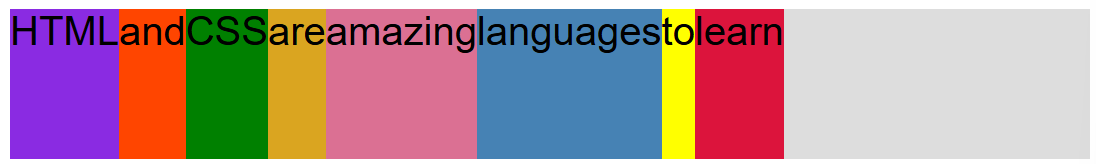
**INTRODUCTION TO FLEXBOX**

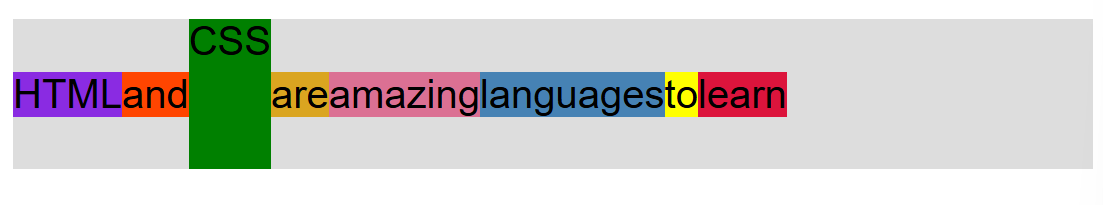
Originally:



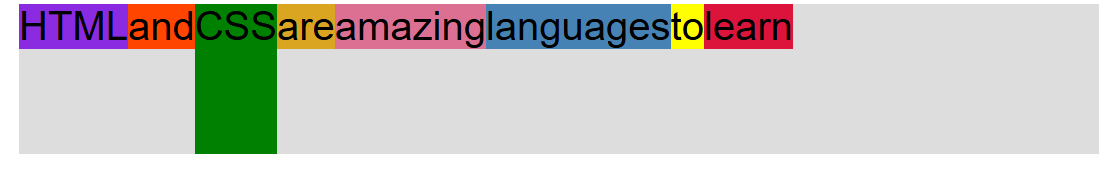
After applying display: flex to the container element.



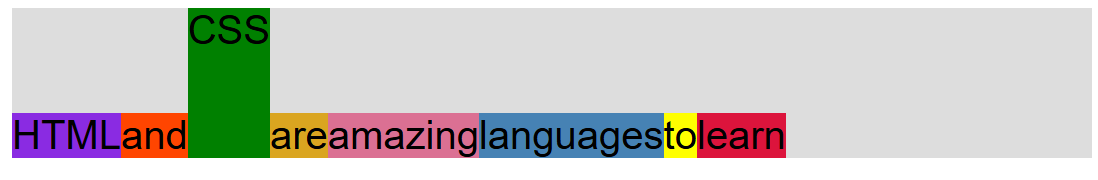
All of them are the size of green element because its height has been set to 150 px

