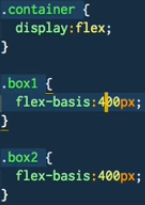
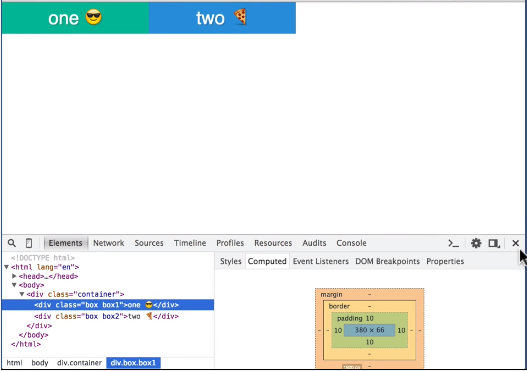
flex-grow / flex-shrink / flex-basis are packed into flex property.

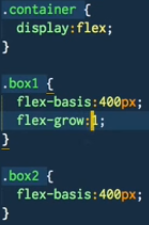
For example: If you say, flex:1, then it mean, flex-grow:1 and flex-shrink:1

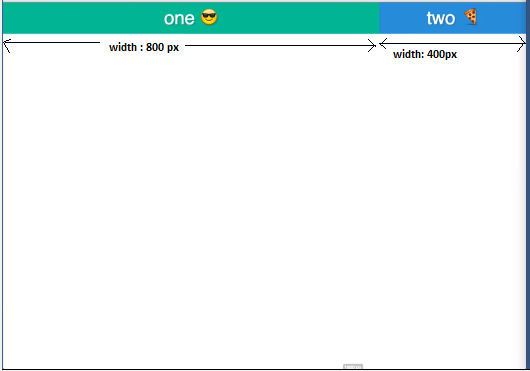
In an ideal world, the size of an element should be decided using flex-basis property.



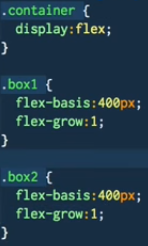


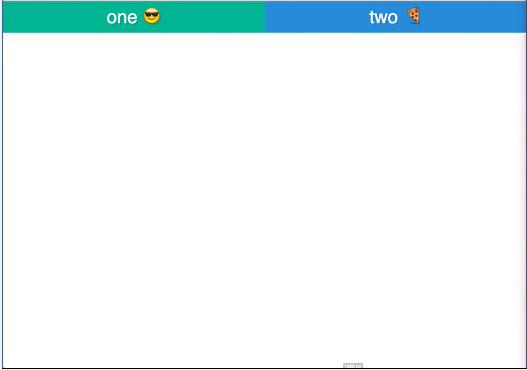
What happens when we have extra space? This is where flex-grow comes in.

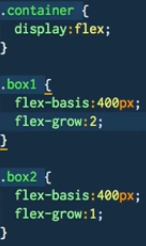


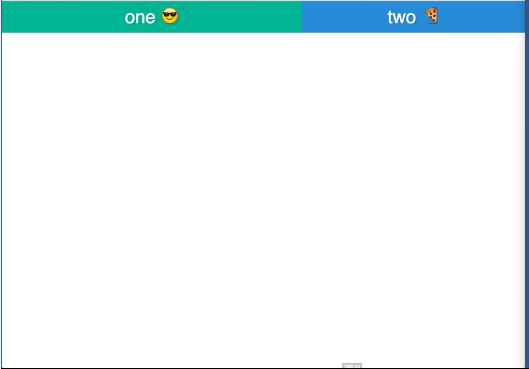


So, for box2, flex: grow is zero (by default)

