INTRODUÇÃO AO DESENVOLVIMENTO WEB

WEB DEV 4 NOOBIES

GUSTAVO VIEGAS

www.gfviegas.com

- Mais de 5 anos de experiência profissional em desenvolvimento;
- Trabalhou a maior parte da carreira com desenvolvimento web;
- Atualmente é prestador de serviços em soluções de TI, em diversas áreas;
- Graduando em Ciência da Computação na Universidade Federal de Viçosa - Campus Florestal;
- Entusiasta, evangelizador e ativista (~~chato!~~) da comunidade dev;

CARACTERÍSTICAS DO CURSO

- Aulas com apresentação do conteúdo;
- Live-coding com demonstrações práticas do conteúdo estudado;
- Exercícios e revisões;
- Projeto final de um layout responsivo simulando um MVP;

CONTEÚDO DO CURSO

1. Como funciona a web;
2. Sintaxe do HTML;
3. Tags HTML;
4. Sintaxe do CSS;
5. Seletores CSS;
6. HTML semântico;
7. Responsividade;
8. Boas práticas;
9. Projeto final;

AS CAMADAS DA WEB – FRONT E BACK

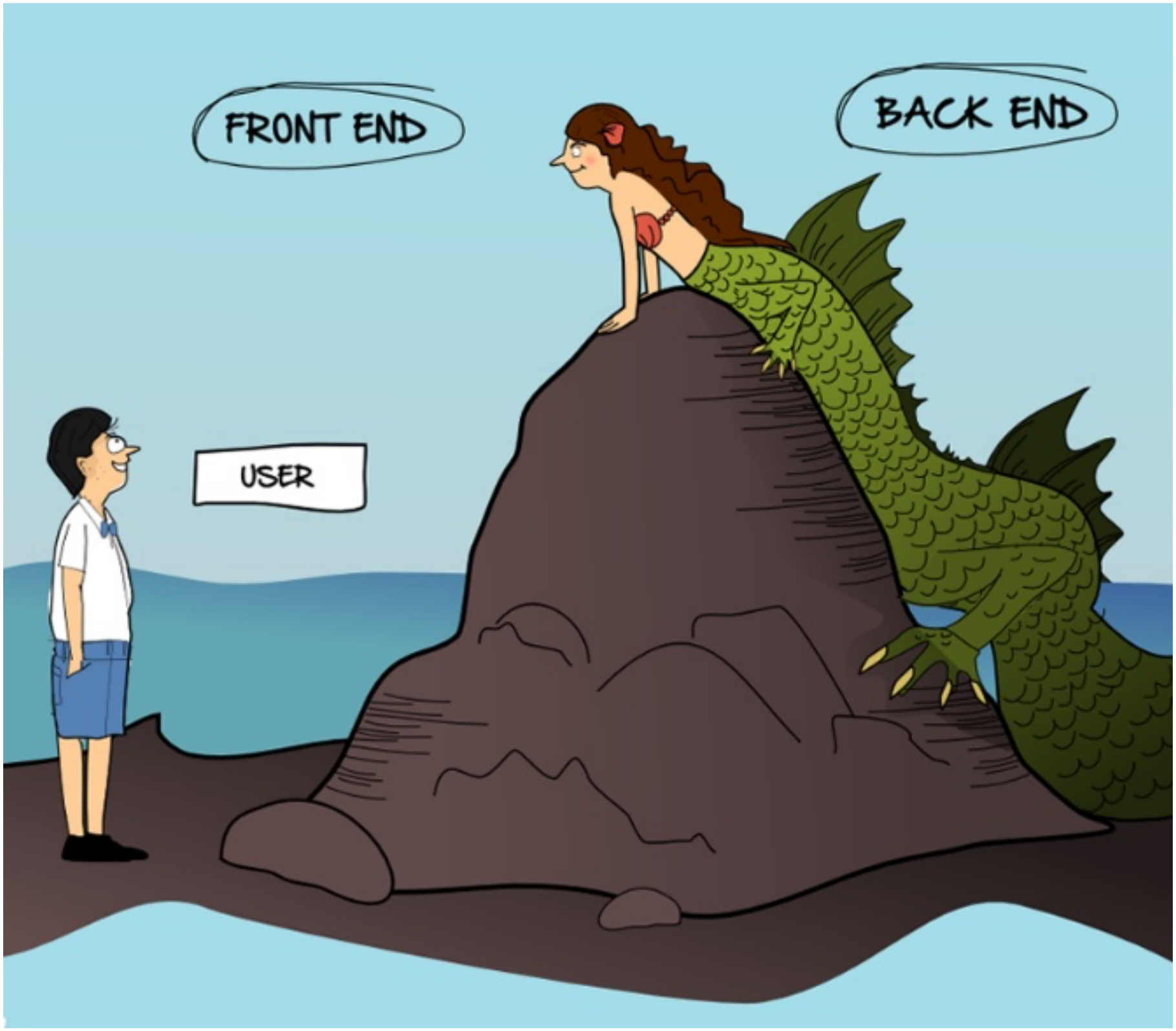
- A interação de usuários com um sistema web passa por algumas camadas.
- Apesar de ter responsabilidades diferentes, devem ter uma harmonia entre elas para o funcionamento correto de uma aplicação.
- Se comunicam através de requisições HTTP.

