

HTML E CSS

ESTRUTURA

Hyper Text Markup Language

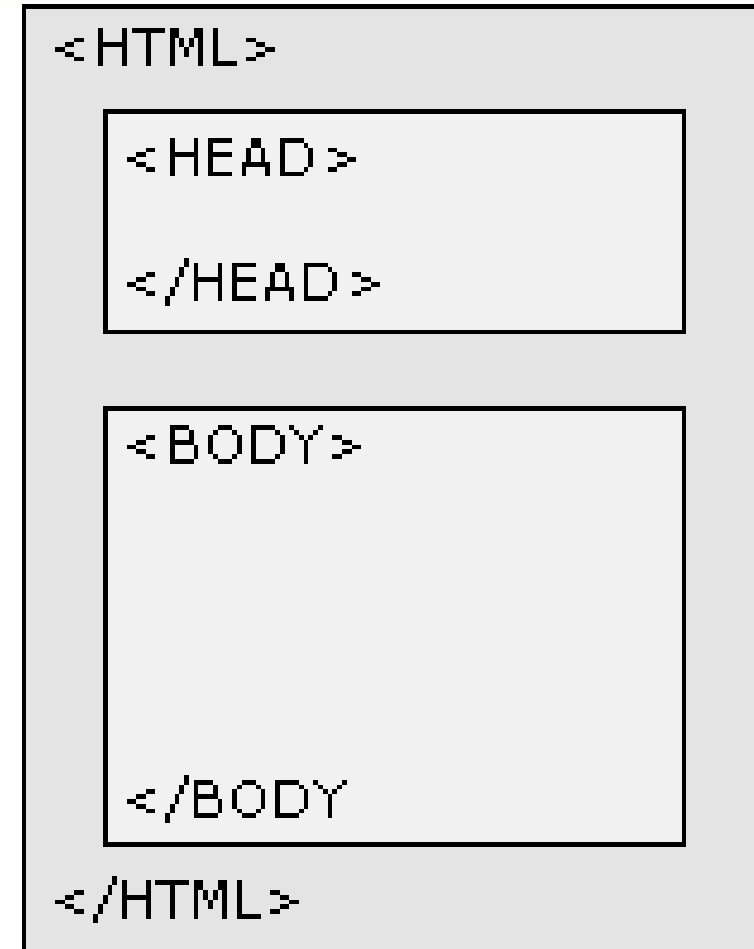
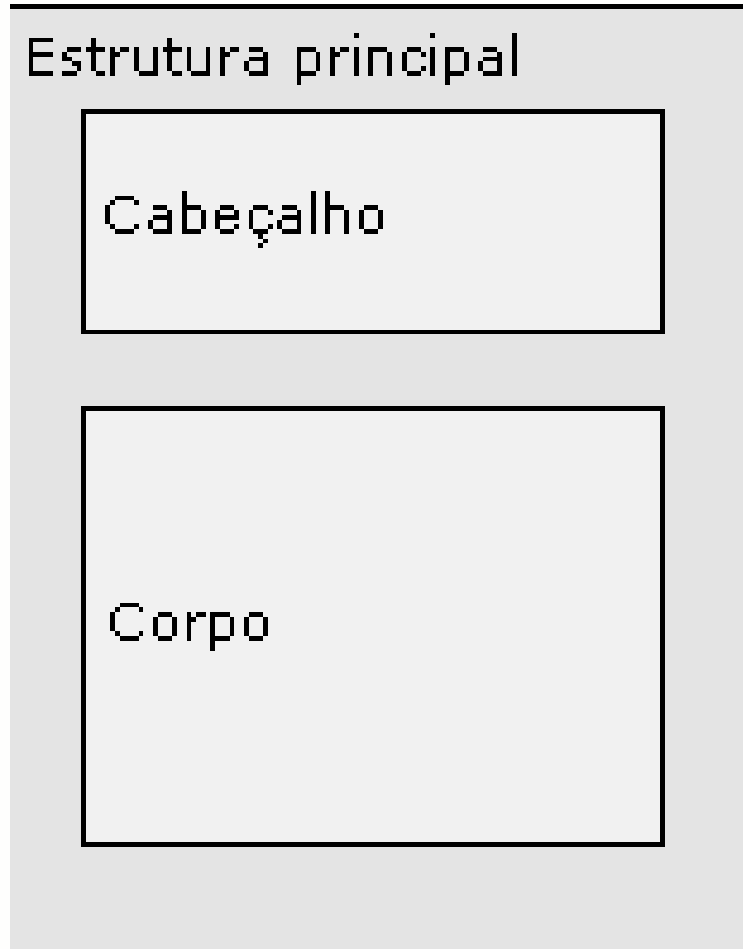
Basicamente separado em HTML e aqui dentro teremos acesso as tags e "comandos" do HTML

