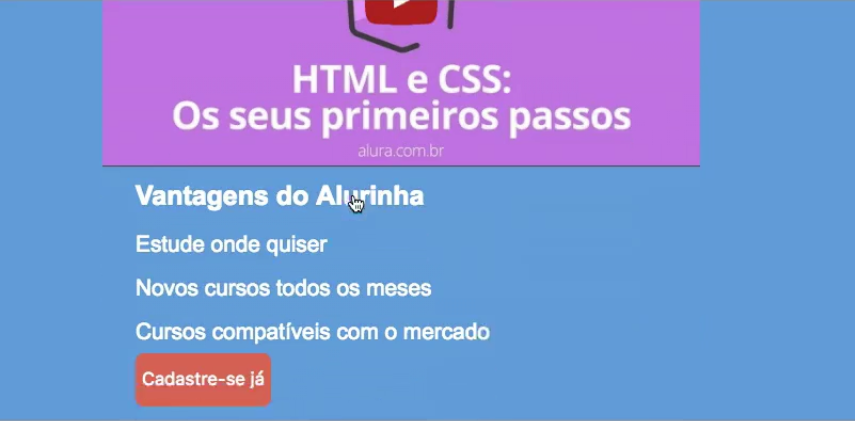
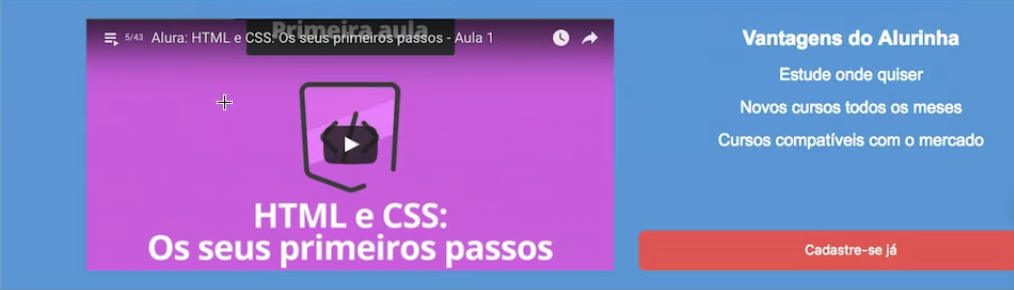
**FLEXBOX - Alura**

**Transcrição**

Finalmente vamos trabalhar em cima da parte de vídeos. Por enquanto está com a seguinte aparência:



Nosso objetivo é alcançar o resultado abaixo :



Observando o código descobrimos que um container envolve tanto o vídeo quanto o texto abaixo do vídeo. Assim, adicionamos .videoSobre e como queremos pegar o container vamos adicionar .videoSobre .container e ainda inserimos o display: flex. No arquivo flexbox.css teremos:

.videoSobre .container {

display: flex;

}

Dando um reload na página temos o seguinte:



Os elementos agora estão um do lado do outro! Repare que tanto a iframe quanto a div possuem a mesma altura justamente pelo comportamento do display-flex.

Falta centralizar os textos! Para alinhar é preciso encontrar qual é a div que engloba o texto. Assim, escrevemos videoSobre-sobre. Vamos escrever text-align: center:

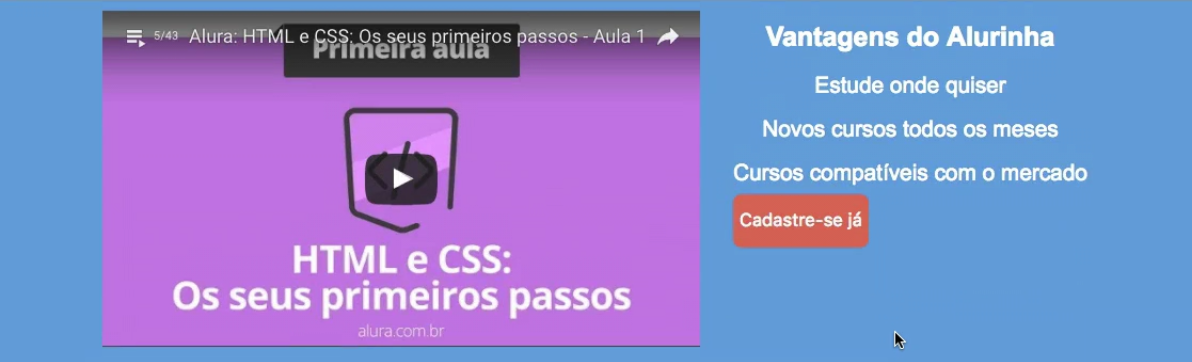
.videoSobre-sobre {

text-align: center;

display: flex;

}

Com isso teremos o texto centralizado! Mas, o botão estará a esquerda:



O que precisamos é adicionar um display:flex para alinhar os tamanhos dos objetos, mas ele por si só não é o suficiente, então, adicionamos flex-direction: column. Por fim, teremos os itens um em baixo dos outros. E o botão "Clique Aqui" fica com uma altura de 100%:

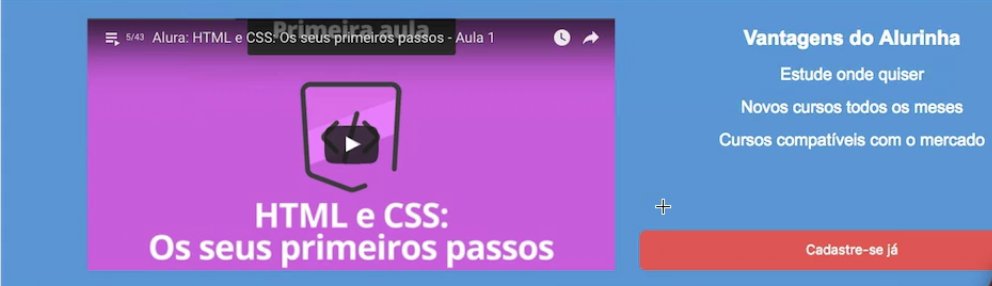
.videoSobre-sobre {

text-align: center;

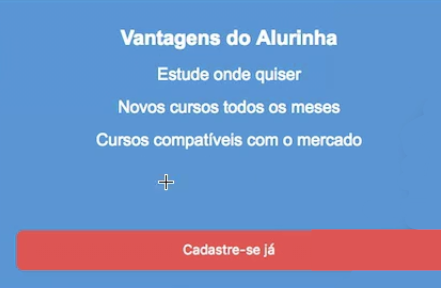
display: flex;

flex-direction: column;

}



Falta apenas lidar com o espaço que fica entre o texto e o botão, com o tamanho dos blocos que estão diferentes e também o espaço vazio a direita:



Para resolver a situação do espaço sem preenchimento vamos incorporar o que não está preenchido junto do Vantagens do Alurinha, é como se aumentássemos o tamanho disso. Assim, acrescentamos no .videoSobre-sobre adicionamos o flex-grow que diz ao flex-item que ele deve crescer, passamos também o 2 como medida:

.videoSobre-sobre {

text-align: center;

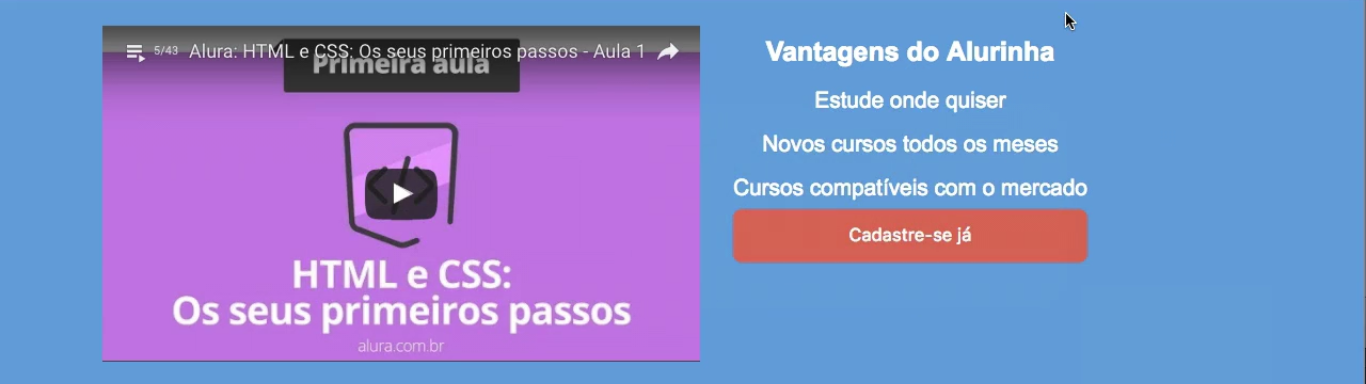
display: flex;

flex-direction: column;

flex-grow: 2;

}

Assim, temos o seguinte:



Ainda falta o botão estar melhor estruturado! Ou seja, é preciso espaçá-lo! Assim, vamos distribuir o espaço vazio que está de baixo do botão, portanto, vamos utilizar o flex-grow: 1. Assim, vamos adicionar o .videoSobre-sobre-list:

.videoSobre-sobre-list {

flex-grow: 1;

}

Assim, teremos o botão efetivamente mais embaixo!

Outra alternativa seria colocar o space-beetween, entretanto, o espaço seria distribuído entre os objetos e isso não é o que queremos! Para seguir fielmente o layout é melhor utilizar o flex grow.

Ainda falta um detalhe que é redimensionar o tamanho do vídeo e o que equivale a ele é o videoSobre-video e adicionamos para ele também o flex-grow: 1:

.videoSobre-video {

flex-grow: 1;

}

Assim, o tamanho do vídeo será distribuído de maneira a dividir o espaço com a parte de Vantagens. Se ao em vez de 1 tivéssemos colocado 2 o espaço total desta parte seria dividido em 3 cabendo ao vídeo 2/3 e ao Vantagens 1/2. Da mesma maneira podemos distribuir mais espaço para o Vantagens do que para o vídeo. Assim, colocando para o Vantagens um espaço maior, flex-grow: 3; teríamos 3/4 do espaço para o vantagens.

Nesse vídeo trabalhamos com flex-grow e sobre como ele divide o espaço vazio entre os flex-itens.