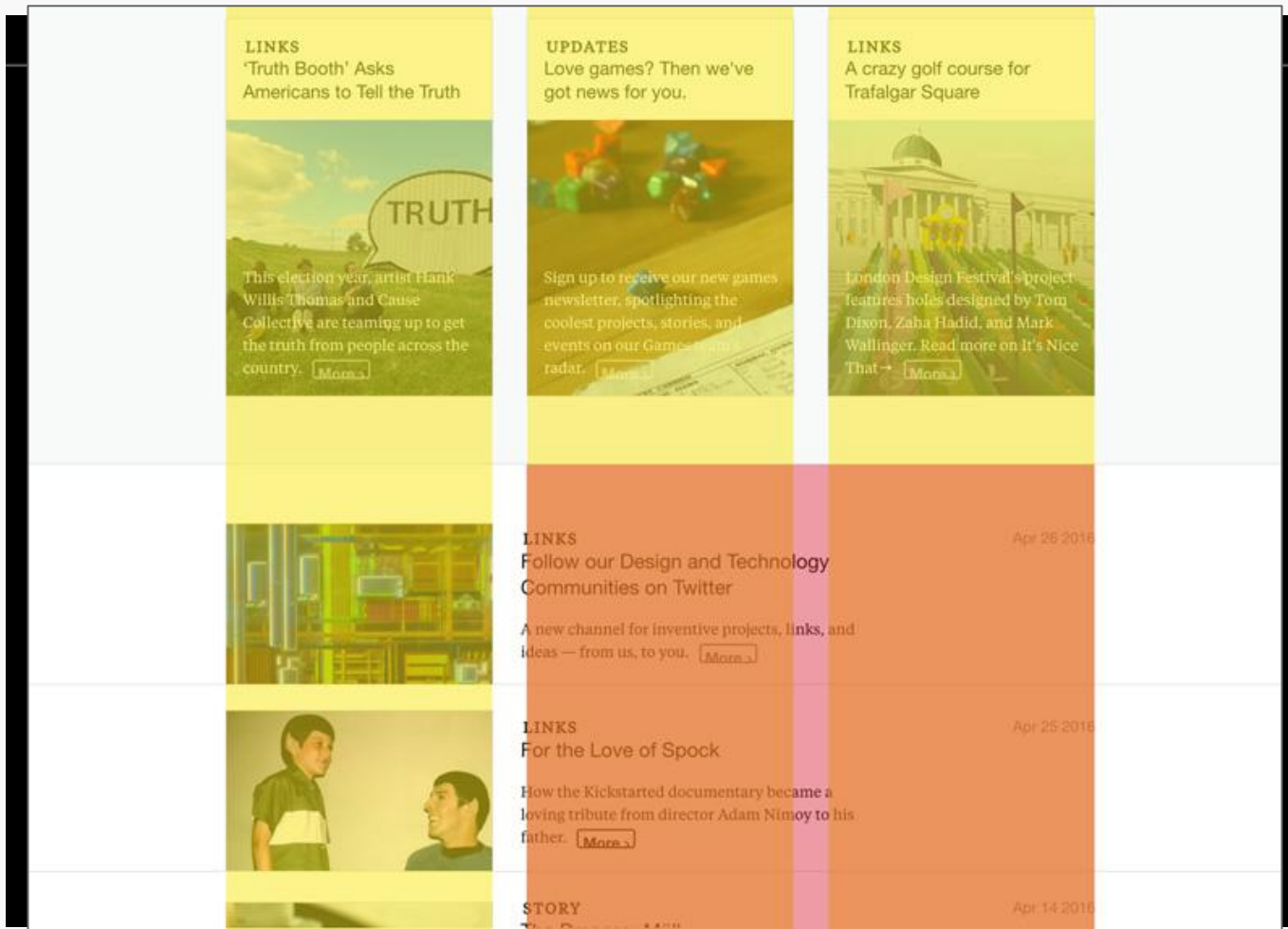


Image source: [w3schools](https://www.w3schools.com)

