



**DIO**  
CEZAR  
**GO**

# CSS Flex CSS Grid



# AGENDA

---

- o que são?
- css flexbox
- css grid
- qual devo usar?
- materiais complementares

# O QUE SÃO ESSES CONCEITOS?

- são basicamente implementações na linguagem CSS3;
- por que utilizar? → cansado de ficar ajustando tudo e não saber o que aconteceu como funciona?
  - problema → float;
- aplicam novos conceitos de formatação do seu html e css;
- já estão preparados para responsividade;
- aqui estão apresentados conceitos superficiais das tecnologias, e materiais para aprofundamento;

# CSS FLEXBOX

# O QUE É CSS FLEXBOX?

- faz parte da especificação do CSS3 que promete organizar elementos na página previsivelmente quando o *layout* precisa ser visualizado em diversos tamanhos de tela e em diversos dispositivos;
- nos ajuda a organizar elementos sem a ajuda do float;
- também nos ajudam a resolver problemas de *box model* → quando acrescentamos, *padding*, *margin* e *border* além da largura do elemento.

# COMO FUNCIONA CSS FLEXBOX?

---

- a ideia é simples: os filhos de um elemento com flexbox pode se posicionar em qualquer direção e pode ter dimensões flexíveis para se adaptar;
- você pode posicionar os diversos elementos independente da sua posição na estrutura do HTML;
- um dos problemas do float é a sua dependência com os elementos na estrutura do HTML;
  - estes elementos precisam estar em uma ordem específica, se não o layout não dá certo.
- com o Flexbox essa ordem não importa, isso quer dizer que você pode organizar a informação do seu HTML de beneficiando o SEO e Acessibilidade

# ELEMENTOS E VOCABULÁRIO

---

- Flexbox é uma nova maneira de posicionar elementos do seu *layout* e por isso precisamos de novos nomes para identificar os elementos da estrutura;
- Flex container → é o elemento que envolve sua estrutura. define-se que um elemento é um Flex Container com a propriedade `display` com os valores `flex` ou `inline-flex`;
- Flex Item → são os elementos filho do flex container;
- Eixos ou Axes → são as duas direções básicas que existem em um Flex Container: *main axis*, que seria o eixo horizontal ou o eixo principal e o *cross axis* que seria o eixo vertical.

# ELEMENTOS E VOCABULÁRIO

---

- Directions → determina a origem e o término do fluxo dos ítems. Eles seguem o vetor estabelecido pelo modo tradicional de escrita: esquerda para direita, direita para esquerda etc;
- a propriedade order determina o lugar que os elementos aparecerão;
- a propriedade flex-flow determina a ordem do fluxo em que os elementos aparecerão;
- você pode definir se os elementos serão forçados a ficar em uma mesma linha ou se eles irão quebrar em várias linhas com a propriedade flex-wrap;



# GUIA DEFINITIVO

---

para aprender e realmente entender... só praticando

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

# EXEMPLOS E ESTUDOS

---

alguns estudos sobre flexbox

<https://github.com/diogocezar/dctb-flex>

# APRENDA JOGANDO

---

guie um sapinho com comandos flexbox e aprenda se divertindo

<http://flexboxfroggy.com/>

# CSS GRID

# O QUE É CSS GRID?

- faz parte da especificação do CSS3 que promete organizar elementos na página previsivelmente quando o *layout* precisa ser visualizado em diversos tamanhos de tela e em diversos dispositivos;
- nos ajuda a organizar elementos sem a ajuda do float;
- também nos ajudam a resolver problemas de box model → quando acrescentamos, padding, margin e border além da largura do elemento.