O FRONTEND

- "Interface" que possui todos os elementos que o usuário interage diretamente.
- Pode ser uma página web, um formulário, um aplicativo, etc.
- Roda diretamente no dispositivo do usuário, como os browsers. Conhece apenas a linguagem de marcação HTML, de estilo CSS e de programação JavaScript (com muitas limitações).

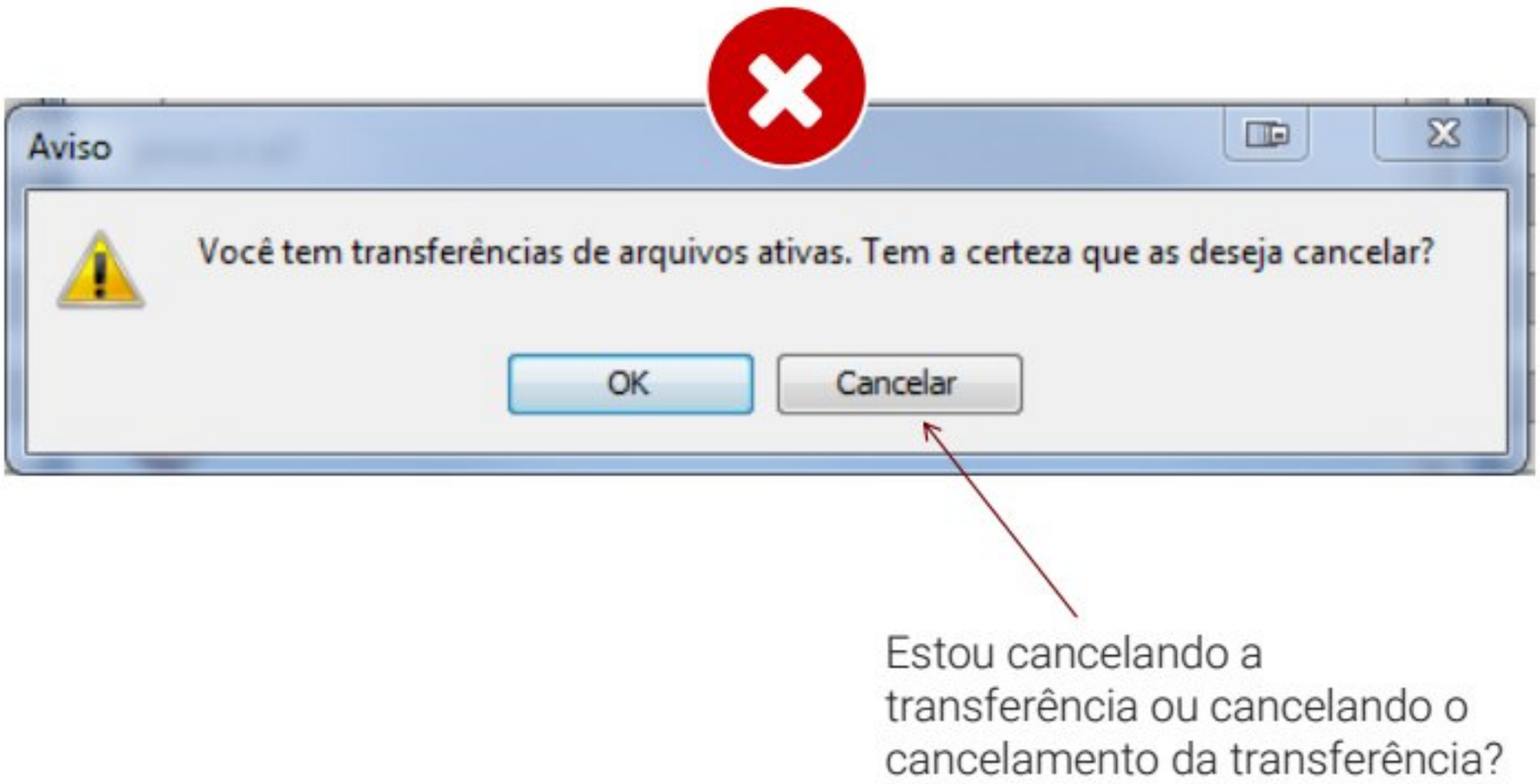
O BACKEND

