СSS grid layout

Hello everyone! My name is Vitali and now I will tell you about CSS Grid Layout. This presentation is a simple demonstration of capability the CSS Grid. This is not a guide. This is just demonstration

Now I`ll show you my presentation plan.

so in the beginning I will give you some useful links, then I will show which browsers support Grid,  
after that, I’ll tell you a little about GRID and in the end we will create a simple web site template

1. Useful links

2. Browser support

3. What is a Grid?

4. Create simple layout

Useful Links

The first link is of course the specification.

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

The second and third links to very interesting and useful articles on css-tricks and MDN.

1. CSS tricks(<https://css-tricks.com/snippets/css/complete-guide-grid/#prop-grid-template-areas>)
2. MDN (<https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout>)

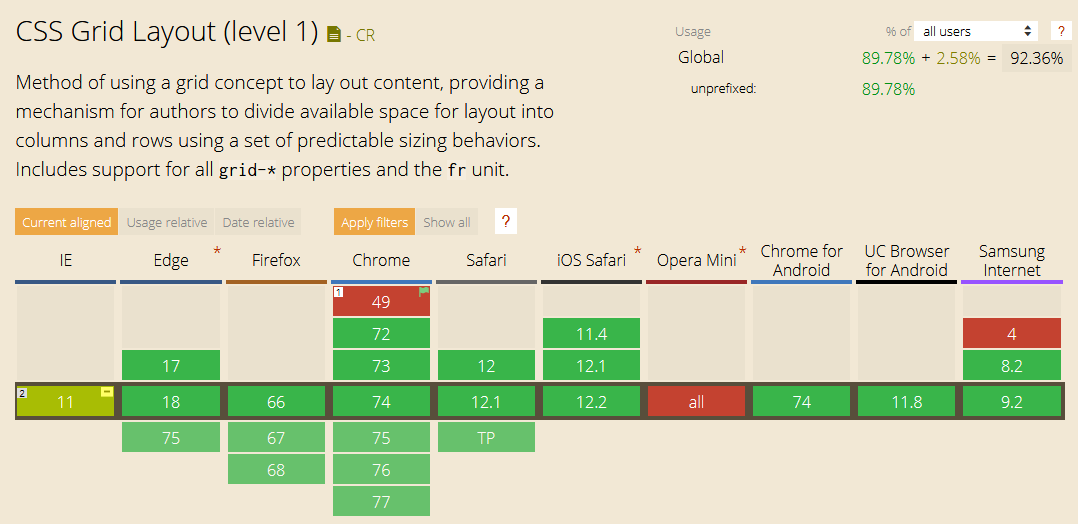
And [the last link to a game for learning CSS grid](http://yandex.by/clck/jsredir?bu=h74s&from=yandex.by%3Bsearch%2F%3Bweb%3B%3B&text=&etext=2164.1ygUnDnsrBz0w8csLfUdjNlLGbS51bEm48bNL4-omr9uhJJeWELY6mFJ-lqLFEmo.d612d9e29446c913beb0b53df793ee2a1d3e899a&uuid=&state=PEtFfuTeVD5kpHnK9lio9dFa2ePbDzX7fH_cbK-eu2V8J4cbFpzDXW9rYi7SZhqEJIq8fXhATr5ZDzTC0wt8G5NugjsS9BAsBeCvIwn5QlQ,&&cst=AiuY0DBWFJ5Hyx_fyvalFAMZLrYSK0vZTglc6SDXrCTpvaWXWBLaomg8BTjEbYcly-TVySF8RLno2yXJLw2N7cVx90FVKNd6hDvQP8RIsRdZg5ztU8dkVxTUifkOHkB9lajrcVhBh8PIDWQyFay2eur6WM4AUzIclxiMBR8t3OZmMVS0rjmUfH2kVPonI1J-sVW7t1zrg23fI3Ekz0qCb2veiYxg4KzozCk6kICZs3vQjzUqKjDtaD65V41QcqfxkZtlF-_-ZjafS93IXAbzid04CLvIiX0fI06K1duu0YTcUPyR7jroZsYxUj0XMtDNejZxQxXuGAeLeou6qeXA_uA4jXpYABdk&data=UlNrNmk5WktYejY4cHFySjRXSWhXSlB4SFZjQ3R6V29KT3ZjOTZkdEhlRl9CdmlLRHJvam5xb3dwdVA5eUxSbGtzaVhGbDBKbW8yQ01yUnFXVWdpajhMd25BY1owNEtYLXBId0xFUjlvOVUs&sign=0f1cd87f22ca98f432ec552a828bea88&keyno=0&b64e=2&ref=orjY4mGPRjkHVRqRT7scnl9k3ZfzgjFjPVyepuGQ4zJn3Cld8-md6chxPvsvF7tfESUS-_9Lp4Y,&l10n=ru&rp=1&cts=1558881220460&mc=1.584962500721156&hdtime=1462)