# Examples of the multi-column grid



3-column grid

Image source:

[Tutsplus](#)

4-column grid

Image source:

[Tutsplus](#)

12-column grid

Image source:

[DePaul University](#)

# Examples of the modular grid

Residence Mixte

NOUS

MENU

RESIDENCE MIXTE

The Book Author Press Events Shop Contact Podcasts

YOUR LIFE

Andrew Simpson

← →

©

Synopsis

Applying design principles to your life.

Author

Vince Frost\*

Shop

E017 Frankie Ratford, founder of The Design Kids with Vince Frost

View the latest events

prescrire

La Redoute

Trois Rôles Vins

laCoop

Cartridge World

SNCF

Vous souhaitez discuter de votre projet ?

Nom \*

Email \*

Votre message \*

Sources:  
[Design Your life](#)

[Résidence Mixte - Agence de communication](#)

[Quincy Réquin & Associés](#)

# Tips for using grids

- You can – and will often want to – use a grid with other CSS positioning methods, such as **flexbox**.
- You can **nest** grids.
- You can use the z-index to **stack** elements in a grid - without specifying absolute/relative positioning.
- You can easily make an element **span the entire grid**:

```
grid-column: 1/-1.
```

- You can space columns differently from rows (but note that all columns will be spaced identically and all rows will be spaced identically):

```
grid-gap: 20px 40px; /*row-gap, column gap*/
```

# Current limitations

- The grid is not inherited: the grid syntax will position accordingly only the immediate children of an element to which you've applied a grid.
- The '[subgrid](#)', i.e. a nested grid that inherits the parent-grid's specifications, is still an experimental feature (but you can nest independent grids).
- You cannot specify differently sized gaps/gutters between different pairs of columns or of rows.

# The Challenge

Use a grid in your project!

## Suggestion

Use an **8-point grid**, i.e. multiples of 8, to define the grid's dimensions, padding and margin. The 8-pt grid is not the only system around, but it's a good system.

## Why?

[Because](#) 'The majority of popular screen sizes are divisible by 8 which makes for an easy fit.'

## Tip

Start with the guide to the 8-pt grid in the **Resources** provided here.

# The Challenge

## Desktop

Vertical Grid Overlay  ⌘K

Horizontal Lines Overlay  ⌘H

Settings Report Advanced

**Columns**

Large  Small

**Gutters**

Vertical - Large  
Inner  Outer

Vertical - Small  
Inner  Outer

Horizontal  
Height

**Screen Sizes**

Max Layout Width

Large to Small Breakpoint

**Offset**

X  Y

Reset Grid Settings

## Mobile

Vertical Grid Overlay  ⌘K

Horizontal Lines Overlay  ⌘H

Settings Report Advanced

**Columns**

Large  Small

**Gutters**

Vertical - Large  
Inner  Outer

Vertical - Small  
Inner  Outer

Horizontal  
Height

**Screen Sizes**

Max Layout Width

Large to Small Breakpoint

**Offset**

X  Y

Reset Grid Settings

Grid  
8-pt  
basis

Mobile  
4 columns  
or  
1 column

Desktop  
8 columns

# Resources

## History of the grid

[99designs: History of the Design Grid](#)

## A comprehensive introduction to the CSS grid

Excellent guide on grid units, positioning, sizing and more

[freeCodeCamp: CSS grid - The Beginner's Guide](#)

## CSS grid cheat-sheet

[Download the PDF from GitHub](#)

## Grid examples

Great collection of resources by CSS grid pioneer Rachel Andrew

[Grid by Example](#)

## Guide to the 8-point grid

Start here before experimenting with the grid in your project

[Spec: The 8-point Grid](#)

## The modular grid

[Modular Grid: Complete Beginner's Guide](#)