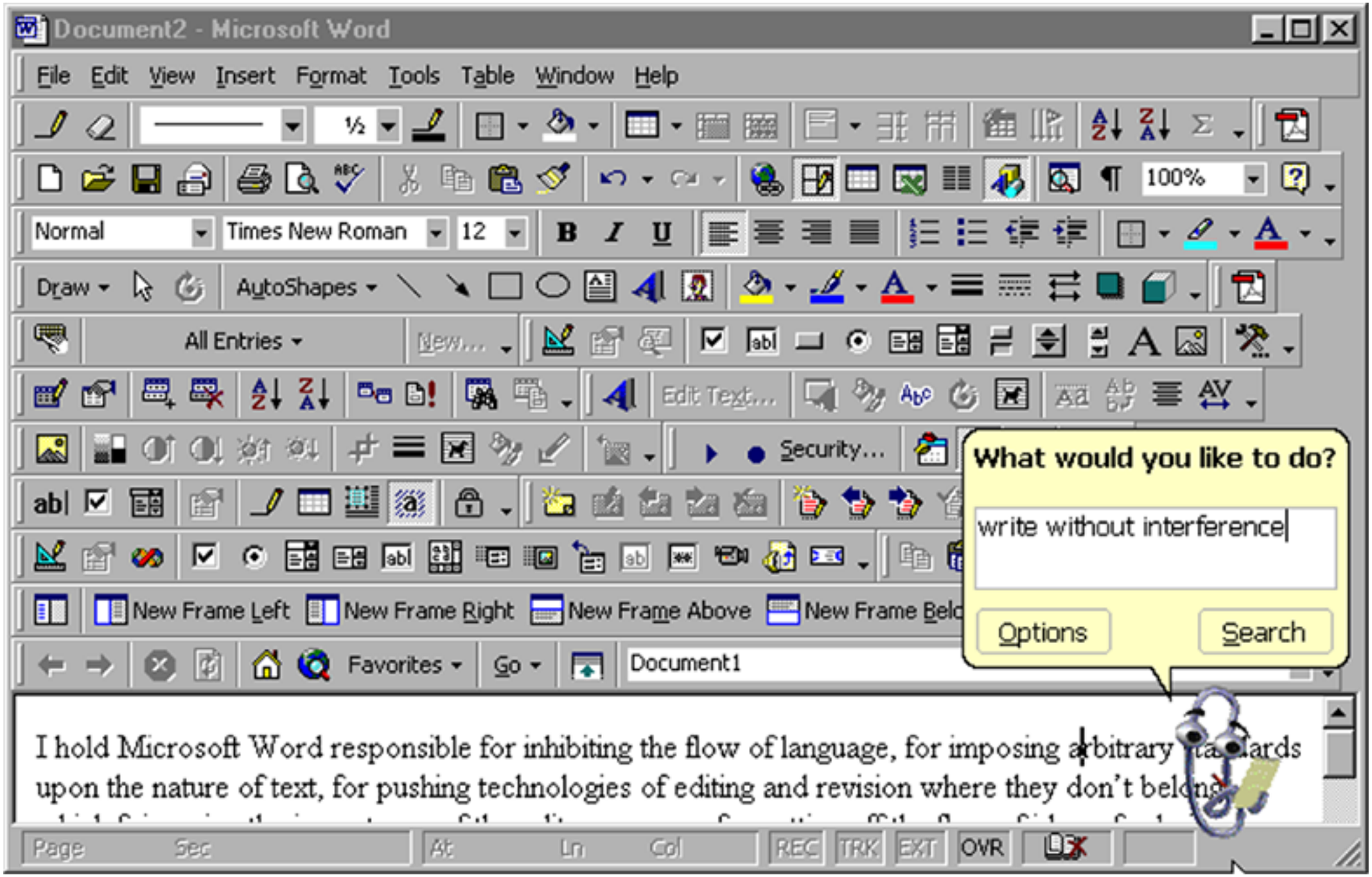
- "Interface" que possui todas as regras de negócio, modelagem de dados e comunicações da aplicação.
- Recebe e retorna dados através de protocolos bem definidos
- Faz o trabalho sujo, que o usuário não tem conhecimento ou visualização.
- Rodam em servidores dedicados com linguagens de programação como JavaScript, Java, PHP, Python, Ruby, C#, C++..... anything!