1. Grid Garden (<https://cssgridgarden.com>)

Browser support

How you can see As of May 2019 of the most popular browsers support for CSS Grid: Chrome, Firefox, Safari, and Opera.

Global support is 92.36%, and I think this is a good sign to start using the grid.



What is CSS Grid Layout?

You can find the following definition on MDN:

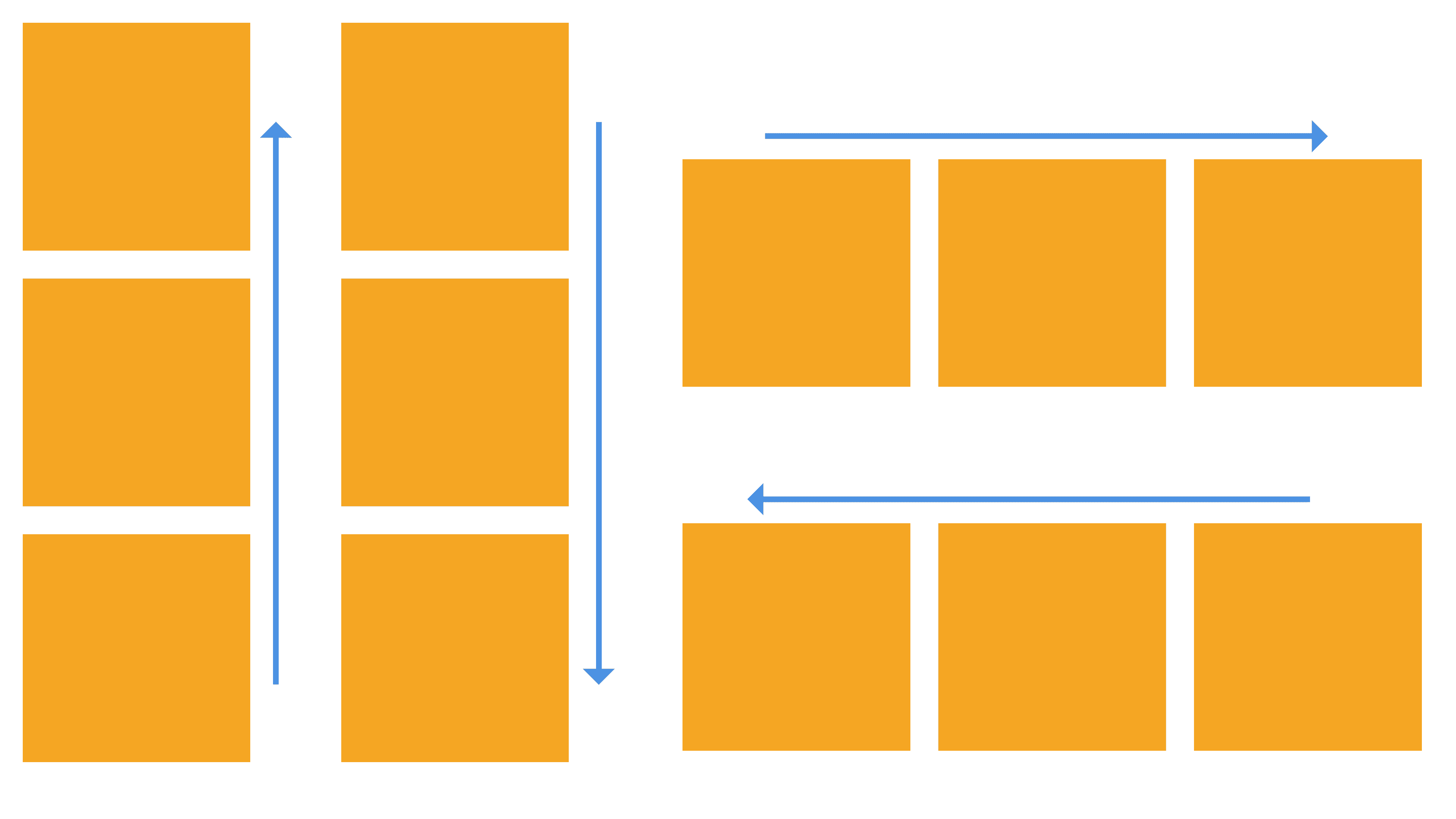
“[CSS Grid](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout) introduces a two-dimensional grid system to CSS. Grids can be used to lay out major page areas or small user interface elements.”

