1)

Today I am going to present about CSS GRID LAYOUT.

First let's talk about some of the basics and terminology of CSS creative . It'll help us establish a mental model of how things work and so we'll be able to understand grid a little bit better.

A grid is a set of intersecting, horizontal and vertical lines, that create columns and rows. Elements can be placed into the grid between these lines and into the columns and rows.

So the first basic things you need to understand is that a grid can also be ref(e)rred to as a grid container. When you establish grid and start to apply CSS grid to your work, you're dictating this container to become the grid container. You can think of a grid container as the parent element that's holding all of the children, so the parent is the grid container, and the children are going to be called grid items.

2)

When you start using CSS grid and building out grid containers, what you're in(e)vitably going to be doing next is building out grid tracks. The word track can be used simultaneously between horizontal or vertical tracks when the thing is about columns or rows. A grid track is the space between any two lines on the grid. When we start using CSS grid and creating grid tracks rows and columns, what CSS grid is gonna do, is to start creating numbered grid lines and so you can see here the graph what we have on the screen as a result. It's inevitably taking those rows and separating them into lines, so that we can navigate the different spaces inside of our grid.

3,4)

A grid cell is the smallest possible unit on the grid conceptually. It looks like a table cell, think of it like an Excel spreadsheet, each individual cell represents a larger part of that spreadsheet. Well that's the same thing here in CSS grid, this one square represents a single grid. So Items can span multiple cells like we can see here and then become grid areas. Grid areas must be rectangular there can be no l-shaped kind of abstracted grid areas at least not now. Maybe in a future version of the grid there could be this option, but right now they must be rectangular areas.

5)

Okay with all that being said let's look up a code and build out a little bit of a layout using CSS grid. In my HTML I have a basic div with a class of wrapper and I have a bunch of divs with class of item and they're just numbered from 1 to 6 in my CSS. First property that we're gonna learn that belongs to CSS grid is the display. Let's go ahead and set the property display to element with class "wrapper".

6)

We can define a fixed number of lines and tracks that form a grid by using the properties grid-template-rows, grid-template-columns, and grid-template-areas. This manually defined grid is called the explicit grid.

7)

If there are more grid items than cells in the grid or when a grid item is placed outside of the explicit grid, the grid container automatically generates grid tracks by adding grid lines to the grid. The explicit grid together with these additional implicit tracks and lines forms the so called implicit grid.

8)

now it's time to go through and talk about some of the most popular CSS grid properties

These properties defines the columns and rows of the grid with a space-separated list of values. The values represent the track size, and the space between them represents the grid line.

9)

If we want each and every column to be the same width we might use clever function called the repeat function. So instead of writing individual numbers we can write repeat. There's two things that go inside of the repeat function. They are the amount of times you want to repeat it and the size of what is to be repeated. So we could say hey we want three columns and we want those to be 200 pixels each

10)

we can create gaps between our rows and columns. there are already pre-defined gaps. That's how it's initially set inside of the grid if we want to expand the gaps in between rows or columns. We could do that by using grid-column-gap and we could say maybe we want it to be 10 pixels and we now have gaps in between our columns we could do the same thing with grid-column-grid 15. In addition, short hand way to write gap if we have same parameters of row and column it set grid-row.

11)

How we can see I've created a class on my fifth item and I've called that class item 5. We can set individual item's property inside of the grid. For instance I could set grid-column and then I can span that column by just writing in span and 2 areas. Element is gonna spend not only its column but span three columns

12)

we can also be a little bit more specific about where your item starts and ends inside of the grid. We could say a grid column starts and we want to start at line one and then we can have a grid column end at line three.

And when we do that we're gonna get it started at line one it's gonna past line two all the way to line three.

grids are really really smart and it's a great handy Butler robot that does exactly what you tell it to do and it makes smart decisions when it can't do exactly what you tell it to do where you've told it to do it .

Thank you for your attention and I hope you’ve enjoyed the presentation.