After applying align-items: center  


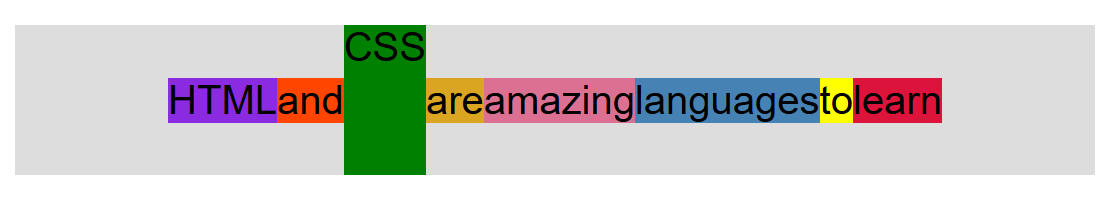
After using align-items:flex-start



After using align-items:flex-end



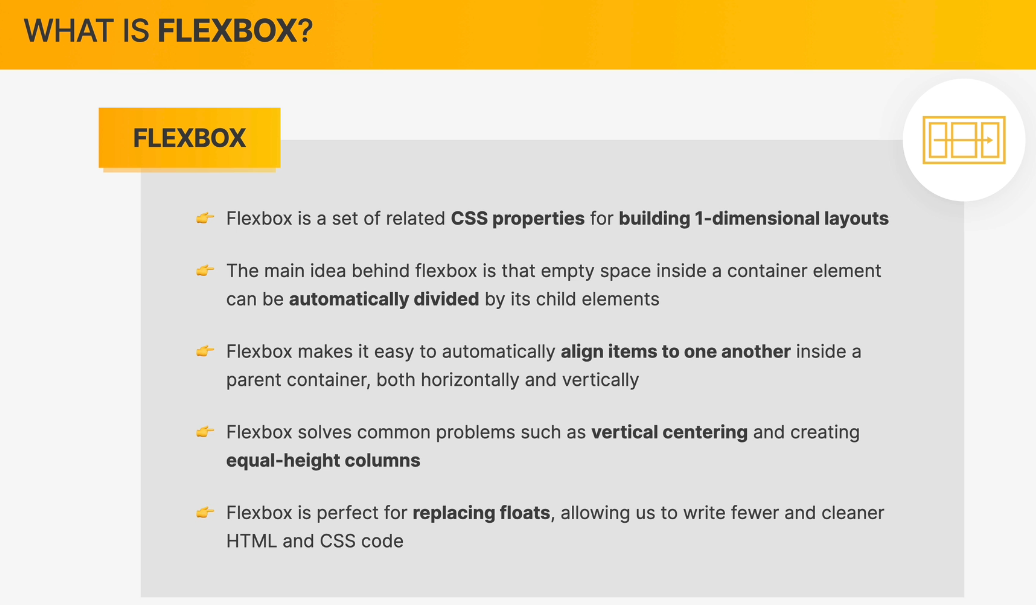
After using justify-content: center {the elements are in the center of the container

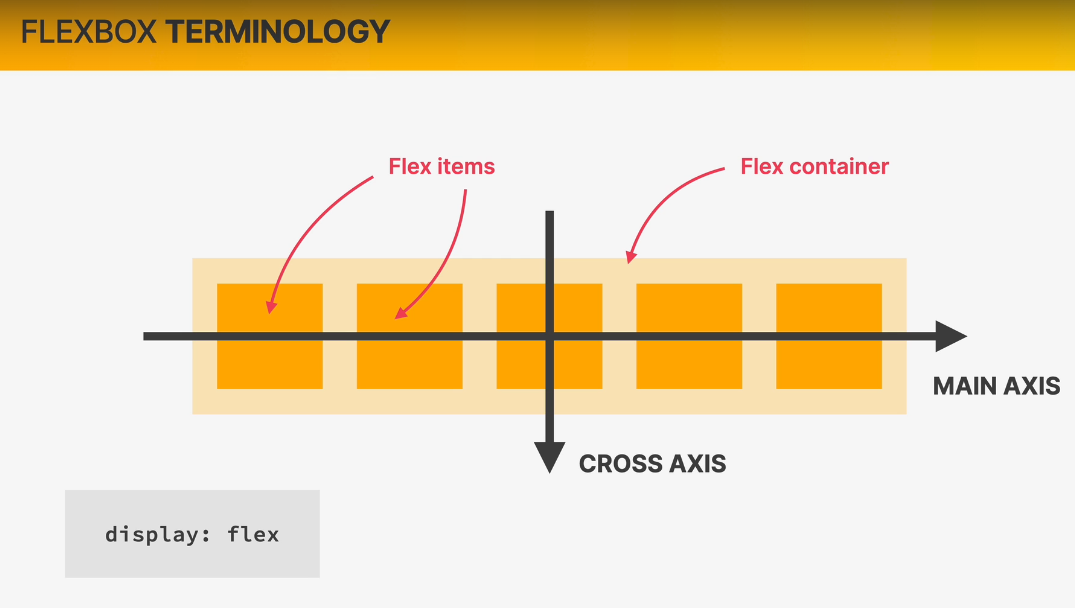


After using justify-content: space-between



**FLEXBOX OVERVIEW**

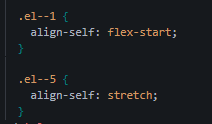
****

****

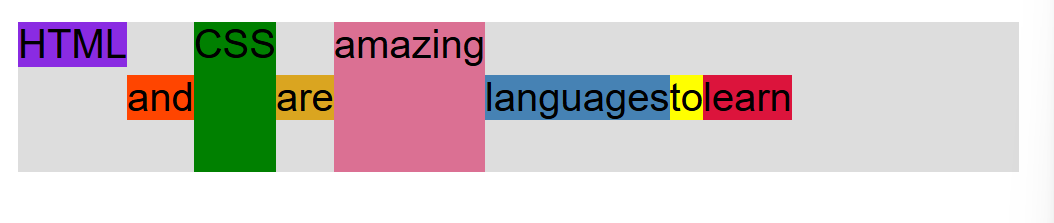
**We need to set the display:flex for the container to start with flexbox**