“a two-dimensional grid” – what is it? What does it mean?

Flexbox Model

I think many web developers already know about flexbox.

In flexbox we can arrange content horizontally or vertically – It is one-dimensional grid system.

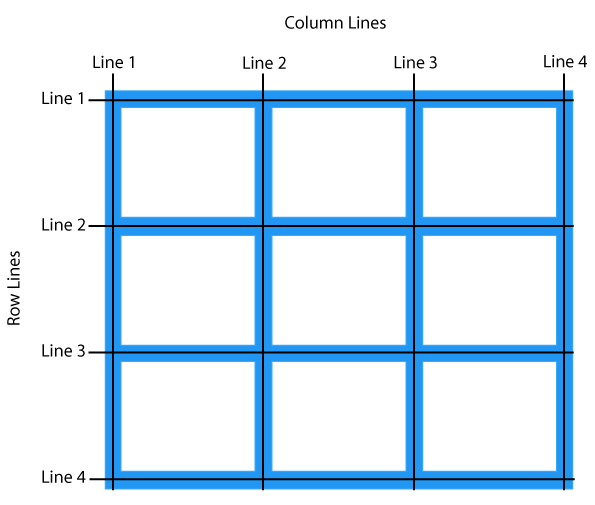


Grid model

But in Grid we can arrange content horizontally and vertically at the same time.

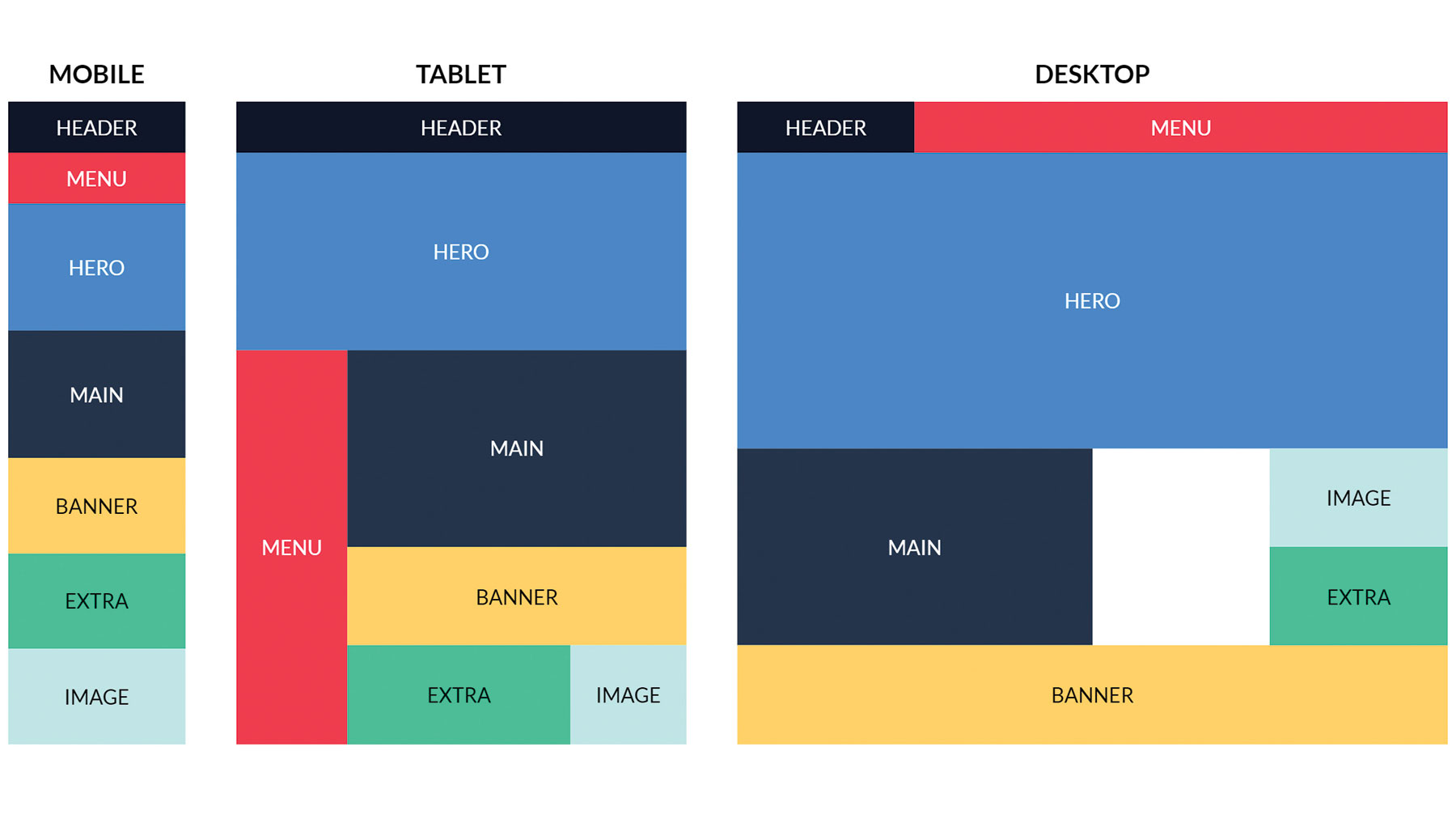
The Grid model consists of intersecting horizontal and vertical lines.