Depois no HEAD, onde temos as informações de cabeçalho, e nesse caso entende-se ABA, e os meta dados, dados de indexação para buscas e inclusões de arquivos extras.

E o tal do BODY, onde teremos as tags e textos que serão exibidos, e coisas mais que eventualmente podem aparecer.

AAAAAAAH TEM UM DETALHE!

<!DOCTYPE html> é uma tag necessária tá!





— HEAD

Tags:

<title>

Informa o título da página, hoje fica somente na aba.

<link>

Utilizado para trazer para o arquivo alguma informação externa a página/arquivo, isso pro CSS

<script>

Também utilizado para trazer informações externas, nesse caso para os arquivos .js, porém isso quando usamos o src, mas da também pra escrever o JS direto na página, abrindo e fechando a tag <script></script>

<meta>

Utilizado para os metadados, sendo algumas propriedades charset, name e etc...



BODY

Vamos falar das categorias, em virtude das inúmeras tags que existem.

Estruturais: Que movem a estrutura da página, tipo div e nav...

Textuais: Estão envolto ao conteúdo de fato, aos textos possíveis, tipo h1...6, p, a e afins...

Multimídia: Ligados à elementos de mídia, tipo img, audio, video...


Formulário: Vinculadas à criação dos formulários, como inputs, form, submit e afins...

Listas: Para criação de listagens, tipo ol, ul, li e só...

Ah e como "comento" no HTML <!-- -->

PASSANDO RAPIDINHO PELAS PRINCIPAIS TAGS

ESTRUTURAIS



DIV	NAV	SECTION	FOOTER / HEADER
Esse é o cara mais genérico que existe, por natureza não tem nada, é só um container em branco.	Aqui já estamos falando em seções na página, como se a gente dividisse ela em espaços nomeados.	Sim, pra deixar as coisas um pouco mais genéricas, aí temos o section.	Sim, temos os elementos que constituem o clássico cabeçalho e rodapé.
Uma div por si só, não é nada, motrarei-lhes.	Seu propósito é o uso em áreas onde serão utilizados links de navegação.	Section não tem nada mesmo, sua finalidade é apenas criar seções CURTAS.	Não necessariamente vai estar fixo, porém são mais dois elementos estruturais com finalidades específicas.
div tem os "parâmetros" de qualquer tag, id, class e align.	Sendo bem sincero, no final é a mesma coisa.	Aí é tipo áreas de texto, seções da página e afins...	





PASSANDO RAPIDINHO PELAS PRINCIPAIS TAGS

TEXTUAIS

H1...6	P / I / U / B	A	SPAN / LABEL
Do h1 até o h6 temos os 6 níveis/tamanhos de títulos possíveis.	Todas tags para texto, dando um estilo diferente para o que estiver em seu contexto.	Links, e somente links.	Esses sim são bons. Quando não quer ter o estilo do paragrafo, vem só com uma frase, então é span ou label.
Sendo 1 o maior e 6 o menor.	p é só para iniciar o paragrafo mesmo, então terá quebra de linha.	Antes se usava para tudo que queria colocar o cursor pointer (mãozinha).	Span é diferenciado porque ele fica isento dos estilos utilizados ao redor, permitindo adicionar novos ou simplesmente remover.
Sua função é só essa, e por favor não usem font-size nessa tag.	i para o ítilico u para sublinhado b para negrito, mas também tem o strong, talvez até mais recomendado.	seguido do href, pois o a sozinho não faz nada, cria o hyperlink para outras páginas e afins.	Já o label é mais pra formulários, quando se tem interação com usuário

