# **PLANO DE AULA**

Aula 04 | Tempo estimado: 1 hora e 30 minutos | Web Starter

Tipo da atividade: Offline

Ferramenta(s): Computador, Git, nodeJS e VS Code

Conteúdos

* CSS positions, flex, grid.

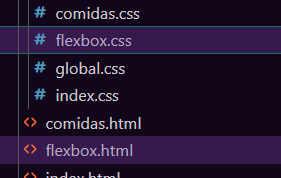
Objetivos

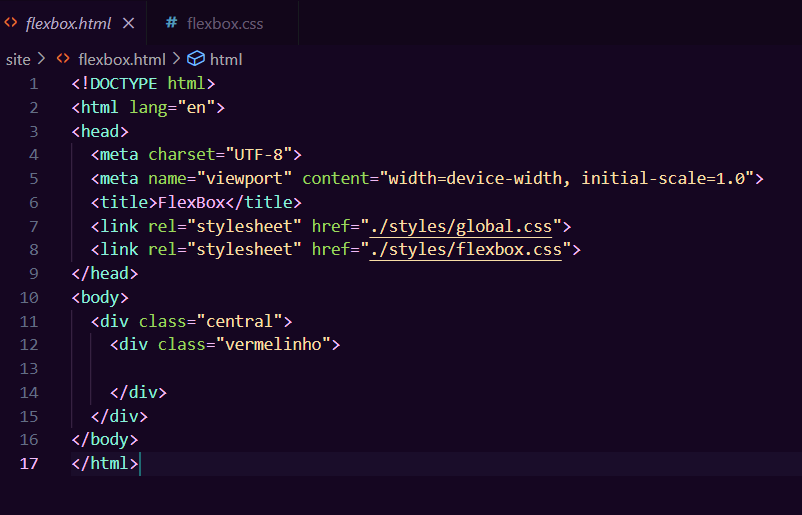
* Ensinar os alunos sobre DIVs e como posicioná-las, entendendo o que o código faz e como movimentá-la.

Estratégias e atividades

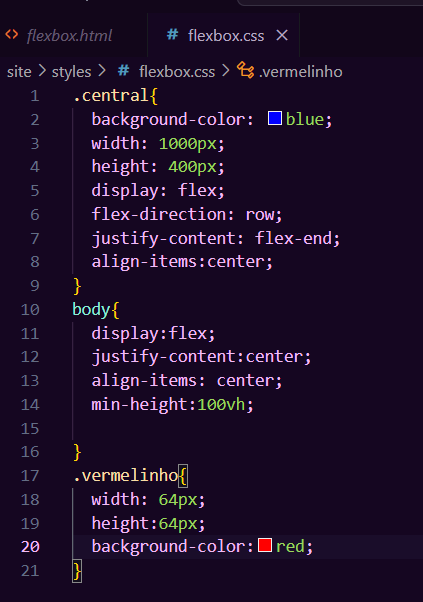
* Estilizar o site da aula passada e aprender sobre posições.
* Entender como funcionam as pastas.
* Aprender sobre novas tags HTML:
  + **<div> </div>**

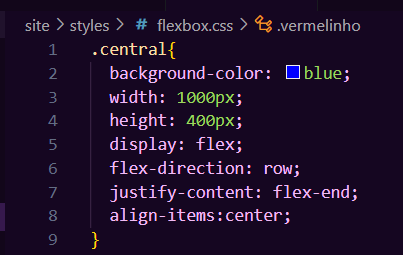
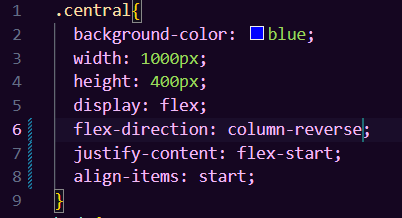
1. Hoje a aula é dividida em duas etapas, a primeira é explicação sobre como funciona o processo de DIVs, e a segunda é a parte prática quebrando a cabeça.
2. Para a aula de hoje serão implementados 2 códigos novos para poder entender melhor sobre como funciona o flexbox:

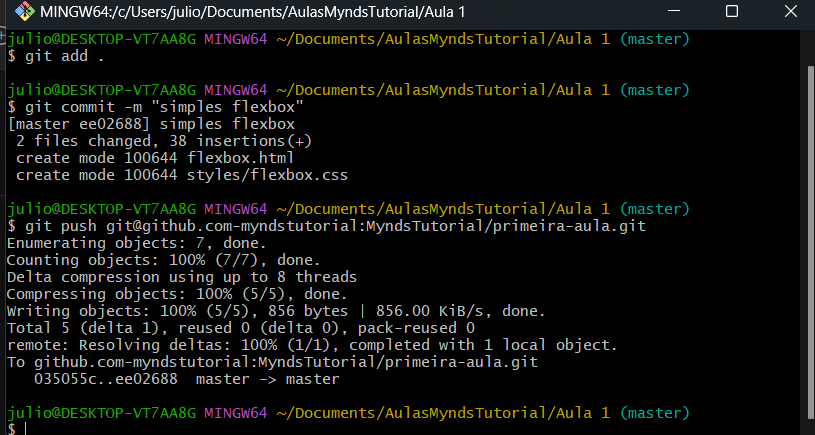


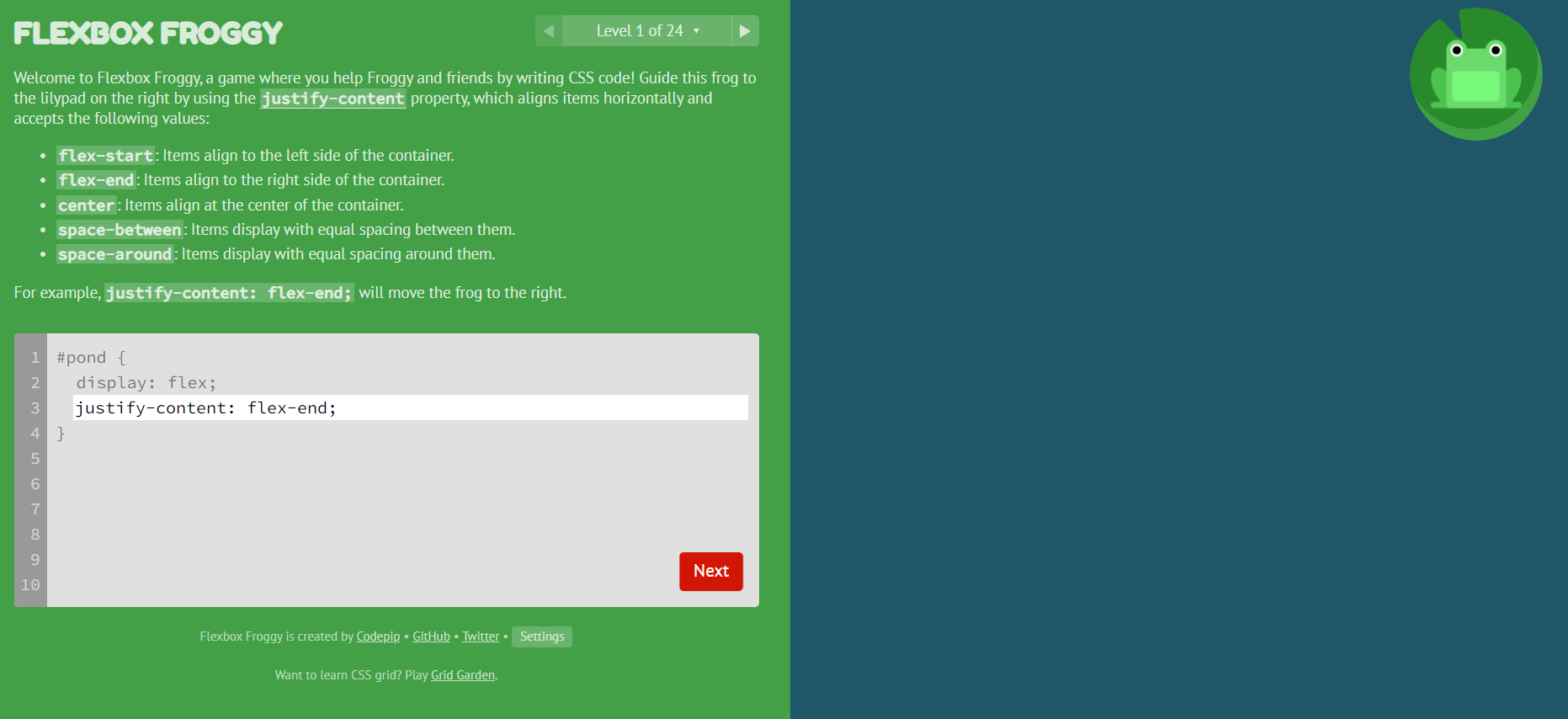
1. Vamos começar entendendo o HTML, que é a marcação do nosso site:  
     
