



Gabriel Michalski
Frontend Developer

a.C

d.C



a.C

d.C

1835

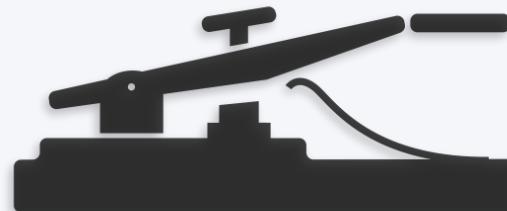


d.C

1835

1945

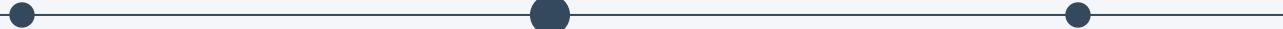
Samuel Morse
cria o telégrafo



d.C

1835

1945



Código Morse



1835

1945

1959

Eniac

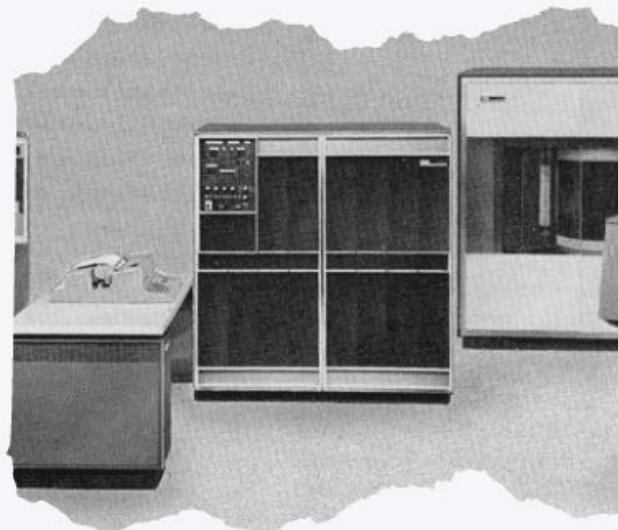


1945

1959

1969

IBM 1401



1959

1969

1969



1959

1969

1972



1969

1972

1973

Ray Tomlinson



1972

1973

1977

Robert Kahn



1972

1973

1977

Robert Kahn

Transfer
Control
Protocol

1972

1973

1977

Robert Kahn

INTERNETTING

1972

1973

1977

Vint Cerf



1972

1973

1979

Vint Cerf

Internetwork
Protocol

1972

1973

1977

Robert Kahn

Vint Cerf

TCP/IP

1973

1977

2015



1973

1977

2015



1977

2015

Hoje





Hoje

1980

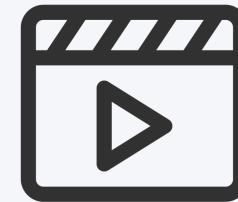
1990

Tim Berners-Lee



1980

1990



HTML

(Hypertext Markup Language)



```
<html lang="pt-br">  
  <head>  
    <meta charset="utf-8">  
    <title>Título da página</title>  
  </head>  
  <body>  
    Conteúdo da página  
  </body>  
</html>
```

HTML



- Primeira versão do HTML;
- Desenvolvido pela equipe de Tim Berners-Lee;
- Derivado do SGML (Standard Generalized Markup Language);
- Extremamente simples.

HTML 2.0



- Grupo HTML Working Group;
- Empresas que criavam os navegadores começaram a auxiliar na criação da linguagem;
- Cada navegador queria implementar as suas próprias tags;
- Criado o W3C.

HTML 4



- Muitas implementações novas;
- É compatível até hoje;
- Grande quantidade de adeptos.

XHTML

(eXtensible Hypertext Markup Language)



- Mozilla, Ópera e Apple criam um grupo chamado **WHATWG**;
- A tecnologia muda bastante em relação ao HTML.

HTML 5



- Utiliza um banco de dados da Web, enquanto as outras versões utilizam o cache do navegador
- Não se baseia mais no SGML, por isso pode ser aprimorado e aperfeiçoado de forma independente
- Suporte para novos tipos de controles de formulários, como data e horas, e-mail, pesquisa, etc.

Tags Estruturais

- `<!-- -->` - Cria um comentário;
- `<html> </html>` - Envolve todo um documento html;
- `<head> </head>` - Envolve o cabeçalho de um documento html;
- `<meta >` - Fornece informações gerais sobre o documento;
- `<title> </title>` - O título do documento;
- `<body> </body>` - Envolve o corpo (texto e tags) do documento html;
- `<style> </style>` - Informações de estilo;
- `<script> </script>` - Linguagem script.

