So far, you've covered different ways in which a flex box can be used to add

functionality to a web page.

In this video, you're going to explore the versatility of flex boxes even further

by using them to create bar charts.

Bar charts are a common way of visualizing data used in statistics.

It's easier to compare and understand data when viewing it in the form of rectangular

bars that have heights in proportion to the values they represent.

The different bars stacked alongside represent the classifying criteria.

There are sophisticated data visualization tools available today that support web

pages but it's fairly straightforward to integrate data using CSS.

CSS provides a straightforward way to do this by using flex box,

with flex boxes, you can generate both vertical and horizontal bar charts.

These bar charts are called flex charts.

The most important property when using flex is the flex property that in turn has

the following three sub properties.

Flex grow, which specifies how much the item will grow

compared to other flexible items, flex shrink,

which specifies how much the item will shrink compared to other flexible items.

And flex basis, which specifies the initial length of the flexible item.

Flex property can be declared with something called a shorthand notation

such as flex 01 auto, where 01 and auto are the default values for

the flex grow, flex shrink and flex basis property.

Let me demonstrate each of these types with the help of an example,

clients visiting the little Lemon restaurant were given a survey to pick

their preferred dish.

The management of little Lemon decided to publish the results on their website

to draw attention to their most popular dishes.

I've been asked to write code to make this possible.

I've set up my basic html document structure in a file

named flex charts dot html.

I begin by creating a class called row by writing dot row and

pressing enter to get the div tag.

This will be a horizontal group across the page that will display the result.

I create two tags inside the div tag, one for the label and another for

displaying the results of the survey.

To add the label, I type dot label on a new line and press enter,

I then add the name of the label preferred dishes.

Next I create another class called container by typing dot container and

then I press enter.

Inside the container I create a class called dish by typing dot dish and

pressing enter as there are three dishes in the popular list, I copy and

paste this item twice.

To distinguish between the dishes, I add a class name for each of the items

after the word dish I name item one pasta, item two, pizza and item three burrito.

I now have to add styling properties as I am using flex basis.

In item one I add flex basis as the property to specify the initial length

of the item.

I add the percentage to indicate the length of this flex item.

I do the same for the other two items as the total here should be 100,

I adjust the percentages of item two and three.

Finally I add the respective labels for each dish.

To do this,

I add the percentage between the angle brackets at the end of each item.

Now that my code is created, I move to CSS by clicking on the flex charts dot CSS

file at the top of my screen and checking that it's referencing the correct file.

I have stacked the CSS file next to the html file for better understanding.

I start with the outer most container which is row by typing dot row and

then pressing enter within row.

I add the display property to which I add flex, I press enter to move to a new line.

I add another property called align items to which I add stretch.

There are two main axes in flex, the main axis that runs horizontally across

the page and the cross axis that runs vertically align items,

controls the alignment along the cross axis.

Next, I address the label div tag which consists of the flex property and the font

family flex as you previously learned is shorthand for flex grow flex shrink.

And flex basis since this label is a part of the row,

I add dot row in front of dot label.

Now I add the container that will consist of all the results of the survey,

I defined only the flex code value here, just like align items.

The justify content property defines how the browser distributes across

the main access of the flex container.

So here I set the alignment to flex start which is the left hand corner of my

web page.

Next, I define the property for the dish selector.

It is the same as for the container above but here the alignments are centered.

This will become clear once you see the results,

I have one more set of selectors to define.

But first let me display the output.

>> To do this, I switch to my HTML file and

click on the display icon in the top right corner of my screen.

The output preferred dishes and the percentage is 35, 50 and 15 appear on

my screen, to improve the visuals of my chart and make it easier to interpret.

I'm going to add individual colors to the chart items at the bottom of my

CSS file I press enter twice and type dot pasta.

I press enter and within this div class I add background color,

followed by the hex code of the color I want to use in the next line.

I specify the background color of pizza.

Lastly I specify the background color of burrito.

Once again I display the output.

Now the percentages display the background colors as indicated.

What's important to note here is that as you change the size of the web page,

the flex adjusts itself while the label remains unchanged.

The next example is very similar to the one I created for the earlier bar charts.

In this example, when I create horizontal objects, I first add four containers for

male and female and then I set the flex spaces for each of them.

If you look at my CSS code, I follow the same order.

First I address the columns which is the outermost container,

then the class container.

And after that the div class bar inside it, and

finally the individual properties that I want to define for male and female.

The main differences here are in terms of the line items, justify content and

flex selection that have been adjusted to accommodate the horizontal bar charts.

When I display the output of this code, the male and

female classes are stacked on top of each other.

What alignment to use may or may not come intuitively, but

if you look at a few more examples, it will start making sense.

Like in this case the container contains male and female classes that need to be

stacked on top of each other, which is why I have used flex direction.

That was a lot of information about the many uses of flex boxes.

So you should have a good idea of how to use them in your own coding now.

The graphical representation of information or

data makes it accessible and allows you to see and understand trends and patterns.

In this video you've explored ways to create bar charts using flex box.

Flexcontainer

That’s not correct! The property flex-direction defines the main axis inside the container.

Justify content

Incorrect

That’s not correct! Justify-content is used to align items along the main axis.

Align Item

That's correct! The property ‘Align items’ aligns multiple lines of items on the cross axis.

/\* This is just how I think about the rule above in my head \*/

.child {

flex: [flex-grow] [flex-shrink] [flex-basis];

}

/\* or... \*/

.child {

flex: [max] [min] [ideal size];

}

Use my height or width.”

“please, please, please just try and take up 1000px of space.”