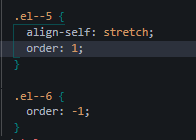


**SPACING AND ALIGNING FLEX ITEMS**



If we want one of the items to behave differently we use align-self



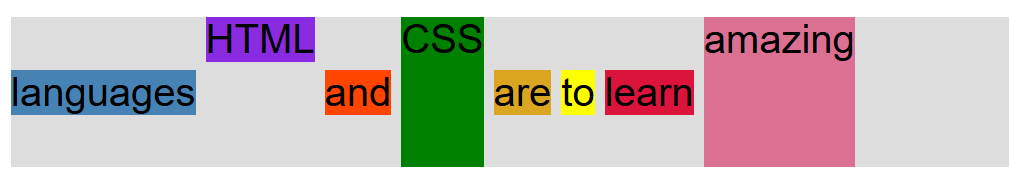


Initially the order of all of the elements is 0.

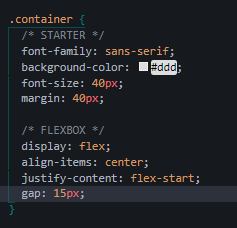


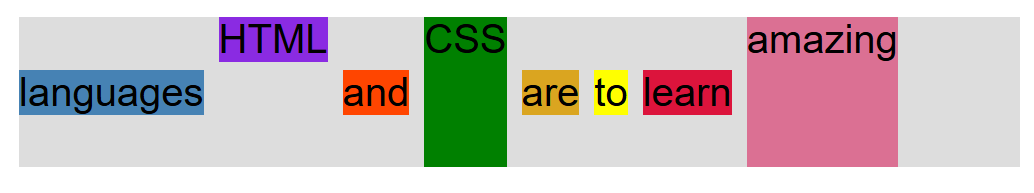
Adding spaces





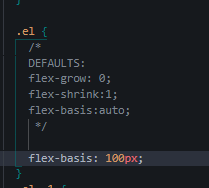
Instead of doing the above we can use gap property of the container

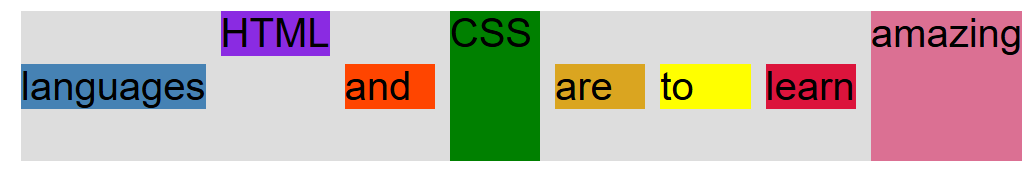




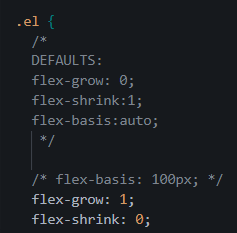
**THE FLEX PROPERTY**

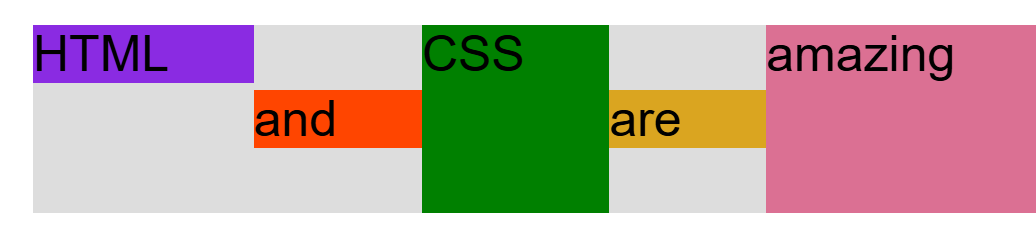
Flex-basis: is used to set the width of the elements. We don’t really use width we use this.