Exercicio

1. Criar uma estrutura básica para o html.

Tags

- **<p> </p>** - Um simples parágrafo:
 - align – Alinhamento do parágrafo: left, right, center e justify.
- Manipuladores de texto:
 - ** ** - Texto em negrito;
 - **<i> </i>** - Texto em itálico;
 - **<u> </u>** - Texto sublinhado;
 - **<strike> </strike>** - Texto riscado.
- **<a> ** - Cria um link e inclui atributos em comum:
 - href – O URL do documento que será vinculado a este. Para e-mail: mailto e link externo: http

Exercicio

1. Criar uma estrutura básica para o html.
2. Criar 5 paragrafos, de acordo com os seguintes itens:
 - o **Texto em negrito**
 - o *Texto em itálico*
 - o Texto sublinhado
 - o ~~Texto riscado~~
 - o Texto com Link

Tags

- ** ** - Uma lista ordenada:
 - type – Tipos de caracteres ordenados: A, a, I, i, 1.
- ** ** - Uma lista não ordenada
 - type – Tipos de caracteres não ordenados: disk, square, circle.
- ** ** - Um item da lista:
 - value – Numeração individual do item da lista;
 - type – Tipos de caracteres ordenados: A, a, I, i e 1 para listagem ordenada e disk, square e circle para não ordenada.

Exercicio

1. Criar uma estrutura básica para o html;
2. Criar 5 paragrafos, de acordo com os seguintes itens;
3. **Criar duas listas com a tabuada do 5 no modelo abaixo:**
 1. $1 \times 5 = 5$
 2. $2 \times 5 = 10$
 3. $3 \times 5 = 15$
 4. $4 \times 5 = 20$
 5. $5 \times 5 = 25$
 6. $6 \times 5 = 30$
 7. $7 \times 5 = 35$
 - $8 \times 5 = 40$
 - $9 \times 5 = 45$
 - $10 \times 5 = 50$

Tags

- **<div> </div>** - Agrupador de conteúdo;
- **<hr>** - Uma régua horizontal
- **
** - Uma quebra de linha
- **** - Insere uma imagem inline no documento e inclui atributos comuns:
 - src – O URL da imagem;
 - align – Determina o alinhamento de uma determinada imagem: top, middle, bottom, left e right;
 - height – É a altura sugerida em pixels;
 - width – É a extensão sugerida em pixels;
 - border – Largura da borda.

Exercicio

1. Criar uma estrutura básica para o html;
2. Criar 5 paragrafos, de acordo com os seguintes itens;
3. Criar duas listas com a tabuada do 5 no modelo abaixo;
4. **Criar duas divs cada um contendo uma imagem, separados por uma linha horizontal.**

Tags

- **<table> </table>** - Cria uma tabela;
 - border - Borda da tabela
- **<tr> </tr>** - Uma linha na tabela;
- **<th> </th>** - Um cabeçalho de célula da tabela;
 - align – Alinhamento horizontal
 - valign – Alinhamento vertical
 - bgcolor – Cor de plano de fundo
- **<td> </td>** - Define uma célula de dados da tabela
 - align – Alinhamento horizontal;
 - valign – Alinhamento vertical;
 - width – Largura da célula;
 - height – Altura da célula;
 - bgcolor – Cor de plano de fundo.

Tags de acessibilidade

- **<article>**: Define um artigo;
- **<aside>**: Define o conteúdo além do conteúdo da página;
- **<figure>**: Define um grupo de mídia e seus captions;
- **<footer>**: Define o rodapé de uma página;
- **<header>**: Define o cabeçalho de uma página;
- **<nav>**: Define os links de navegação;
- **<section>**: Define uma área ou seção.

Exercicio

1. Criar uma estrutura básica para o html;
2. Criar 5 paragrafos, de acordo com os seguintes itens;
3. Criar duas listas com a tabuada do 5 no modelo abaixo;
4. Criar duas divs cada um contendo uma imagem, separados por uma linha horizontal;
5. **Criar uma tabela com 4 linhas e 3 colunas e com borda de tamanho 1.**



<https://www.codenation.dev>

CODENATION

PARA EMPRESAS

LOGIN

COMECE AGORA

/ NOSSA METODOLOGIA

- Aprendizado com base em desafios e projetos práticos;
- Encontros semanais com tutores de mercado e mentorias *one on one*;
- Possibilidade de conexão com empresas do Brasil e do exterior.