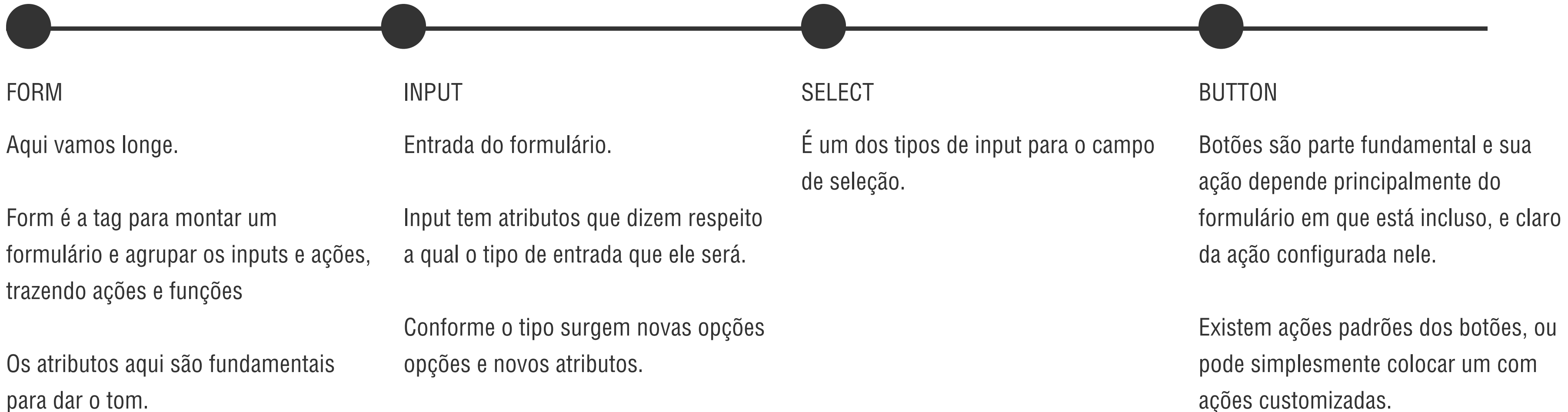
PASSANDO RAPIDINHO PELAS PRINCIPAIS TAGS

MULTIMÍDIA

			
IMG	AUDIO	VIDEO	IFRAME
Basicamente, traz uma imagem pra tela.	Áudio, pronto.	Vídeos e áudios, porém áudio é audio.	É um frame de alguma outra página dentro da tua página.
Tem 2 atributos básicos.	Para reproduzir um ou mais áudios com uma possibilidade grande de possibilidades.	Vídeos também tem as mesmas coisas que o áudio, grandes possibilidades e poucas extensões.	Possibilidade de trazer partes de um conteúdo pra cá.
src para colocar o source da imagem, pode ser o base64 dela, o link ou mesmo o caminho.	Verificar sempre a extensão do arquivo, pois são poucos os válidos. Também varia bastante conforme o navegador.		É muito utilizado, por trazer muitas possibilidades.
alt para adicionar descrição, e isso é útil pra acessibilidade.			

PASSANDO RAPIDINHO PELAS PRINCIPAIS TAGS

FORMULÁRIO



PASSANDO RAPIDINHO PELAS PRINCIPAIS TAGS

LISTAS

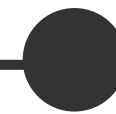


TABLE

Para criação das tabelas, que talvez é a tag mais útil.

Table abre um lack de oportunidades, tipo para colocar um título na tabela, posso usar um atributo próprio.

E sim, para criar linhas e colunas uma tag específica.