# O QUE É CSS GRID?

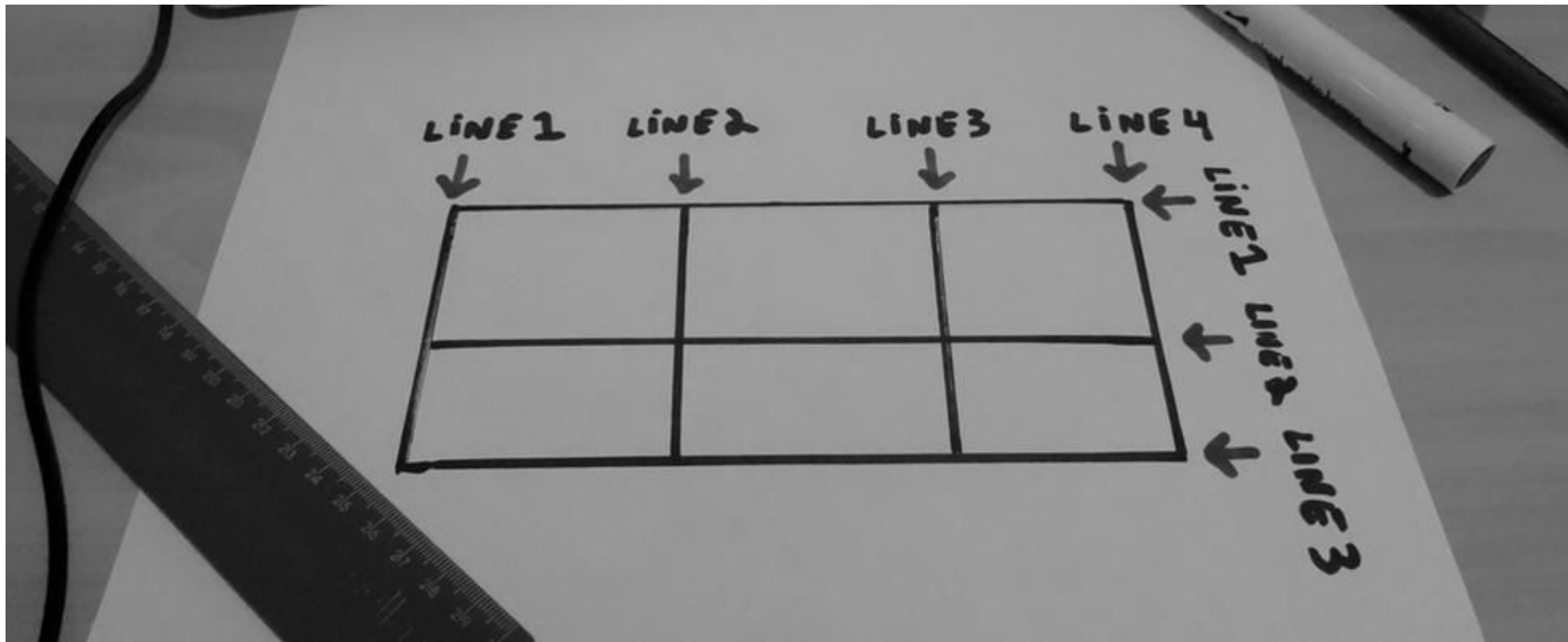
- utiliza um conceito de 2 dimensões, enquanto o flexbox utiliza conceitos de 1 dimensão;
- já se pode utilizar?
  - <http://caniuse.com/#feat=css-grid>

# TERMINOLOGIA

---

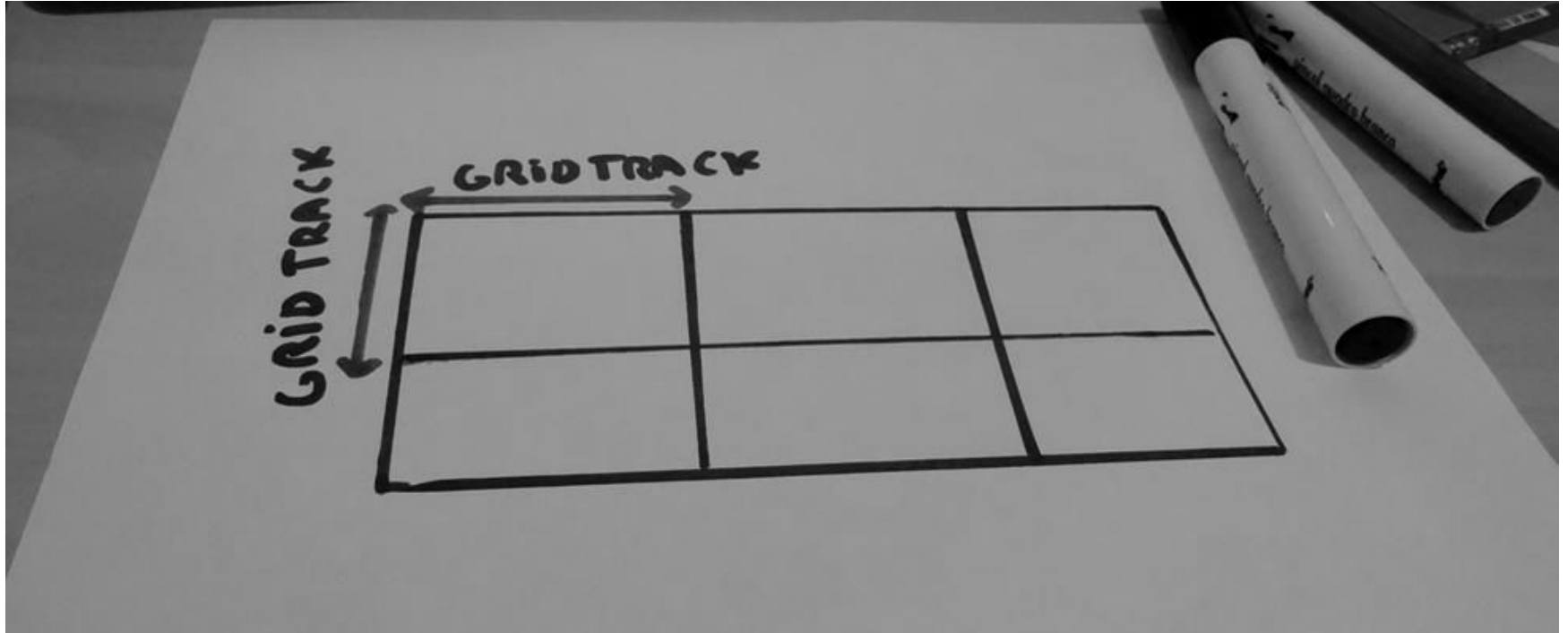
- vamos definir alguns termos utilizados com grids:
  - grid lines → são as linhas que definem o grid, elas podem ser distribuídas de forma horizontal ou vertical;
  - grid tracks → é o espaço horizontal ou vertical entre duas grid lines;
  - grid cell ou grid item → é o espaço entre quatro Grid Lines, sendo a menor unidade em nosso grid, conceitualmente podemos fazer uma analogia com uma célula de tabela;
  - grid áreas → é qualquer espaço no Grid usado para exibir um ou mais Grid Cells/Items.