CSS

(Cascading Style Sheets)



```
...
<style>
  p {
    text-align: center;
    color: red;
  }
</style>
</head>
<body>
  <p>Conteúdo da página</p>
  ...

```



```
...
<style>
  p {
    text-align: center;
    color: red;
  }
</style>
</head>
<body>
  <p>Conteúdo da página</p>
...

```

p {

Seletor



...

```
<style>
```

```
  p {  
    text-align: center;  
    color: red;  
  }  
</style>
```

```
</head>
```

```
<body>
```

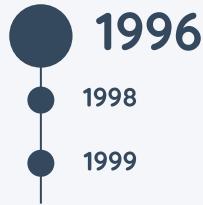
```
  <p>Conteúdo da página</p>
```

...

Propriedade

text-align: center;

CSS 1



- 53 propriedades;
- Principais propriedades:
 - Background;
 - Border;
 - Font;
 - Margin;
 - Padding.

CSS 2



- 122 propriedades;
- Principais mudanças:
 - Font family;
 - Font-size;
 - Color;
 - Background color;
 - Height;
 - Width;
 - Posição.

CSS 3



- 363 propriedades;
- Principais mudanças:
 - Translate;
 - Transition;
 - Animations.

Seletores

#identificador



.html

```
<div id="container">  
    ...  
</div>
```



.css

```
#container {  
    width: 960px;  
    margin: auto;  
}
```

Seletor do tipo **identificador** permite atingir um elemento com a propriedade *id* definida na tag HTML. Seletores desse tipo são rígidos e não permitem reuso. Por isso não são recomendados.

.classe



.html

```
<div class="container">  
    ...  
</div>
```



.css

```
.container {  
    width: 960px;  
    margin: auto;  
}
```

Esse é um seletor de **classe**. A diferença entre um identificador e uma classe é que, com a classe, você pode selecionar vários elementos. Use uma classe quando você precisar estilizar um grupo de elementos.

elemento



.html

```
<div>  
  ...  
</div>
```



.css

```
div {  
  width: 960px;  
  margin: auto;  
}
```

Esse é um seletor de **classe**. A diferença entre um identificador e uma classe é que, com a classe, você pode selecionar vários elementos. Use uma classe quando você precisar estilizar um grupo de elementos.

Propriedades

Propriedades

- **font-weight:** Define a propriedade de intensidade de uma fonte na sequência de valores de ‘100’ a ‘900’, cada número indica uma fonte mais intensa (escura) que o valor anterior;
- **font-size:** Define o tamanho da font;
- **text-decoration:** Define um efeito decorativo no texto. Podendo entre eles ser:
 - none - sem decoração;
 - underline -sublinhado;
 - line-through -linha cortando o texto;
 - blink - efeito piscante.

Propriedades

- **text-transform:** Controla os efeitos de capitalização do texto. Com ela, podemos definir a caixa das letras (minúsculas e maiusculas):
 - capitalize – transforma o primeiro caracter de cada palavra em maiúscula;
 - uppercase – transforma todas as letras de todas as palavras em maiúsculas;
 - lowercase – transforma todas as letras de todas as palavras em minúsculas;
 - none – cancela algum valor que tenha sido herdado.
- **color:** Define a cor do texto de um texto:
 - color: red;
 - color: rgb(255,0,0);
 - color: #FF0000;

Propriedades

- **width:** Define o comprimento (largura) de um elemento.
 - width: 100px;
- **height:** Define a altura de um elemento.
 - height: 50px;
- **border:** Define bordas para um elemento
 - border:1px solid #CCC;

Propriedades

- **margin:** Controla as margens de um elementos. Se forem indicados quatro valores, eles dizem respeito, respectivamente, às margens superior, direita, inferior e esquerda. Se for fornecido apenas um valor, ele é aplicado às quatro margens.
 - margin: 15px;
 - margin: 10px 1px 5px 20px;
 - margin-top: 10px;
 - margin-right: 1px;
 - margin-bottom: 5px;
 - margin-left: 20px;

Propriedades

- **padding:** Controla os espaçamentos de um elementos. Se forem indicados quatro valores, eles dizem respeito, respectivamente, aos espaçamentos superior, direito, inferior e esquerdo. Se for fornecido apenas um valor, ele é aplicado aos quatro espaçamentos.
 - padding: 5px;
 - padding: 10px 5px 5px 10px;

<https://www.codenation.dev>

CODENATION



Encontros semanais com tutores de mercado e mentorias *one on one*;

Possibilidade de conexão com empresas do Brasil e do exterior.



