







[MUSIC] Responsive design was born

primarily out of the need to deal with the explosion of mobile

devices that started being able to browse the web much in the same way that

a desktop browser user browsed the web. I found this line graph on the web and it's attributed to some research

done by Morgan Stanley Research. I don't really know how exact

any of these numbers are, but the general point is well known. Browsing on the mobile devices exceeds

that of regular desktop browser. And if you look at this graph, that sounds just about right, that this

actually happened around 2013, 2014. So it's been a little

bit since that happened. So, the point is,

you just can't ignore mobile. You have to design your sites for

mobile, as well. But when it comes to mobile,

there's so many devices. So, which devices do we want to target and

which devices do we want to support? And the answer is really, all of them. You can't really afford

to ignore any of them. There's such a plethora of

these devices out there, and ignoring some of them would

ignore a chunk of your user base. So how does responsive

design help us do that? Well, I think the answer lies

in what responsive design is. So what is a responsive web site? Well, that's a site that's designed

to adapt its layout to the viewing environment by using fluid,

proportion-based grids, flexible images, and CSS3 media queries. And by proportion-based grids,

what we mean is or hinting to Is that

the width of these grids or these columns as we'll see in a moment

should be specified using a percentage. I really like this visualization

that somebody put together and put on Wikipedia when you search for

responsive design client. So this is one of the images on that page. And this is basically paraphrasing or

maybe actually taking verbatim, a quote from Bruce Lee,

a famous martial arts expert. And basically it says you put water

in a a cup, it becomes the cup. You put water in a bottle,

it becomes the bottle. And you put it in a teapot,

it becomes the teapot. And the idea is pretty simple that

content should be like water. You put it in whatever different

devices you put it into and it should still display,

the content should be the same, and the information conveyed

should be the same. And perhaps the only thing that's

different is the shape of that content, right? The way it's being displayed. So using kind of the idea of the previous

graphic, we can certainly understand now, that the site's layout is supposed

to adapt to the size of the device. In practice,

most of the time if not all the time, this means that the site should adjust

based on the width of the device. And I should also point out

that content verbosity or its visual delivery may change. For example for a restaurant site,

a phone number which used to be displayed somewhere in the site without attracting

too much attention to itself for a desktop version of that

site Is now front and center, and the most prominent

item on the page on a mobile site. And that's, perhaps, because we

want people to have their phones. Once they have their phone, to be able to

just tap and call, and put their order in, or perhaps, we just don't have

the flexibility in the mobile device. To let them go find that

phone number somewhere else. Or just glance at it because of

our real estate being so small. That's number one, number two is

some parts of the site may even be hidden on the mobile version

as we now have to measure very carefully which parts

are the most crucial. Now, are there any alternatives to

the responds of design principles? Well actually they are? What you could do is you could have

a service high technology that detects your user agent, in other words

the type of browser that you using and figure out whether or not you're on na

mobile device or on a desktop device and then based on our information either

serve up a regular desktop version of the website or serve up to the client

the mobile version of that website. And this is basically what the websites

that wanted to have a mobile version as well. Used to do before the responsive

design kind of came to the forefront. But the problem is there's a couple

of issues with this type of approach. Number one is, you have high risk

of feature diversion between apps, because now you really

technically speaking now, we now have really two different

applications running your website. One is a mobile version,

one is a desktop version. Number one, number two is, well mobile devices are just too

varied in size among themselves. So it's hard to make a mobile

site that satisfies every client. So what ends up happening is that you end

up sacrificing user experience because your trying to kind of make one,

mobile website that fits them all and that, basically, never works. So, nowadays, by and large, because of these issues, people

really stay away from this approach. The largest part of responsive design is,

obviously, the layout. And the most common layout out there or

responsive layout, is a 12-column grid responsive layout and this is what,

a bootstrap, Twitter bootstrap uses and just about almost really every responsive

framework out there uses nowadays. And, the reason 12 is what's used

is because of the factors of 12. You could basically, t's evenly,

nicely divisible by 1, 2, 3, 4, 6, and obviously 12 itself, which means,

we could sub-divide our page into sections that are evenly

split and nicely layout themselves. So for example we could have content

that is split into three columns out of the 12 so it's three, three,

three, three so that all adds up to 12. Now in case you're confused,

I just left some padding here, I kind of assumed there's some

padding just for visualization and be able to see the different

columns easier. And obviously you can six and

six in this layout. And four, four and four. And any other combination you want. You can have eight and four,

you could have nine and three and so on. When I approach these responsive layouts, Is that we know that the browser

window is 100%, right? That's the width of the browser window. Which means that we calculate what one

column in this 12 column grid layout is. That's 100% divided by 12. And that's turns out to be

about 8.33% which means when we specify in our framework,

in our bootstrap or whatever it is we were using,

how wide a particular column should be. We can convert this

threes into 25% because three columns is 25% of 12 columns. And the 6 columns become 50% of the 12

columns and the 444 becomes 33.33%. And obviously we're not limited

to having just one 12 grid. 12 column grid layout. We can certainly have nested ones so

if you have 4 4, 4, and 4. Inside each 4 columns we

could consider that 100% and then have another 12-column

grid layout inside of it. So here in the first in

the left one we see. 6 and 6. In the middle one we see a bunch of 3's. And the most right one we see 4, 4, and 4. And that's totally acceptable,

in fact is done all the time. Okay, so enough theory for

part one of this lecture. And in part two of this lecture

we're going to go ahead and jump into the code editor. And see these concepts in action.