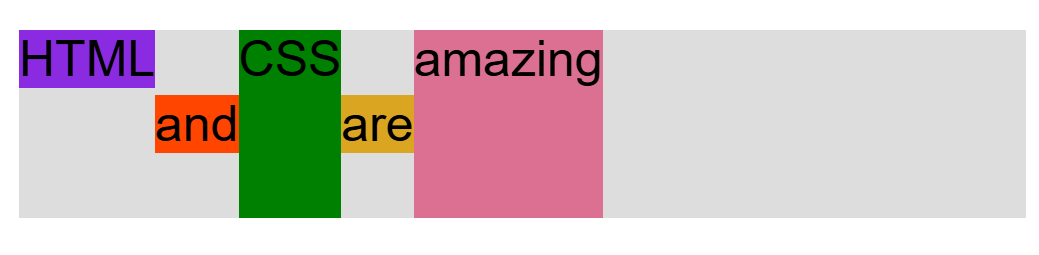
Flex-grow: 1

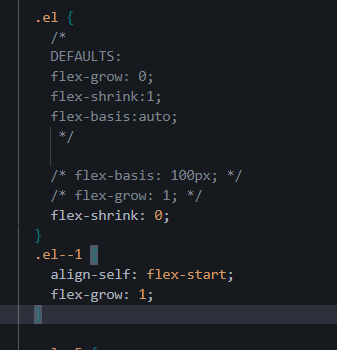


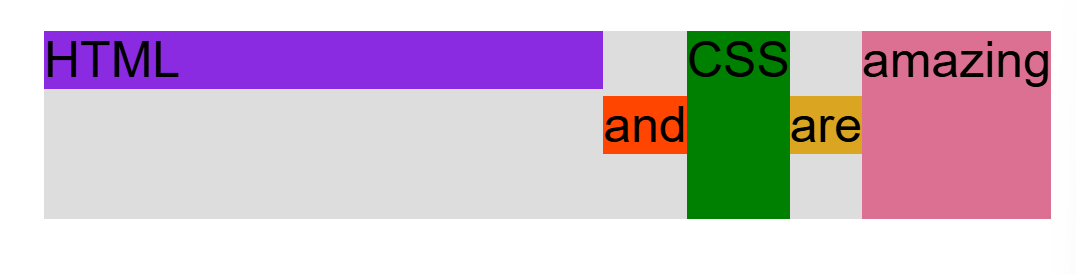


Sets equal width for all elements.

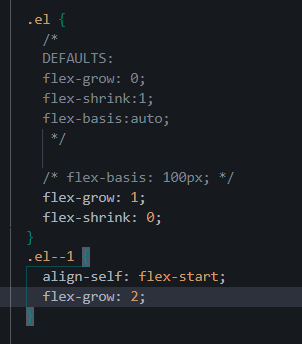
Flex-shrink: 0  
all of the elements will occupy the area they need nothing more than that



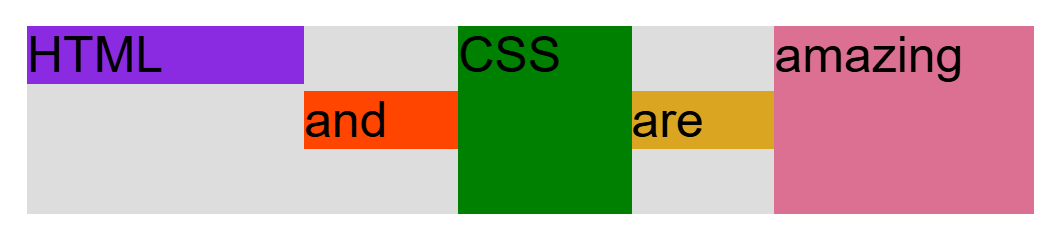




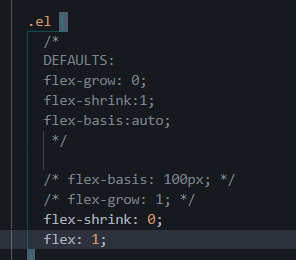
Now only this element is filling up the remaining space all of the other are using the amount needed.



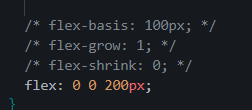
This element will get twice the remaining space not twice the whole space just twice the remaining space.



We just type flex: 1 instead of flex-grow it means the same thing



Short hand way to write:



First value is flex-grow then flex-shrink then flex-basis