Vertical lines define columns, and horizontal lines define rows.



This means that the positioning of the element has become very simple. And of course an example:

Examples

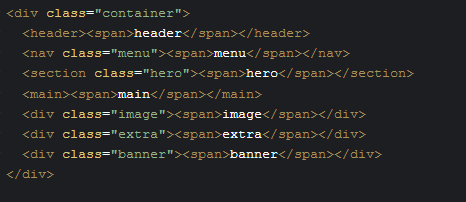
This is example of a typical web site

I think it will be quite difficult to do this with the help of old technologies. But I will show you how to quickly and easily create this.

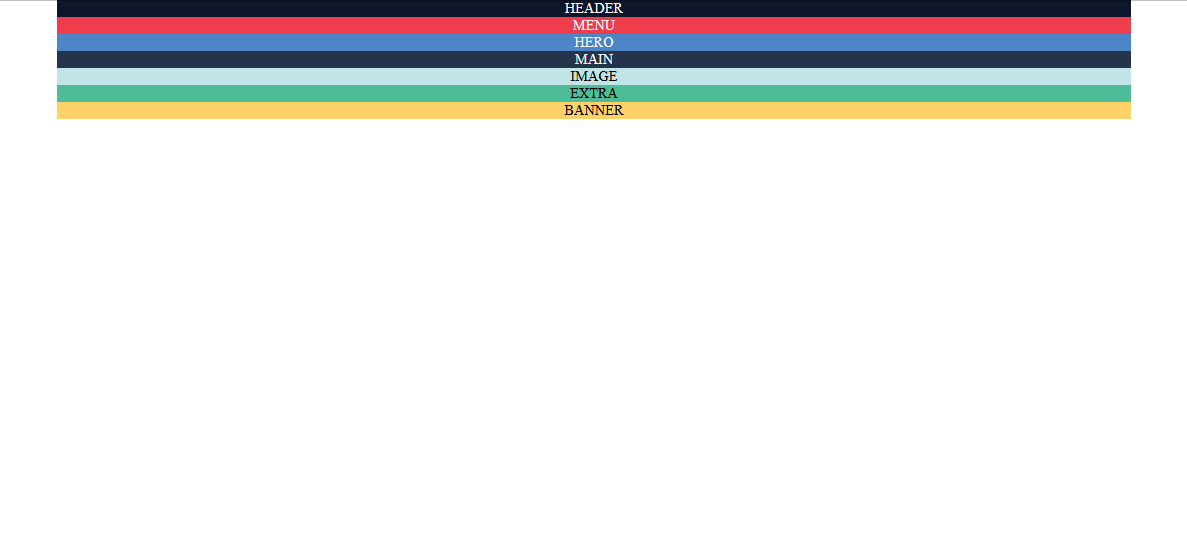
Get started

Step1: create html markup  
In the first, we have to create HTML markup and to define some decoration properties of html elements: max-width, color, background-color, text-align