FRONT END
BACK END
USER

INTERFACE HOMEM-MÁQUINA

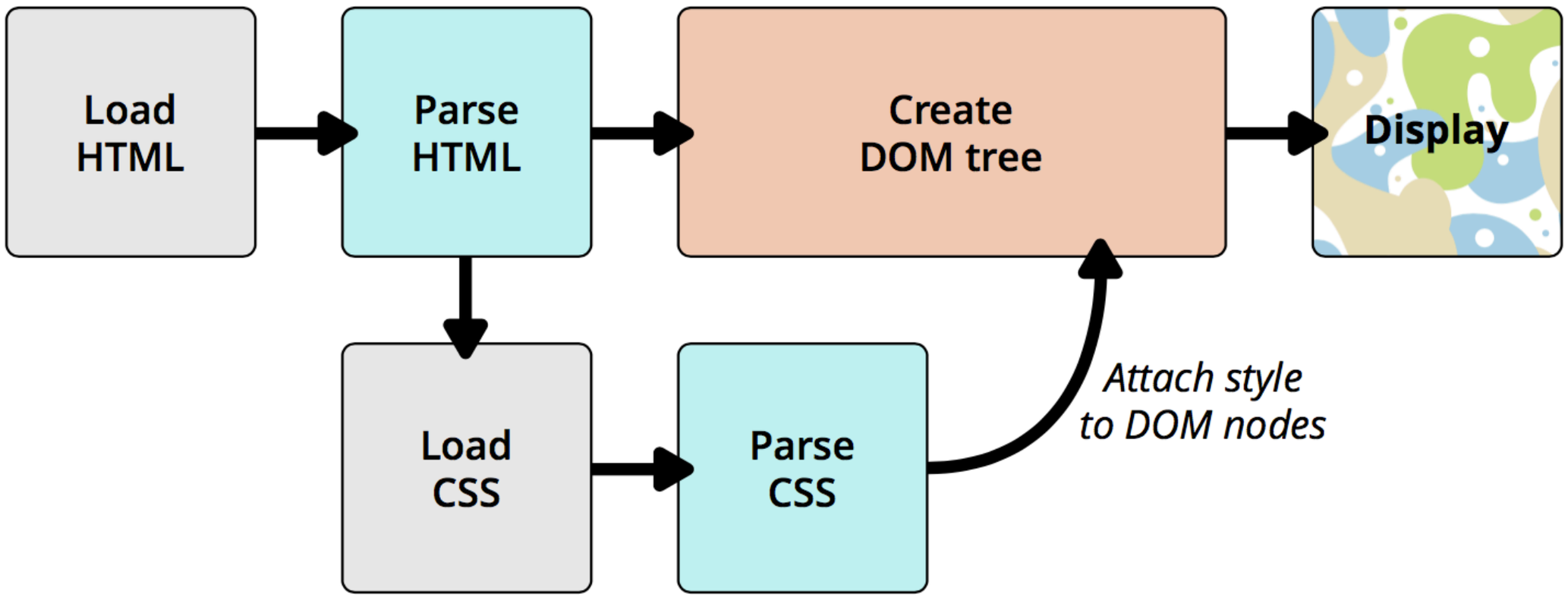
- Parece contraintuitivo, mas... muitas vezes o mais importante de um sistema é o seu **FRONTEND**!!!
- Um layout funcional, intuitivo e convidativo é essencial para o sucesso de uma aplicação.
- Se o usuário não conseguir, ou não gostar de utilizar um sistema, por mais útil que seja, ele se torna instantaneamente descartável para ele!
- Pense nos sistemas e aplicativos que você usa. Quais deles tem uma interface difícil de ser utilizada?


Aviso
Você tem transferências de arquivos ativas. Tem a certeza que as deseja cancelar?
OK
Cancelar
Estou cancelando a transferência ou cancelando o cancelamento da transferência?