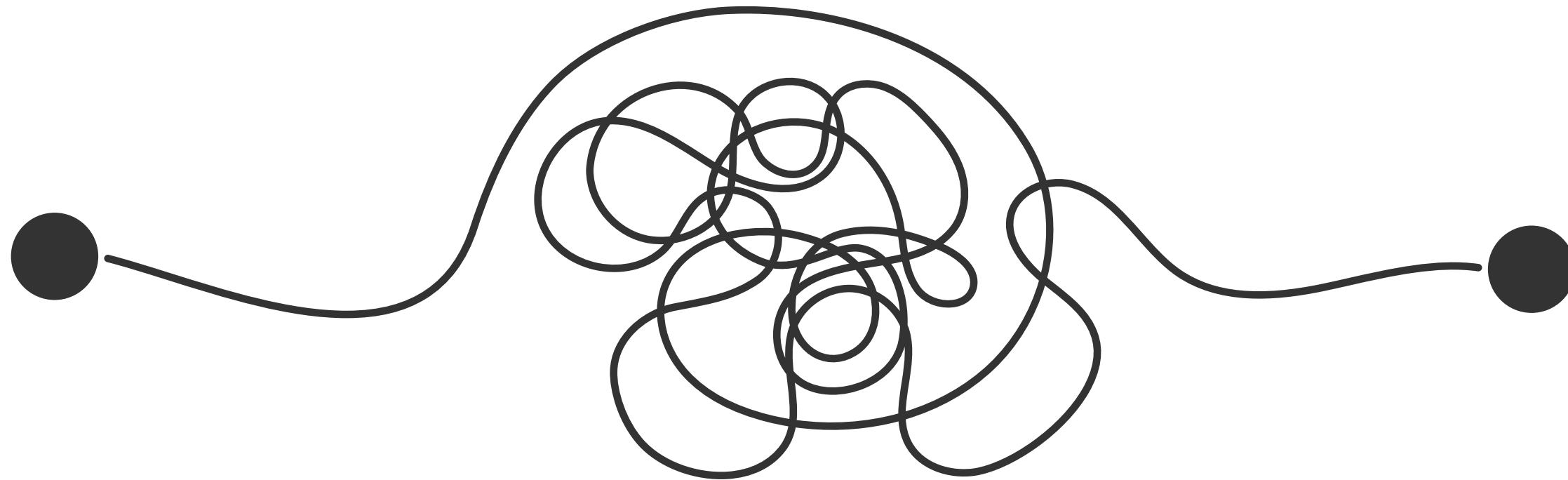


UL

Para listas organizadas e ordenadas.

Possivelmente você nunca vai saber qual é o primeiro, UL LI OL.

ul ou ol para lista com número ou com o ponto, e o li que vai adicionando nova linha.



— DESAFIO HTML

1. Crie um arquivo HTML chamado "profile.html".
2. Dentro do arquivo HTML, adicione a estrutura básica de uma página HTML, incluindo a declaração do tipo de documento, a tag `<html>`, `<head>` e `<body>`.
3. No `<head>`, pelo menos a tag `<title>` e defina um título para a sua página, se preferir coloque aquelas imagens.
4. No body, utilize tags semânticas apropriadas, como `<header>`, `<main>`, `<section>` e `<footer>`, para organizar o conteúdo da página.
5. Pense nas sessões e as tags adequadas para tal.
6. Adicione links para suas redes sociais, como Facebook, Twitter ou LinkedIn.
7. Utilize pelo menos uma tabela para exibir informações tabulares, como seu histórico educacional ou experiência profissional.
8. Faça uso de pelo menos uma imagem em sua página.
9. Adicione um rodapé à sua página com informações adicionais, como o ano atual ou uma mensagem de direitos autorais.
10. Caso esteja a vontade, use cores e o que mais souber. Mas só HTML.



CSS

Cascade Style Sheets

Vamos ao monstro!

Tem uma estrutura que parece/lembra um arquivo JSON, temos o seletor, aí abrimos chaves {} e colocamos os atributos e seus respectivos valores.

Linguagem de estilos para tratar arquivos HTML ou XML, dá os atributos dos elementos, deixando-os no grau. É na real o que realmente faz a diferença na página, o embelezar!

Para comentar usamos `/**/`.

Aqui a coisa começa a complicar para quem não tem habilidades de design ou UX/UI.

Mas ferramentas diversas e gente que só pensa nisso ajuda e muito.

```

h1 {
  color: red;
  font-size: 20em;
}

.default-button {
  border: 1px solid black;
  border-radius: 2%;
  background-color: blue !important;
}

#apresentation-photo {
  width: 200px;
  height: auto;
  border-radius: 50%;
  margin-bottom: 10px;
}

.links ul li {
  cursor: pointer;
}

.links ul li:hover {
  color: yellow;
}

```

ELEMENTOS CSS

Sobre a estrutura vamos primeiro aos seletores, ou seja, como selecionamos quais elementos vão receber os atributos declarados

Elementos HTML: Diretamente elementos do HTML

Classes: Classe é um atributo do HTML que pode ser utilizado múltiplas vezes. No CSS declaramos com o . (ponto).