# GRID LINE

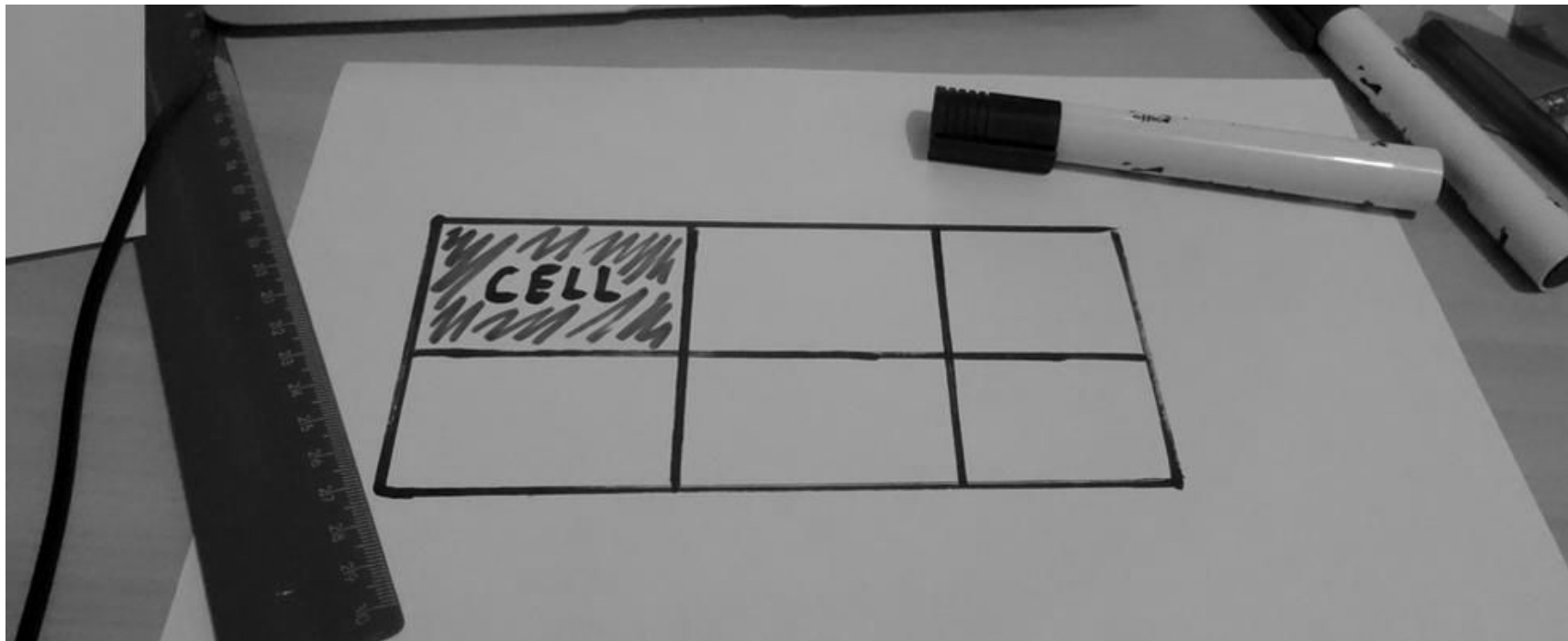




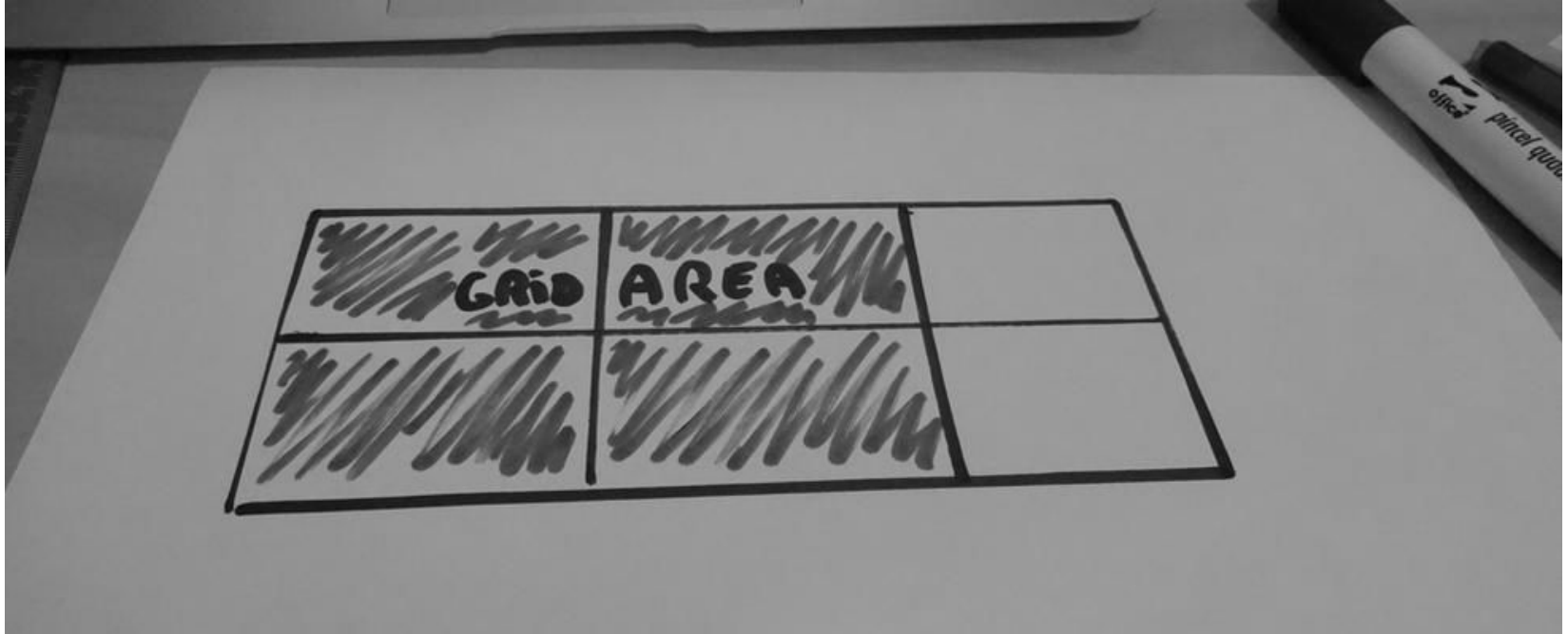
# GRID TRACKS



# GRID CELL



# GRID ÁREAS



# TERMINOLOGIA

---

- vamos definir alguns termos utilizados com grids:
  - grid containers → é o pai direto de todos os itens do grid sendo o elemento que recebe a propriedade display: grid;
  - grid items → são os itens que representam o conteúdo do grid, cada filho direto do grid container torna-se um grid item;
  - para definir as colunas e linhas do grid use as propriedades grid-template-columns e grid-template-rows.
  - use grid-column-start e grid-row-start para especificar a grid line onde o item começa e grid-column-end e grid-row-end para especificar a grid line onde o item termina;

# TERMINOLOGIA

---

- vamos definir alguns termos utilizados com grids:
  - area naming → No grid container podemos manipular a posição e comportamento dos grid itens, isso através da propriedade grid-template-areas, onde podemos literalmente montar todo o nosso layout em apenas uma propriedade;
  - para facilitar e atribuir mais semântica ao que estamos fazendo, atribuímos um nome a nossos grid itens através da já conhecida grid-area.

# EXEMPLOS E ESTUDOS

---

alguns estudos sobre grid

<https://github.com/diogocezar/dctb-grid>

# GUIA DEFINITIVO

---

para aprender e realmente entender... só praticando

<https://css-tricks.com/snippets/css/complete-guide-grid/>

# APRENDA JOGANDO

---

aprenda se divertindo no grid garden

<http://cssgridgarden.com/>



**QUAL DEVO USAR?**

# QUAL DEVO UTILIZAR?

- flexbox → é para layouts unidimensionais; ou seja, qualquer coisa que precise ser disposta em uma linha reta;
- grid → é a solução certa quando você deseja controlar o dimensionamento e o alinhamento em suas dimensões;

# **MATERIAIS COMPLEMENTARES**

---

- <https://goo.gl/hVh29F>
- <https://goo.gl/65YKWC>