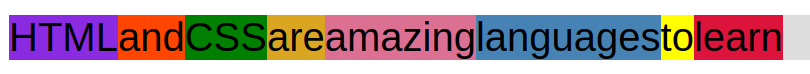
flexbox.html:

.container:

display: flex; // container中所有元素side by side, 并且每个元素box刚好容纳text，如下图：（其中高度比文本高度大是因为，文本CSS带有height,而默认的justify-content是stretch，即container中所有其他元素被拉伸成一样高）



如果把CSS的height属性去掉，将得到刚好容纳文本的boxes,如下图：



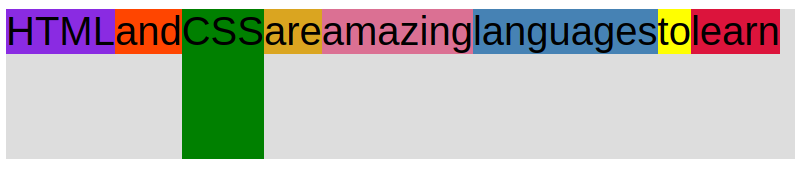
1. align-items:

恢复CSS的height: 150, 给container加上属性align-items

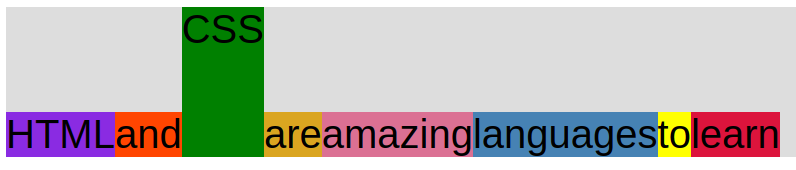
align-items: center; 如下图，除了CSS自带height不变外，其余文本均居中于box，且box在垂直方向也居中于container:



align-items: flex-start; 如下图，除了CSS自带height不变外，其余box处于上方：



同理align-items: flex-end; 如下图，其余box处于下方：



当然，还有其他属性值，可以一一测试，看看变化

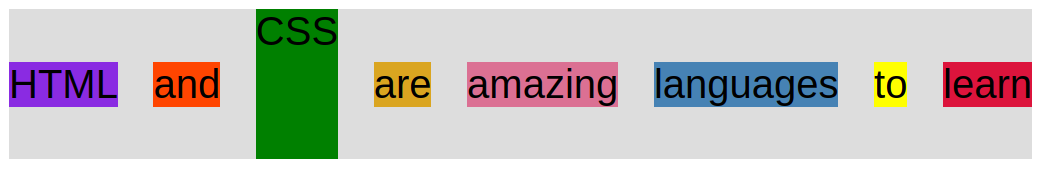
1. justify-content:

将align-items设置为center, 给container加上属性justify-content(默认是stretch), 所以其余box起初会拉伸和CSS一样高

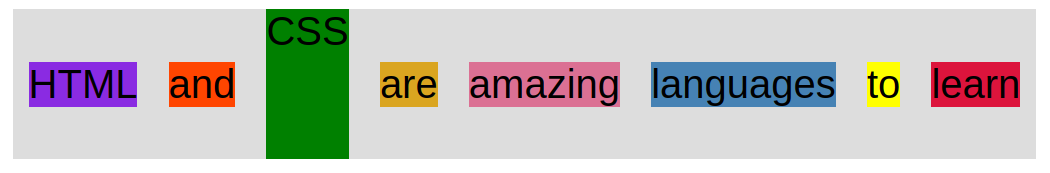
justify-content: center; 如下图整个container中所有元素boxes合成体，即彩色区块，居中于container中间（灰色区域）



justify-content: space-between; 所有boxes中间间隔一样，如下图：



justify-content: space-around; 所有boxes间隔以及和边缘的距离均一致，如下图：



其余属性可一一测试

1. flex-grow, flex-shrink, flex-basis:

flex-grow默认为０，el中设置为１时，所有元素将充满整个container, 如果文本内容均小于平均宽度，则所有文本box宽度一致，宽度超出平均宽度的元素，其box自适应填满文本。也可以在单个元素中使用，比如将el--1的flex-grow设置为２，则el--1将占有比其他元素两倍的宽度，由于box宽度根据文本内容自适应的存在，el--1的宽度不一定刚好是其他元素宽度的２倍。

flex-shrink默认为１，即所有元素会收缩适应container的尺寸，如果设置为０,则元素可能根据其自身的宽度超出container的尺寸。

flex-basis默认为auto，即自适应尺寸，如果设置为100px，即代表所有文本宽度小于100的其box尺寸均延至100, 大于100的其宽度根据文本宽度自适应。

以上三个属性可直接使用flex代替，如flex: 1; 代表flex-grow为１，其余为默认; 可以同时设置三个值，如flex: 0 0 100px;代表flex-grow为默认值０，flex-shrink设置为０，flex-basis设置为100px。