ID: Também um atributo do HTML, que indica o identificador único do elemento, no CSS declaramos com o # (hastag).

Descendência e Filho: Da pra navegar entre os elementos filhos, respeitando a hierarquia do HTML podemos navegar entre eles e selecionar elementos abaixo, atribuindo à eles, e não há árvore.

Eventos: Temos um esquema de eventos ou ações no CSS também, onde dado um evento é aplicado o estilo descrito.

PASSANDO RAPIDINHO PELOS PRINCIPAIS ATRIBUTOS

ESTRUTURAIS



MARGIN	PADDING	BORDER	BOX-SIZING
Adiciona margem da borda do elemento pra fora.	Adiciona preenchimento da borda do elemento pra dentro.	Adiciona uma linha/borda no limite do elemento.	Diz respeito ao tamanho da caixa que está adicionando.
Tem -top, -bottom, -left, -right todas as direções.	Tem -top, -bottom, -left, -right todas as direções.	Tem -top, -bottom, -left, -right todas as direções.	É um elemento bem importante atualmente, quando se fala em trabalhar com o HTML quase que como uma tabela.
Aceita valor negativo. Aceita os valores em pixel, que é mais comum, e também os valores dinâmicos (em, %)	Aceita valor negativo. Aceita os valores em pixel, que é mais comum, e também os valores dinâmicos (em, %)	border: espessura estilo cor	Definimos o padrão utilizando o valor border-box .
Bem dinâmico.	Bem dinâmico.	Tem vários atributos relacionados, que dão suporte e trazem novas possibilidades ao estilo.	

PASSANDO RAPIDINHO PELOS PRINCIPAIS ATRIBUTOS

ESTRUTURAIS



BACKGROUND

Tem vários atributos.

Utilizando só o background é possível colocar imagens de fundo e alterar seus atributos.

Comumente temos:

- attachment
- clip
- color
- image
- origin
- position
- repeat
- size

WIDTH/HEIGHT

São os tamanhos, dizem respeito, respectivamente, a largura e altura.

tem os prefixo min- e max- para definir a escala de tamanho dos objetos

FONT

Trata tudo sobre os textos.

Também possuí vários atributos complementares que dizem respeito ao estilo do texto.

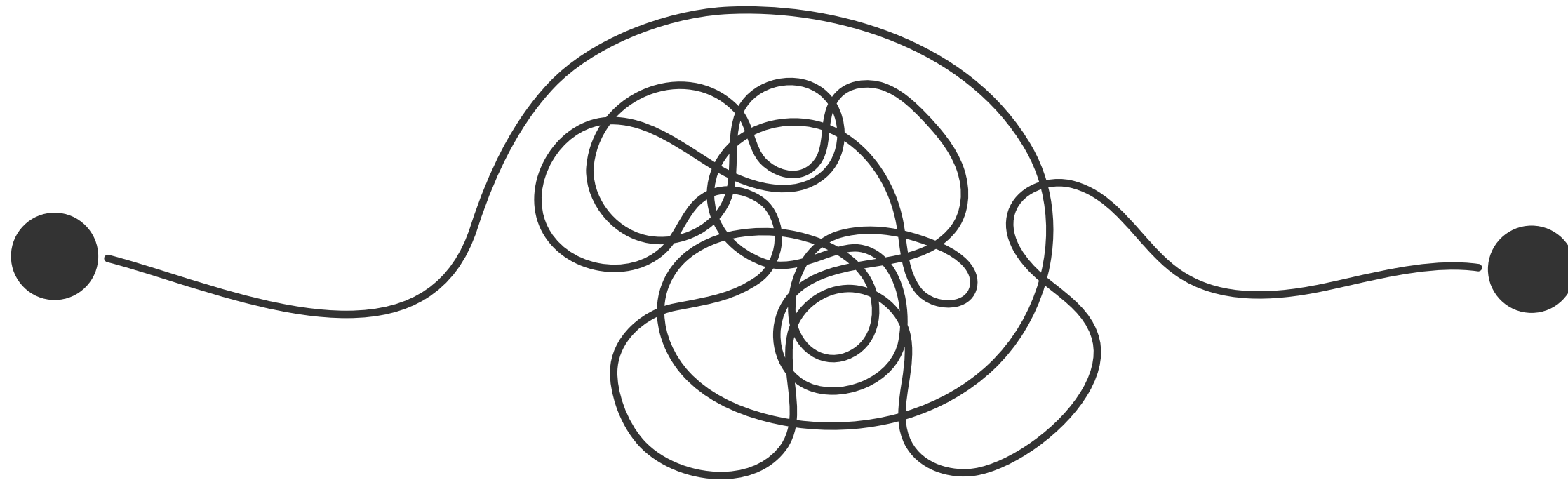
Temos:

- family
- size
- stretch
- style
- height
- weight

DISPLAY

Diz respeito sobre como os elementos se comportam na apresentação do todo.

Conforme os valores estipulados e atribuídos, os blocos de conteúdo serão organizados novamente.



— DESAFIO CSS

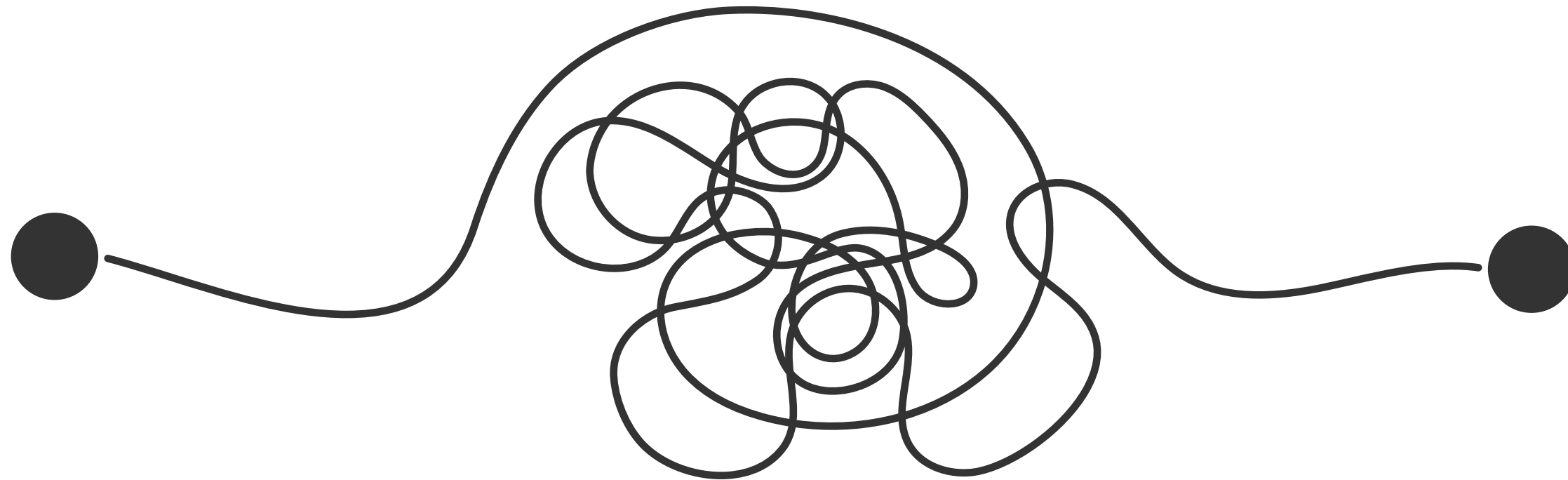
Existem 3 exercícios pra gente ir pegando o jeito e quebrando o gelo com o CSS, coisa mais simples.

A ideia é resolvê-los, dá-se um tempo para tal, depois resolvemos juntos.

