In this video, you will learn about how differently

commonly used grid layouts are

created and where you can use them.

Let's start by examining

the grid layout named the pancake stack.

I've opened extra.HTML in Visual Studio Code.

Now I can start adding the different elements.

First is the restaurant's name,

little lemon, which I put into the header.

In the main element, I type menu and for the footer,

I type About Us.

After saving, next,

I will open up extra.CSS.

First of all, write the rules for the body,

followed by the header name and then the footer.

First for the body,

I need to use the display property to code the grid.

I define as high as 150 VH.

One VH is equal to 1 percent of the viewport height.

Now, I add the grid template rows

property that will track

the sizing functions for the grid rows.

I set it to one FR or one fraction for the main section.

Now, I define the rules for the header.

I start by setting the background color

and follow that up with text alignment,

which is centered and 2EM padding.

2EM is a relative height which is

twice the size of the current font height.

Let's examine the output.

It looks good. I'm going to do the

same for the main and the footer elements.

I've changed the colors, but everything else is the same.

Let's examine the output again.

It looks exactly like I was expecting.

The header main and footer are

all formatted in different colors.

If I modify the size of the page,

the layout changes accordingly.

Next, let's examine the 12th section

or 12 column grid system layout,

also called a twelve span grid.

This type of grid, conceptually to divide

the page into 12 columns of equal size.

One can add different elements to the grid by determining

their starting point and

the number of columns each element spans.

Don't worry, this will all become a lot

clearer when I go through the following example.

I'm opening gridtwelve.html in Visual Studio code.

First, I create a container with class parent

and inside it I'll add six items that I call segments.

I had a class name for each of the div tags matching

the span or width I'm going

to give each of these segments,

these widths are randomly assigned.

Now, I'm going to save and open up my gridtwelve.css file.

I'm going to add CSS selectors for each class.

I will first add CSS properties for the parent class,

and then for the individual segments are defined.

For the parent class, I write

display grid and then add

the grid template columns property,

which I'm going to repeat 12

times with one fraction each.

Next, I define the properties of the segment.

Again, I display grid

following up with centering the items at the text.

I set the grid template rows property.

I want the segments to be a little wider,

so I'll set it to 100 pixels.

Let's examine the output.

It's not much for now, but let's continue.

Now, let's set the properties

for the different segments.

The first one spends 12 columns,

I start by adding a light green background.

The next part is important.

This is where I use the grid column property.

This value tells the element to start from

the first column and span all 12 columns.

Let's examine the output again.

These two segments are now in better shape,

starting from the first column and

spending all the way to the 12th.

Let's add another selector for eight.

Again, I copy the same code,

but instead of starting from the first column this time,

I'll start from say, the fifth column,

and it will span across eight columns.

Let's change the color here to bisque.

The color now makes it obvious that the elements

starts from the fifth column and spans eight columns.

Let's add the other segments and

then we'll examine the code again.

Let's examine the output again.

Now, this is looking a lot more presentable.

There's also another way to view these columns.

Click on "More browser auctions"

and then on "Open Devtools" pane.

You can select the button in the

corner to select an element

on the page to inspect then scan over the grid.

The page is systematically

divided into 12 different columns.

You can configure your code

according to your requirements.

Grids like these are really useful when it

comes to designing something like a magazine,

newspaper or page layout,

which can replicate these models.

Now, you know about the basic grid layout

in the 12 column grid.