Document2 - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Normal
Times New Roman
12
Draw
AutoShapes
All Entries
New...
Edit Text...
Security...
What would you like to do?
write without interference
Options
Search
New Frame Left
New Frame Right
New Frame Above
Favorites
Go
Document1
I hold Microsoft Word responsible for inhibiting the flow of language, for imposing arbitrary standards upon the nature of text, for pushing technologies of editing and revision where they don't belong
Page
Sec
At
Ln
Col
REC
TRK
EXT
OVR

O DOM

- O navegador converte HTML e CSS para dentro do **DOM** (**D**omain **O**bject **M**odel). O DOM representa o documento na memória do computador (navegador).



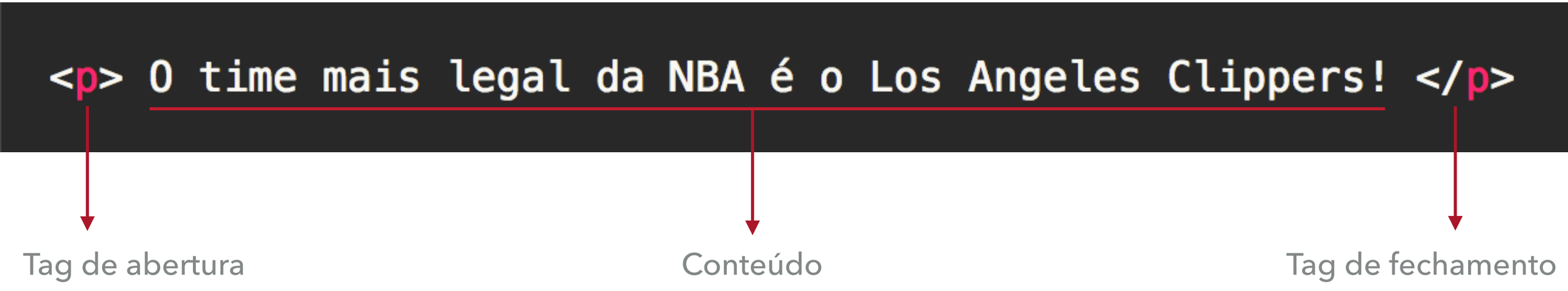
INTRODUÇÃO A HTML

- É a linguagem que o navegador interpreta para a exibição de conteúdo.
- HTML é uma sigla para **H**yper**T**ext **M**arkup **L**anguage.
- Cada elemento visual de uma página web consiste de uma ou mais tags.
- Normalmente, websites possuem centenas de tags, na maioria das vezes aninhadas.
- Exemplo: a página inicial do g1 possui mais de 4200 tags!

INTRODUÇÃO A HTML

- Existem diversas tags, com os mais diversos propósitos.
- Exemplos: parágrafos, tabelas, links, imagens, rodapés, etc.
- Um elemento HTML consiste basicamente de um conteúdo encapsulado por uma tag de abertura e uma tag de fechamento.

A ANATOMIA DE UM ELEMENTO HTML



- Tag de abertura: nome do elemento (p) envolvido entre colchetes angulares de abertura (<) e fechamento (>). Indica onde o elemento começa ou inicia o efeito.
- Conteúdo: pode ser um texto, ou outro elemento.
- Tag de fechamento: O mesmo de abertura, mas tem um / antes do nome do elemento. Indica o fim do elemento.

ANINHANDO ELEMENTOS

```
<p> O time <strong> mais </strong> legal da NBA é o Los Angeles Clippers! </p>
```

Quando colocamos elementos dentro de outros elementos, chamamos isto de **aninhamento**. Para por exemplo, enfatizar uma palavra em negrito, podemos usar a tag strong.

```
<p> O time <strong> mais legal da NBA é o Los Angeles Clippers! </p> </strong>
```

Deve-se, entretanto, garantir que quando abrimos um elemento dentro de outro, o último elemento aberto **deve** ser o primeiro a ser fechado. O código acima, portanto, está errado.

ERROS DE SINTAXE OU ABERTURA

No slide anterior, um elemento sobrepôs outro, o que é essencialmente *errado*. Entretanto, o navegador não *compila* código HTML e deixa de interpretá-lo caso algo esteja errado.

Ao invés disso, o navegador vai tentar *adivinhar* o que você quis dizer e exibir o conteúdo assim mesmo.

Isso gera resultados inesperados! **Nunca** deixe o seu código errado mesmo que ele apareça corretamente no navegador!

TAGS SEM CONTEÚDO

Existem tags que são especiais ou já exibem algum elemento visual e não possuem conteúdo dentro delas. Neste caso, não há necessidade de se criar uma tag de fechamento.