PASSANDO RAPIDINHO PELOS PRINCIPAIS ATRIBUTOS

EVENTOS AÇÕES

HOVER	FOCUS	ACTIVE	AFTER / BEFORE
<p>houver é o evento disparado ao passar o cursor do mouse por cima da determinada área ou elemento.</p> <p>Dentro do atributo CSS, sempre indicamos como:</p> <pre>ELEMENTO:hover {}</pre> <p>E dá-se os atributos de quando o mouse passar por ali.</p>	<p>Esse evento é mais voltado para inputs ou áreas de seleção.</p> <p>Focus significa que, o evento será disparado quando um elemento tiver foco, em outras palavras, quando você estiver com o campo selecionado.</p>	<p>O mais clássico e talvez simples de entender.</p> <p>O evento active é disparado ao clicar em um elemento. Mas aqui existe várias ressalvas que é preciso se ater, por exemplo, um elemento pode estar atrás de outro, impedindo o click.</p>	<p>Esse é o mais da ora.</p> <p>Ambos significam a criação de novos elementos em página. Na real, pseudo elementos. É como se "criássemos" uma nova área no HTML e, conforme a ação utilizada, adiciona antes (before) ou after (depois) do elemento indicado no estilo.</p>



— DESAFIO CSS

Vamos voltar no último exercício e nele adicionar mais esses detalhes de eventos. Se estiver sem criatividade dou ideias:

1. Coisa simples dessa vez, tipo passa o mouse por cima, muda a cor do fundo.
2. Clicou em um elemento, faz ele se mover para algum lugar da tela, ou simplesmente vai pra lá e vem pra cá.
3. Saiu do elemento, adiciona algum texto, tipo "volte aqui", nesse sentido.

Tudo comitado lá no seu repositório e vamos ver alguns aleatoriamente.

Inline CSS

```
<p style="color: blue;">This is a paragraph.</p>
```

Internal CSS

```
<head>  
  <style type = text/css>  
    body {background-color: blue;}  
    p { color: yellow;}  
  </style>  
</head>
```

External CSS

```
<head>  
  <link rel="stylesheet" type="text/css" href="style.css">  
</head>
```

CSS E ONDE COLOCAR

Existe formas de se escrever o CSS.

Ou melhor, onde colocar o CSS.

Inline, ou "na linha" é a forma de escrevermos o CSS diretamente na tag HTML, para tanto, adicionamos o atributo style e escrevemos o o CSS na mesma linha, com a mesma sintaxe já clássica.

Internal o CSS está no mesmo arquivo, porém dentro da tag head, área onde, o HTML ou o que tiver lá não é exibido, colocamos a tag style, que indica o inicio da estilização, e então escrevemos normalmente o CSS lá dentro.

E o external, que é o clássico arquivo .css. Basta colocar a tag de link dentro do arquivo HTML para fazer a "chamada" do estilo.

MEDIA QUERIES

MEDIA QUERY

O tal do @media.