   Repare que nele não temos muito código, então porque ele é importante? Porque a partir do momento que entendemos o que o CSS faz, conseguimos criar coisas muito lindas e funcionais, mas para chegar lá, precisamos entender sobre duas coisas, primeiro: **DIVs**, e segundo: **como posicioná-las**, afinal não queremos que isto ocorra:

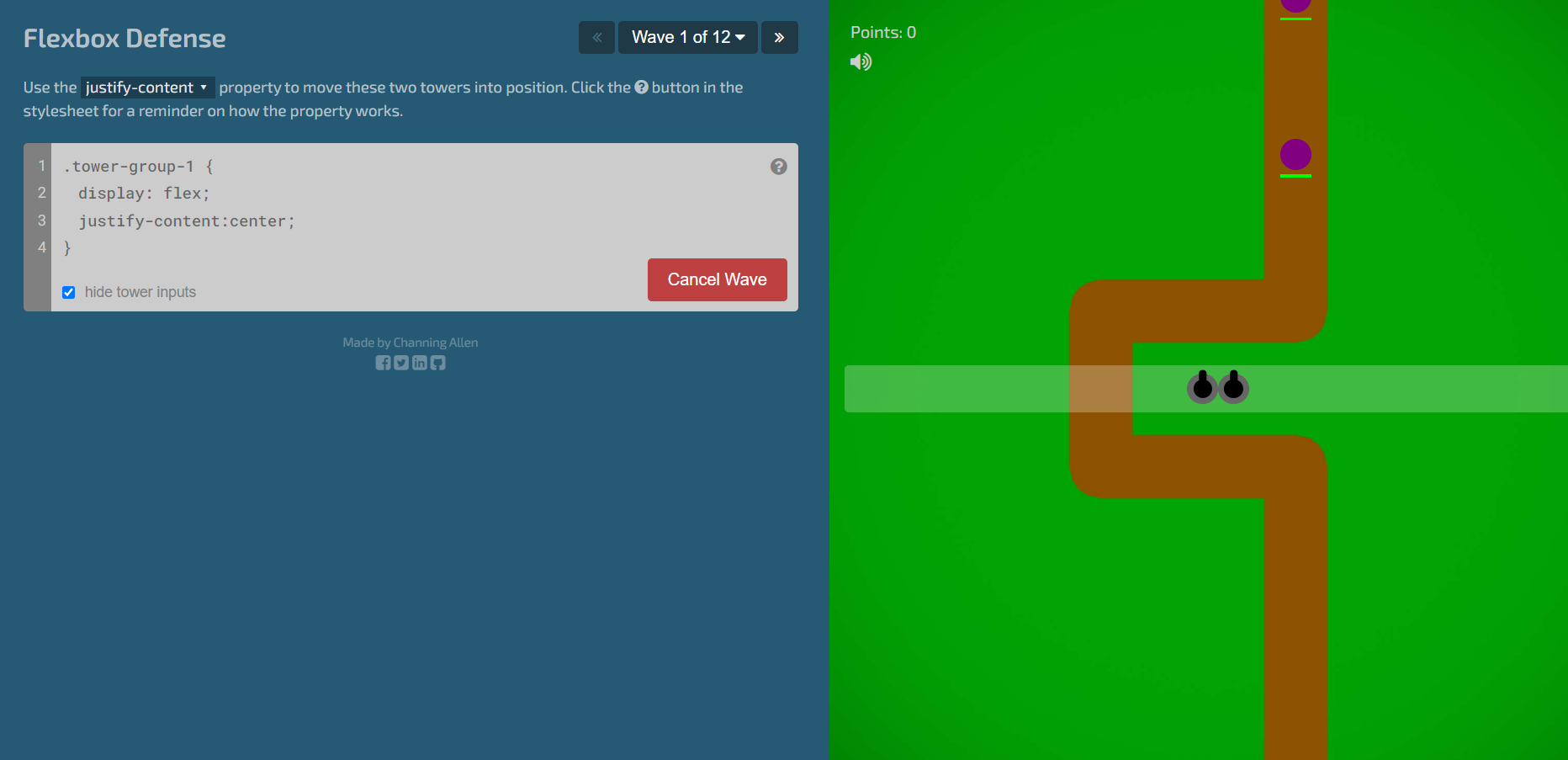
 \*Meme

1. Mas afinal o que é uma DIV?
   1. Imagine que você quer fazer uma cerca ao redor da sua casa, para juntar suas coisas. É exatamente isto que a DIV faz, ela nada mais é que um espaço para poder juntar todas as coisas do site (botões, imagens, etc) dentro de um mesmo grupo.
2. Okay, agora que entendemos o porquê da importância da DIV, por que temos uma dentro da outra? Vamos com o nosso querido CSS criar um espaçamento para poder entender como centralizar as coisas.
3. Se reparar bem, cada DIV tem uma classe - uma característica dela que pode passar para outras tags HTML, mas como está o CSS de ambas as DIVs?  
   
4. Vamos reparar neste CSS, tem algumas coisas interessantes:
   1. O body do site está com 100% da tela de min-height (altura mínima);
   2. O central tem a cor de fundo azul, width (largura) de 1000 pixels e a height (altura) de 400px;
   3. O vermelhinho tem apenas 64px de height e 64px de width, com a cor de fundo (adivinha só) vermelha;
   4. Mas agora o importante: por que eu não falei sobre o display?
5. O display é uma tag que faz com que a DIV se transforme em uma plataforma “flexível”, ou seja, o display flex transforma a DIV em uma parte onde se possa posicionar os itens.
6. Mas com isto, nós temos 2 coisas: justify-content e align-items. Esses 2 são, com certeza, um dos recursos mais utilizados por programadores, eles servem para posicionar as DIVs:  
   Mas olhe só o que acontece se eu alterar o justify-content e o align-items do central…