<https://www.codenation.dev>

CODENATION



/ NOSSA METODOLOGIA

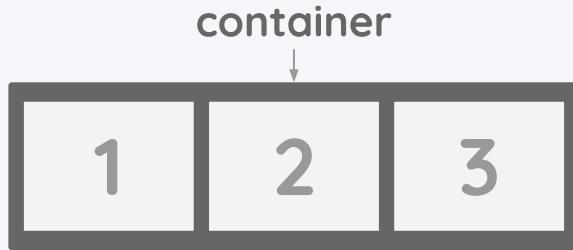
Aprendizado com base em desafios e projetos práticos;



FLEXBOX

(Cascading Style Sheets)

Estrutura



Propriedades para o pai
(flex container)



Propriedades para as crianças
(itens flexíveis)



Flex container



Torna o elemento um flex container automaticamente transformando todos os seus filhos diretos em flex itens.



```
.container {  
  display: flex;  
}
```



flex-direction

row



row-reverse





column



column-reverse



Define a direção dos flex itens. Por padrão ele é row (linha), por isso quando o display: flex; é adicionado, os elementos ficam em linha, um do lado do outro.



```
.container {  
    flex-direction: row | row-reverse | column | column-reverse;  
}
```



justify-content

flex-start



flex-end



center



space-between

O primeiro grudado no início e o último no final.