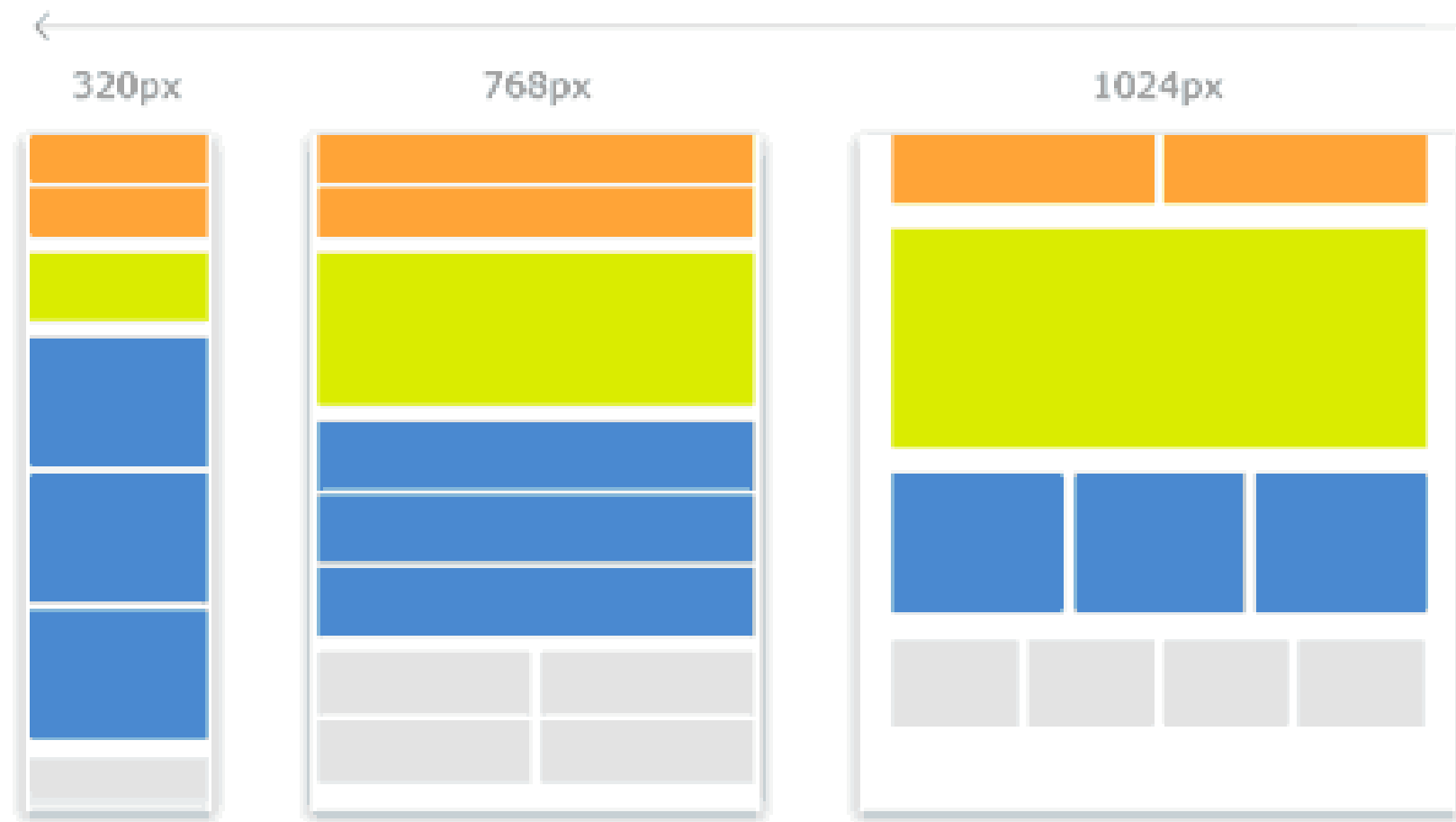
@media é a função do css que trata uma espécie de condicional para a aplicação do estilo, conforme regra.

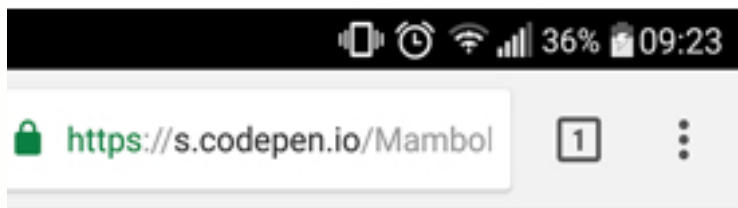
Ou seja, é possível fazer com que elementos tenham estilos completamente diferentes ou aplicamos novos eventos e assim por diante, conforme o tamanho máximo da tela, por exemplo.

Não só pra tamanho de tela, mas para orientação da imagem, tipo de dispositivo, limitação do meio, enfim.

Basta:

```
@media (condicional) {  
    elemento { atributos }  
}
```





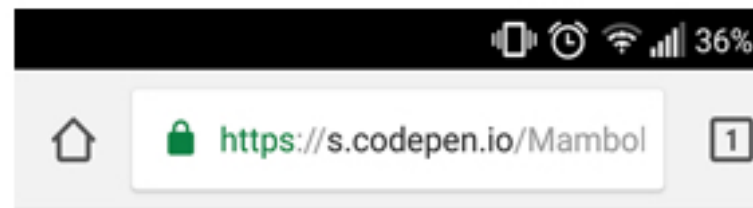
20%

40%

60%

80%

100%



20%

40%

60%

80%

100%



VIEWPORT

Aqui é algo um pouco mais voltado a escala e tange mais técnicas. Viewport é a área gráfica da tela, disponível pra exibição da nossa aplicação.

Mas também vamos tratar acessibilidade e design em escalas menores do que uma tela de computador com maior resolução.

Temos por exemplo o uso de uma nova escala para tratarmos o tamanho completo da tela, ao invés da escala 100%:

width: 100vw

height: 100vh

Também podemos utilizar a tag meta para trazer alguns recursos:

<meta name="viewport" content="width=device-width">