HTML:



And now let's see how it looks in the browser. Nothing interesting happened.  
In browser:



Step2: Set CSS properties

To get started we have to define a container element as a grid with [display: grid](https://css-tricks.com/snippets/css/complete-guide-grid/#prop-display), set grid-template-areas property, and then define grid-area its child elements.

**grid-template-areas**

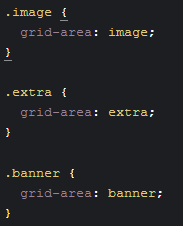
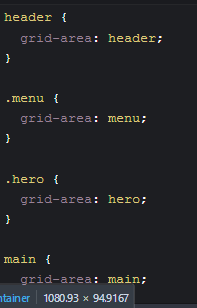
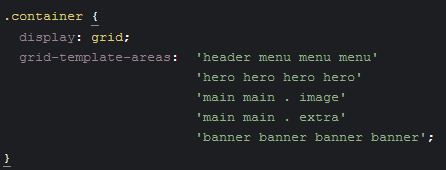
Defines a grid template by referencing the names of the grid areas which are specified with the [grid-area](https://css-tricks.com/snippets/css/complete-guide-grid/#prop-grid-area) property. Repeating the name of a grid area causes the content to span those cells. A period signifies an empty cell.

**grid-area**

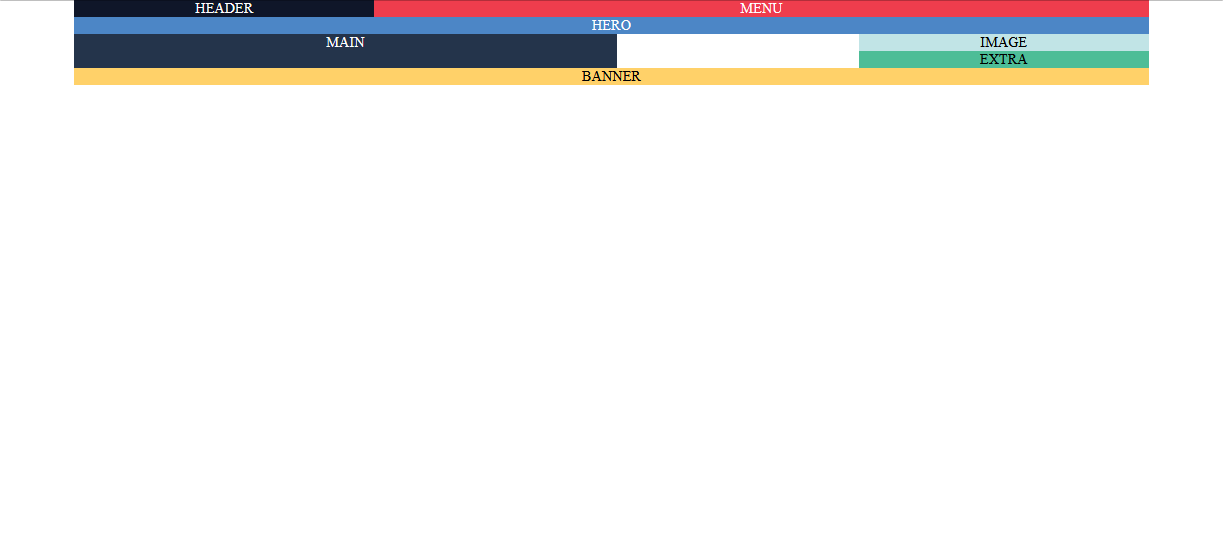
Gives an item a name so that it can be referenced by a template created with the [grid-template-areas](https://css-tricks.com/snippets/css/complete-guide-grid/#prop-grid-template-areas) property.