Exemplos: imagens, inputs, meta tags, source de CSS, etc..

```
1    <img src="./imagens/logo.png">
2    <meta name="charset" content="utf-8">
3    <link rel="stylesheet" href="./css/master.css">
```

ATRIBUTOS

Além dos elementos mostrados anteriormente, as tags também podem possuir atributos.

Os atributos adicionam peculiaridades e nos permite identificar e customizar cada tag.

Por exemplo: um campo de entrada (input) pode ser do tipo numérico, texto, senha ou de data. Não há necessidade de ter uma tag pra cada tipo de input, basta alterarmos um atributo.

ATRIBUTOS

```
<input type="text" name="username" value="">
<input type="password" name="senha" value="">
```

Os atributos seguem a sintaxe **nome**="**valor**". As tags podem ter inúmeros atributos e o valor deve estar em torno de aspas.

```
<img src="./imagens/logo.png class="imagemLogo"">
```

Deve-se garantir o fechamento das aspas e o fechamento de um atributo antes de começar outro. Também é importante verificar por atributos duplicados na mesma tag.

COMENTÁRIOS

A linguagem HTML permite a inserção de comentários delimitando-os com **<!--** e **–>**

Qualquer elemento dentro das tags delimitadoras de comentários será ignorada pelo browser e não será interpretada de qualquer maneira.

```
<!-- Aqui eu insiro um comentário -->
<img src="./imagens/logo.png">
<!-- <input type="password" name="senha" value="">
A tag acima não será exibida, pois está delimitada pelos comentários
-->
```

CASE-INSENSTIVE

A linguagem HTML é “case-insensitive”, ou seja, não distingue letras minúsculas e maiúsculas.

Porém é considerado boa prática manter a marcação HTML em letras minúsculas.

```
<!-- Todas as tags abaixo são válidas. -->
<H1> Aqui tem um título </h1>
<p> Aqui tem um <sTrOnG> parágrafo </STRONG> </p>
<IMG SRC="./IMAGENS/LOGO.PNG">
<!-- Porém os valores são case-sensitive... -->
```

O ESQUELETO DE UMA PÁGINA HTML

- Uma página HTML contém algumas tags ~~*obrigatórias*~~ que define aspectos importantes.
- Podemos dizer que uma página HTML é delimitada pela tag <html> e é dividida em duas importantes seções: o cabeçalho (**head**) e o corpo (**body**).
- Além dessas seções, é comum declarar a tag **!DOCTYPE** , que é a primeira tag do arquivo, especificando qual versão do HTML está sendo usada. No contexto do curso, usaremos o HTML 5.

O ESQUELETO DE UMA PÁGINA HTML

```
<!DOCTYPE html>
<html>
  <!-- Cabeçalho -->
  <head>
    <title>Título da página</title>
    <meta charset="utf-8">
  </head>

  <!-- Conteúdo da página -->
  <body>
    <h1>Hello world!</h1>
    <p>Página simples que só possui um título e um parágrafo.</p>
  </body>
</html>
```

CABEÇALHO (HEAD)

- A seção head é somente de interesse do navegador e é onde se configura elementos como o título da página (obrigatório), codificação, dimensões suportadas, idioma, atributos para compartilhamento, etc.
- É o primeiro elemento a ser carregado pelo navegador. Dessa forma é comum utilizar essa seção também para carregar arquivos de estilização.

CORPO (BODY)

- É a seção onde o navegador efetivamente renderiza seus elementos. É obrigatório ter ao menos um elemento dentro dele.
- O conteúdo é renderizado seguindo a ordem de declaração no arquivo. Então se você colocou um título e logo após um parágrafo, o parágrafo será renderizado após o título. Posteriormente veremos que é possível alterar a ordem de exibição com CSS.

DESAFIO #0

- Crie um arquivo “index.html” de acordo com as especificações:
  - Defina o título da página com o seu nome
  - Insira um texto na tag body
- Abra o arquivo no browser.

TÍTULOS E SUBTÍTULOS

- Títulos e subtítulos são elementos importantes e comuns.
- Em HTML existem 6 tipos de títulos, definidos do mais importante para o menos importante (de **<h1>** à **<h6>**).
- A tag **<h1>** é a mais importante, além do tamanho do título, também é mais relevantes para *crawlers* e outras ferramentas automatizadas.
- A tag **<h6>** é a menos relevante, e possui o menor tamanho comparado a **<h5>**, **<h4>**, **<h3>**, etc...