Antes:  
  
  
Depois:  
  
Mas e o resultado?

* 1. Agora que terminamos a parte do código, vamos subir para o Github e salvar a aula de hoje:  
     

1. Agora que já entendemos que são esses atributos que alteram a posição, o que fazer para aprender? Na programação só tem um jeito de se aprender: P R A T I C A N D O, e por isso, nós vamos jogar alguns jogos online de CSS.
   1. FlexBox Froggy: <https://flexboxfroggy.com/>   
      Ajude o sapo a ir para a vitória régia:  
      
   2. Grid Garden: <https://cssgridgarden.com/>  
      Ajude a regar as plantas com um quebra cabeças de CSS (grids):
   3. FlexBox Defence: <http://www.flexboxdefense.com/>

Destrua os inimigos com o flexbox:

* 1. FlexBox Zombies <https://mastery.games/flexboxzombies/>  
     Destrua os zumbis com CSS, este por ser o mais difícil, deve ser indicado por último:  
     

Recursos

* Git, Chrome e VS Code.

Observação

* Caso os alunos perguntem algo sobre como funciona tal coisa, ou quero estilizar tal coisa, SEMPRE, mesmo sabendo a resposta, é recomendado dizer: Não sei vamos pesquisar, e ensinar ele a pesquisar exemplo: “como centralizar uma DIV” e ir abrindo os stack overflow - é um site de perguntas e respostas para profissionais e entusiastas na área de programação de computadores. É extremamente importante que os alunos se sintam confiantes em jogar as dificuldades no Google para achar as soluções, principalmente porque a ideia de usar o Github é estimulá-los a continuar os projetos em casa.
* Dica: muito provavelmente, para os alunos, será um bom desafio, sempre ajude-os quando as fases começarem a ficar difíceis, mostre tentativas erradas na lousa e construa com eles a solução “fingindo resolver” com eles.

Tarefas

* Tentar zerar os jogos que não conseguir em aula, em casa.