CSS:

we defined grid-area to children elements and defined grid-template-areas to grid container



In Browser:



I think this is starting to look like our template

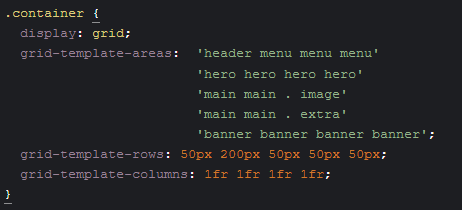
Step3:

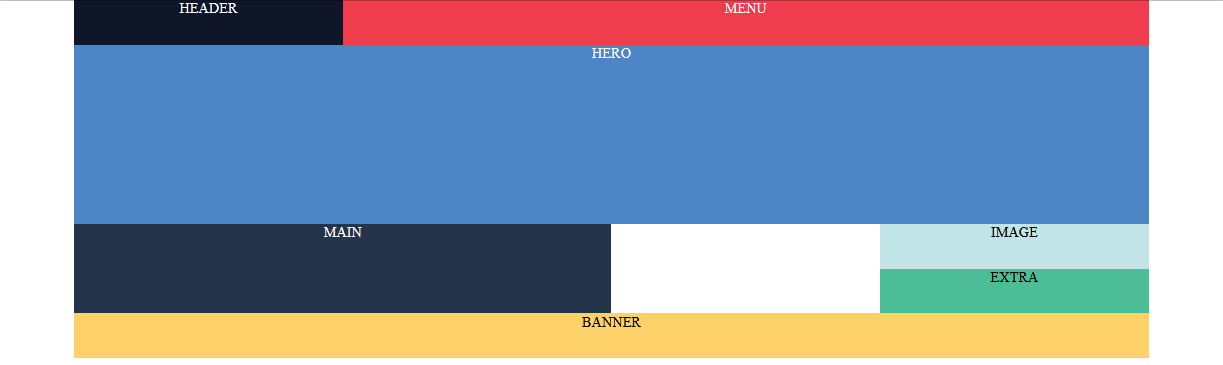
we need to set the container properties of grid-template-rows and grid-template-columns

**grid-template-columns  
grid-template-rows**

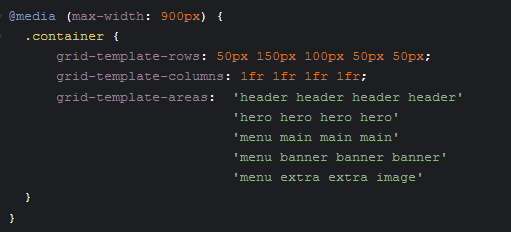
Defines the columns and rows of the grid with a space-separated list of values. The values represent the row or column size, and the space between them represents the grid line.

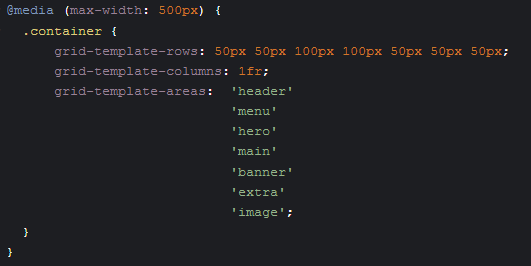
their values can be a length (px), a percentage, or a fraction of the free space in the grid (using the fr unit).

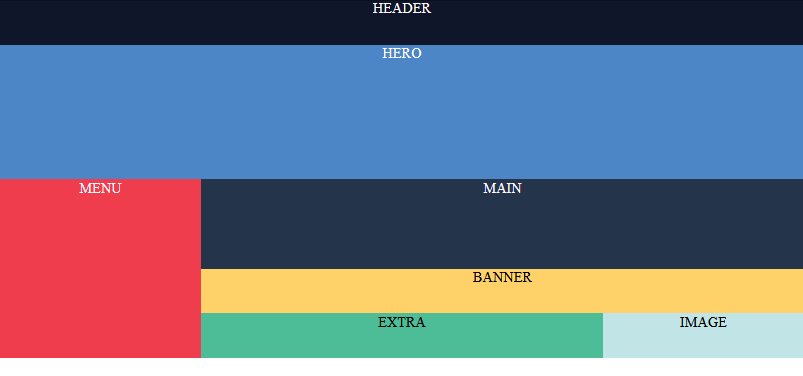
CSS:  