PARÁGRAFOS, QUEBRA DE LINHA E DIVISORES

- Um parágrafo é delimitado pela tag **<p>**. O navegador já formata como um parágrafo devidamente.
- Para fazer uma quebra de linha (equivalente a um shift+enter nos editores de documentos), é utilizada a tag self-closed **
**.
- A tag **<hr>** cria uma linha que ocupa todo o seu campo delimitador horizontalmente, útil para dividir seções de textos.

ÊNFASES

- A tag **** coloca o texto em negrito. Há uma tag correspondente (**<b>**) mas é mais fraca semanticamente. Além do seu estilo, ela é utilizada para dar uma ênfase forte ao seu conteúdo.

- A tag **** coloca o texto em itálico. A tag **<i>** é a sua correspondente mais fraca. Além de colocar o texto em itálico, ela é utilizada para dar uma ênfase simples.

- A tag **<small>** coloca o texto em uma fonte menor, útil para descrições de imagens ou textos de ajuda

LISTAS

- Existem dois tipos de listas: ordenadas e não-ordenadas.
- Para definir uma lista ordenada, se usa a tag **<ol>** e não-ordenada a tag **<ul>**. Cada elemento da lista é delimitado pela tag **<li>**.

```
<ol>
  <li>Lista</li>
  <li>Ordenada</li>
  <li>
    <ul>
      <li>Lista</li>
      <li>Não</li>
      <li>Ordenada</li>
    </ul>
  </li>
</ol>
```

DIVISORES

- Existe tags com o simples intuito de dividir o conteúdo em blocos. Esses blocos, que podem ou não ter diferenças visuais, são importantes para depois podermos delimitarmos e estilizarmos blocos de elementos.
- Existem tags semanticamente corretas para cada situação, mas por enquanto basta usar a tag ** **.
- Alguns blocos que podem ser interessante ser delimitados em tags são: cabeçalhos, menus laterais, rodapés, etc.

LINKS

- Para criar um link, basta usar a tag <a> e colocar no atributo href o destino.
- Pode-se usar o atributo target para definir onde esse link deve ser aberto (_self, _blank, _top).

```
<!-- Abre em uma nova aba/janela -->
<a href="https://www.twitter.com" target="_blank">Twitter</a>

<!-- Abre na aba/janela atual. Equiavalente a target="_self" -->
<a href="https://www.reddit.com">Reddit</a>
```

IMAGENS

- Para inserir uma imagem, basta usar a tag <img> e colocar no atributo src o caminho da imagem.
- Esse caminho pode ser uma URL externa ou interna, com qualquer extensão válida.

```
<img src="https://avatars1.githubusercontent.com/u/6178801">
```

DESAFIO #1

- Crie um arquivo “index.html” de acordo com as especificações:
  - O título deve ter o nome de uma série que você goste.
  - Deve possuir um título com o nome da série e um subtítulo com um slogan que você desejar.
  - Deve possuir uma lista ordenada dos seus 5 personagens favoritos.
  - Deve possuir dois parágrafos, separados por uma linha horizontal, contendo a descrição da série, um link para a série na wikipedia, ao menos um elemento com ênfase forte e um com ênfase simples.
- Abra o arquivo no browser e inspeccione os elementos no modo desenvolvedor.

DECLARANDO ESTILOS

- A tag HTML **<style>** define estilos para os elementos HTML.
- Estes estilos são definidos pela linguagem de estilização CSS.
- CSS é uma sigla para **C**ascading **S**tyle **S**heets (folha de estilo em cascata)
- A maioria das tags HTML já possuem estilos definidos por padrão. Eles podem ser substituídos, em cascata.

INTRODUÇÃO A CSS

- A tag **<style>** pode declarar estilos diretamente no HTML.
- Você também pode declarar estilos com o atributo style em qualquer elemento HTML.
- O mais comum é carregar os estilos de um arquivo externo, com extensão .css, com a tag **<link>**.
- Pode ser definidos inúmeras tags de estilização, entretanto o último estilo definido para um elemento X é o que tem maior prioridade e sobrescreverá qualquer estilo em conflito declarado anteriormente.

INTRODUÇÃO A CSS

- O mais comum é definir os estilos CSS na **<head>**.

```
<head>
  <title>Título da página</title>
  <meta charset="utf-8">
  <style>
    body {
      background-color: red;
    }
  </style>
  <link rel="stylesheet" href="./css/master.css">
</head>
```

- Caso o arquivo *master.css* possuir alguma definição de cor do background para o elemento body, ele que será considerado. Regras que não derem conflito são “misturados".