space-around

Os espaçamentos do meio são duas vezes maiores que o inicial e final



space-evenly

Todos os espaçamentos são iguais

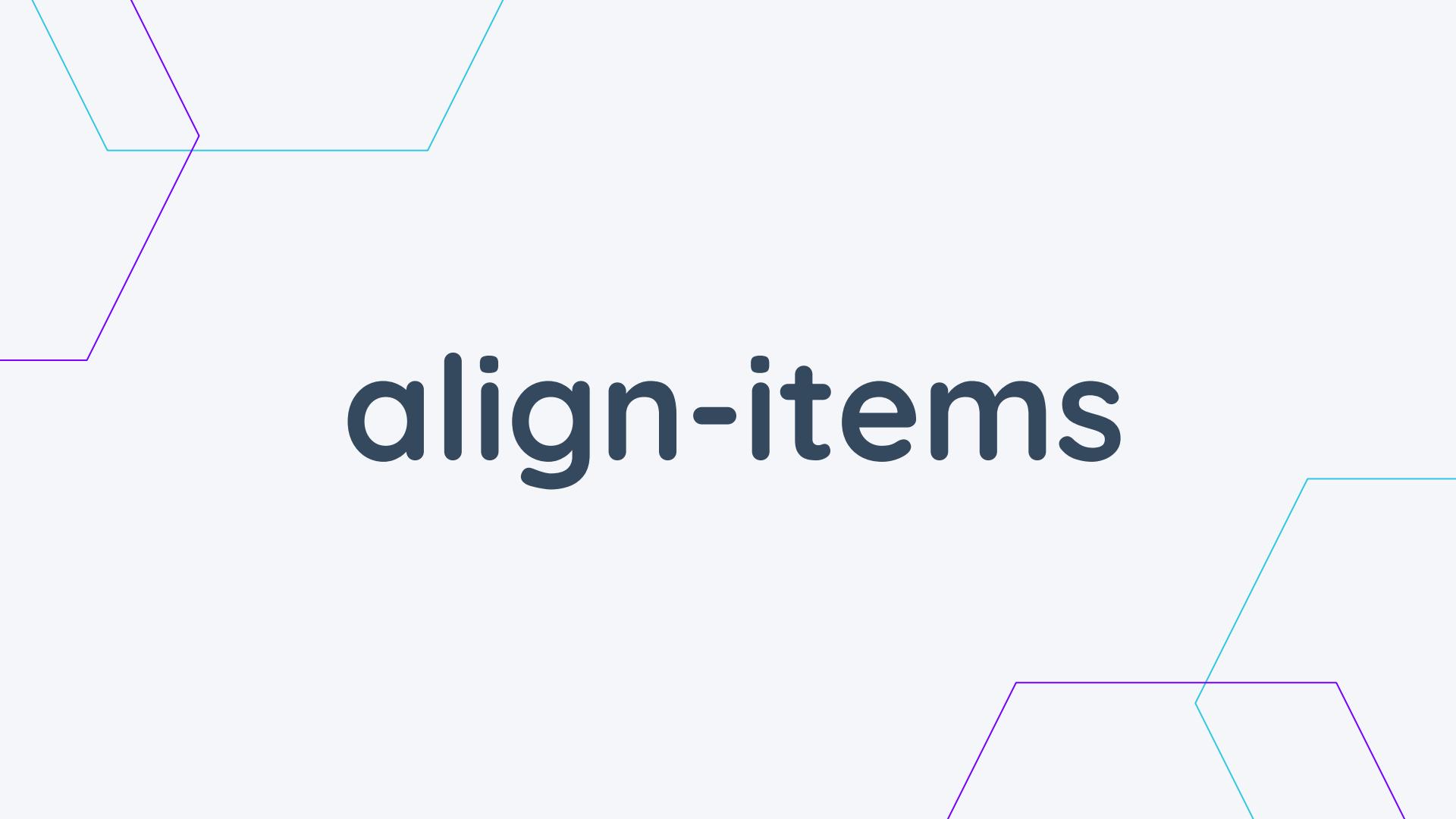




Isso define o alinhamento ao longo do eixo principal. Ele ajuda a distribuir o espaço livre extra restante quando todos os itens flexíveis de uma linha são inflexíveis ou flexíveis, mas atingiram seu tamanho máximo.

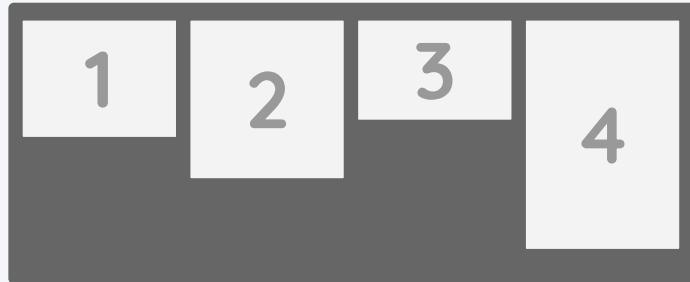


```
.container {  
  justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;  
}
```



align-items

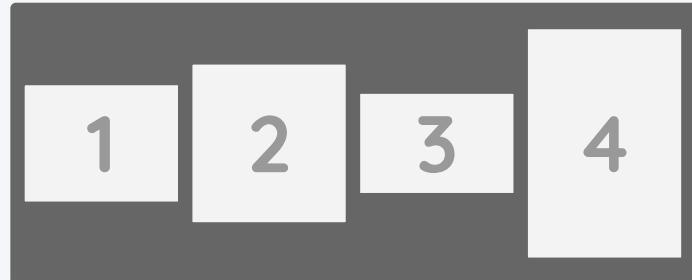
flex-start



flex-end



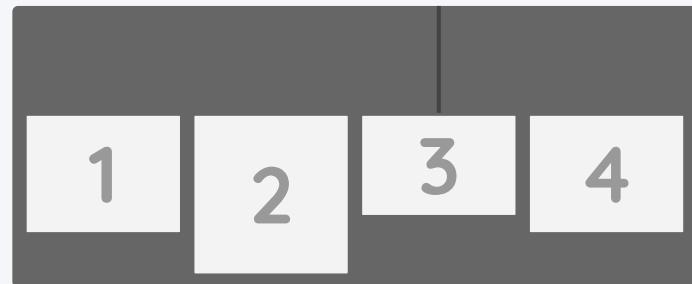
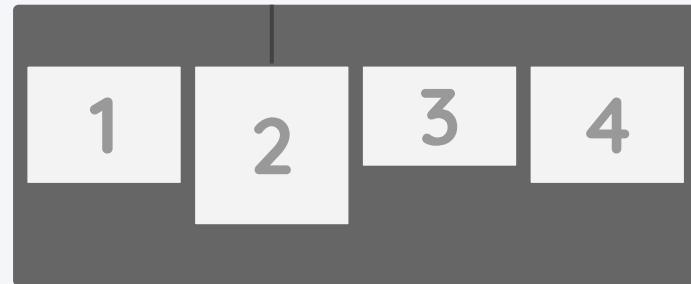
center



stretch



Baseline





Isso define o comportamento padrão de como os itens flexíveis são dispostos ao longo do eixo cruzado na linha atual. Pense nisso como a *justify-content* versão para o eixo cruzado (perpendicular ao eixo principal).



```
.container {  
  align-items: stretch | flex-start | flex-end | center | baseline;  
}
```