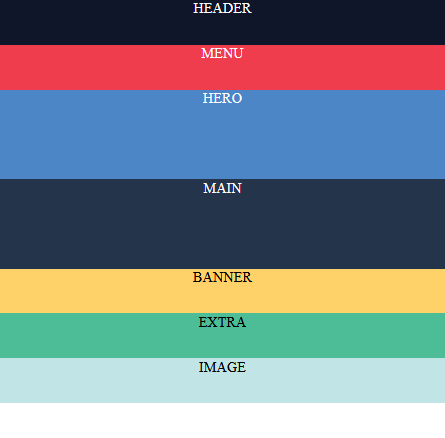
In Browser:   


Step4:

After that we can set media queries for our grid-container:   
  
CSS:  


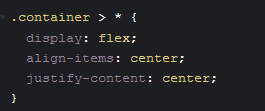


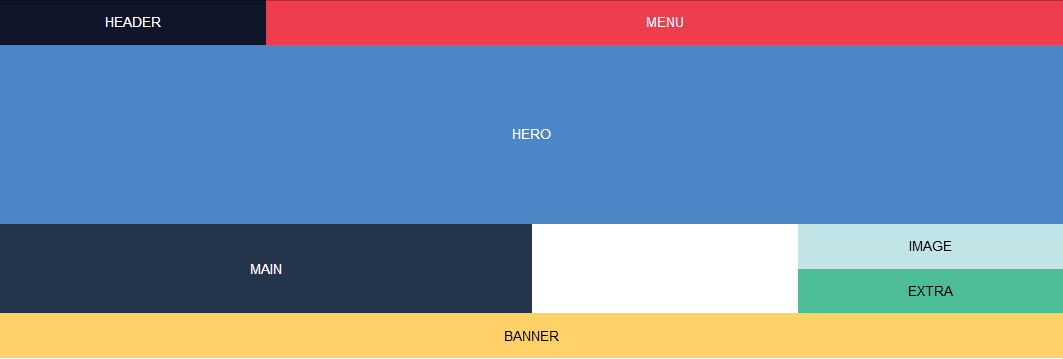
In Browser:   
Window width is 900px:  


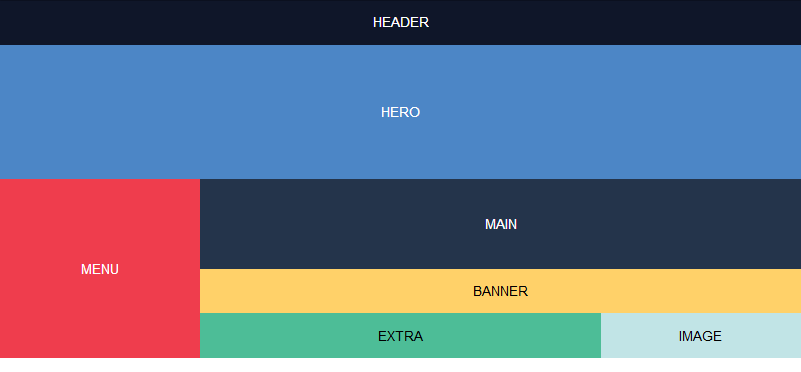
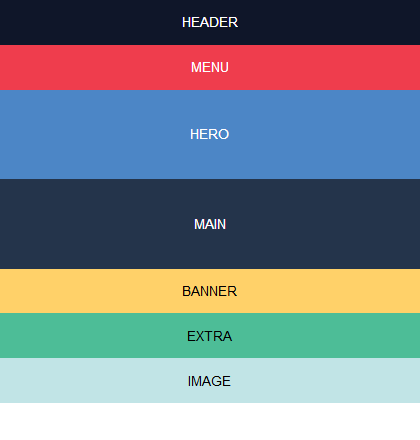
Window width is 500px:  


Final step:

and finally let's define the children following CSS properties: display: flex; align-items: center; justify-content: center;



In browser:   


Link to CodePan (https://codepen.io/vitali\_a/pen/arGaMe)

Last word

Of course, this example was very simple and I did not consider many other useful properties such as: grid-gap, align-content, justify-content and many others. But, as I said at the beginning, this is a demonstration of opportunities, not a guide.

My goal was to show you what happens if you start using the СSS grid right now.

I think you liked his features.  
  
See you next time! Bye bye!