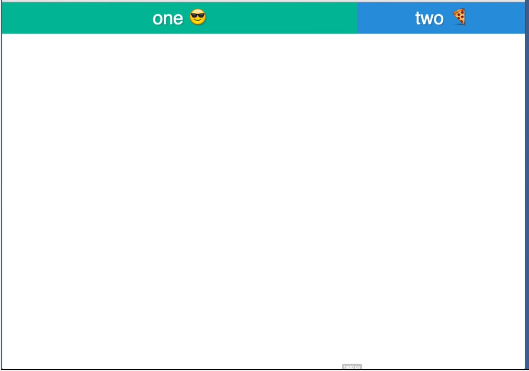




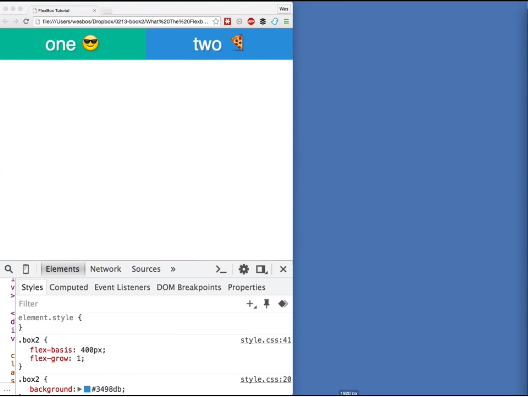




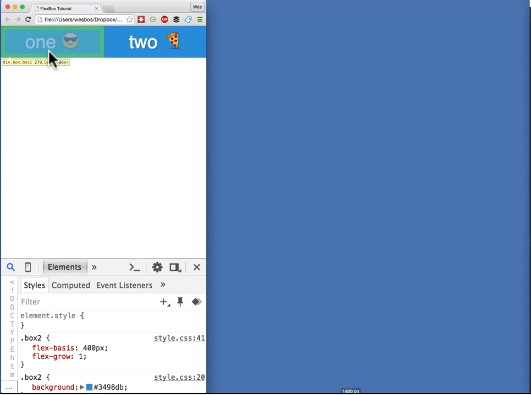


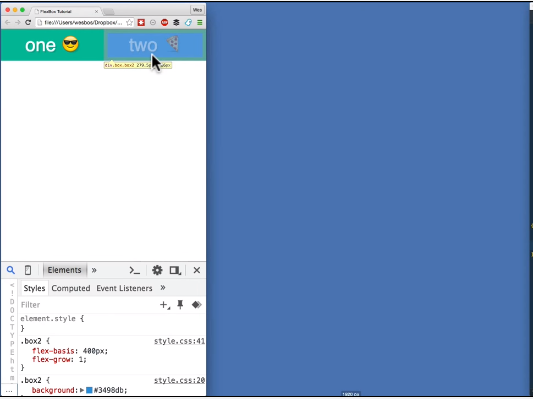


When the browser width shrinks, this is where flex-shrink kicks in. flex-shrink is 1. In the below diagram, browser width is exactly 800px, so flex-basis drives the size of each element. flex-basis can be in em or rem.

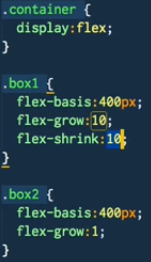


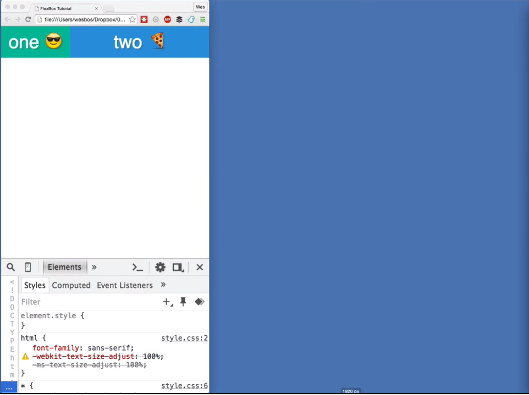
Browser width goes less than 800px, but as flex-shrink is 1(default), so elements have same width.





So, How much of myself should I give up, when the browser size shrinks? flex-shrink tells that.





Here box1 is shrinking 10 times to box2, when the browser does not give enough space for two elements.

Flex syntax can be combination of flex-grow flex-shrink flex-basis. flex syntax is also like flex: 10 5 400px;