INTRODUÇÃO A CSS

- Uma regra de estilo é definida do tipo:

```
seletor {
    propriedade: valor;
}
```

- Uma mesma regra pode ser reescrita diversas vezes para um mesmo seletor.
- Regras são aplicadas a um elemento com diversos seletores diferentes.

UNIDADES DE MEDIDAS

- Para estilos de alturas, tamanhos de fontes, larguras, espaçamentos, margens, entre outras, é necessário usar uma unidade de medida.
- Pode-se usar unidades relativas ao *"container"* que o elemento está contido. Neste caso pode-se usar %.
- Existem também unidades relativas ao **tamanho da fonte**. Elas são *em*, *rem* e *ex*.
- E há as tradicionais unidades absolutas: *px*, *cm*, *in*, etc.

UNIDADES DE MEDIDAS

- O ideal é utilizar **px** para tamanhos de fontes (propriedade size).
- E para o restante utilizar **rem**, que é a unidade relativa mas ao root. Ou seja, sempre será relativo ao body!
- Isso faz com que quando algum usuário com baixa visão ou que simplesmente quer dar um “zoom" na página, o layout todo corresponda.
- E é uma convenção de boa prática.

CORES

- A forma mais simples de utilizar uma cor em CSS é usar o nome dela. Existem mais de 140 cores já definidas com seus nomes. Ex: blue, red, mediumaquamarine, gainsboro.

- Cada cor nomeada é associado a um valor hexadecimal. Este valor tem 6 caracteres e cada unidade de 2 caracteres define a força da sua cor RGB.

  Ex: **#FF00AA** = máximo de cor vermelha, nada de verde e um pouco de azul.

CORES

- Também se pode usar a sintaxe rgb(x,y,z) e rgba(x,y,z,o) onde o é a opacidade (canal alpha).
- Há também o formato HSL, HSV e VEC3 (pouco utilizado).

SELETORES

- Os seletores são, basicamente, o caminho para um ou mais elementos.
- Para facilitar a identificação de elementos, pode se definir classes e identificadores únicos nos seus elementos (**class** e **id**).
- A class pode ser reutilizada em inúmeros elementos e é identificado no css com o prefixo “.”
- A id é identificada no css com o prefixo “#”.

SELETORES

- Para o CSS não importa se o ID é único ou não.
- Mas para o JavaScript importa... e importa muito!
- Além disso existem validadores de HTML que acusam caso você utilize um mesmo ID para mais de um elemento.
- Os seletores podem ter todo o caminho até um elemento, onde seus filhos são separados por espaço.
- Existem também separadores para irmãos, filhos diretos, etc.

SELETORES

CSS Selectors

Selector	Role
p{ }	Tag selector, **all p tags**
#para{ }	Id para (***unique***)
.para1{ }	Class para1 (***multiple***)
p.para{ }	P tag with class para
P .para{}	P with child having class para
div p{}	p tag having parent div.
*{}	All tags{ **Universal Selector**}
h1, h3, h5{}	Only h1, h3 and h5 (***grouping***)
.para a{ }	A with parent para class
body{}	Parent of all tags

PSEUDO-CLASSES E PSEUDO-ELEMENTOS

- Pseudo-classes são seletores que seleciona um elemento baseado no seu elemento que não está presente no DOM. Exemplo: a:visited vai capturar todos os links visitados.
- Pseudo-elementos por sua vez são seletores que seleciona entidades que não estão presentes no HTML.
  Ex: p::first-line captura a primeira linha de um parágrafo.

LIVE-CODING

- Aprendendo algumas propriedades CSS, posicionamento, largura, altura, bordas, espaçamento, margens, floats e etc.
- Utilize o navegador para brincar com as regras de alguma página!

DESAFIO #2

- Modifique o HTML do desafio anterior, dividindo as seções devidas e adicionando as devidas classes e ids necessários.
- Defina o tipo e tamanho da fonte e um background na raiz do documento.
- Estilize a lista para que o primeiro elemento seja exibido em negrito. Os elementos de índice ímpar deve ter a cor vermelha.
- Defina bordas coloridas para todas os divisores definidos. Garanta que cada um tenha um espaçamento grande do outro.
- Estilize o título para que tenha uma fonte diferente da usada no resto do corpo.

DESAFIO #3

- Implemente o layout ao lado, para um jornal de Fake News.
- Os blocos verdes e amarelo devem ter parágrafos de texto.
- O bloco azul deve ser um rodapé com dados de contato.
- O bloco vermelho deve ter uma logo e um menu.