

# front-end

*prof. Lucas Ferreira*



engenharia de  
software



engenharia de  
computação





# Fundamentos da Web & Introdução ao Front-end: HTML e CSS





# Como funciona a internet?

- Conhecida como "rede mundial de computadores"
- O que é errado... na verdade é um conjunto de várias redes interligadas
- O que acaba "descentralizando" a internet (sem dono)
- **Rede Local x Rede Externa** (conexão entre computadores)
- **Rede Externa** → Provedor de Internet (TCP/IP)
- Cada ponto da rede têm um endereço IP
- Sites disponibilizados através de servidores (também conectados sobre IP)



## O Caminho da Internet





---

# Backbone

- Ponto inicial de referência do funcionamento da Internet
- Backbone significa Espinha Dorsal em inglês
- Responsável por interligar servidores que estão distantes
- São pontos-chave das redes que compõem o núcleo das redes de Internet
- Existem poucos backbones espalhados pelo mundo, e estes são os responsáveis por distribuir o acesso mundial a rede e se ligar as centrais das operadoras de internet



## Provedor de acesso (*maior*)

- → **Backbones** → Chegada de sinal aos provedores de acesso
- Empresas que contratam e se ligam ao sinal de backbones para distribuir dados aos seus usuários
- Em geral, empresas ligadas ao setor de telecomunicações
- (*Ex: Embratel, Oi, Vivo, Claro e etc*)



---

## Provedor de serviço

- Quem trafega e transporta os dados até os usuários
- Recebem os dados do provedor de acesso e distribuem aos usuários
- Por linha telefônica, fibra ótica ou via rádio (por tecnologia sem fio)
- Empresas devem sempre ser regulamentadas pela Anatel
- (podem inclusive serem os mesmos provedores de acesso)
- *(Ex: Vivo, Claro, Engeplus, Banda Turbo e etc)*



# E usuário final?

- O consumo não é passivo, o usuário não apenas "recebe" os dados
- O sinal de Internet passa a repetir todo o caminho novamente de forma inversa
- Usuário final, também envia sinais - com as suas requisições - para a Internet





---

# MAS E OS SITES?!?!





---

# Endereços dos sites x DNS

- Obviamente ninguém sabe o endereço IP de cada servidor de cada site
- Aí que entra o endereço “www”
- Endereços de sites são baseados na tecnologia DNS, que basicamente cria atalhos entre os endereços “www” à endereços IP



## Mas o que é o DNS?

- **Domain Name System - *DNS***, ou Sistema de Nomes de Domínios
- É um computador com uma espécie de banco de dados que relaciona o endereço "nominal" (www...) com o endereço real (número de IP)
- Os servidores de DNS traduzem o endereço para o IP do servidor do site
- Existe um servidor DNS configurado em sua rede de internet
- E cada domínio também indica um servidor DNS responsável por gerenciar suas rotas
- Algo Legal → whois do **registro.br**



## "Traçando a rota"

tracert ou traceroute [www.satc.edu.br](http://www.satc.edu.br)

```
traceroute to www.satc.edu.br (177.54.50.195)
 1  10.2.1.1 (10.2.1.1)  7.178 ms  111.821 ms  26.659 ms
 2  10.65.0.1 (10.65.0.1)  39.676 ms  5.593 ms  44.607 ms
 3  192.168.255.230 (192.168.255.230)  107.088 ms  8.268 ms  9.199 ms
 4  177.54.51.242 (177.54.51.242)  6.786 ms  11.509 ms  7.686 ms
 5  189.28.177.22 (189.28.177.22)  14.897 ms  37.153 ms  10.005 ms
 6  189.28.177.18 (189.28.177.18)  6.924 ms  45.036 ms  42.444 ms
```



---

# PROTOCOLOS





## Pela definição de protocolos (TI)

Um protocolo é uma convenção que controla e possibilita uma conexão, comunicação, transferência de dados entre dois sistemas computacionais.

Protocolo pode ser definido como "as regras que governam" a sintaxe, semântica e sincronização da comunicação.

Os protocolos podem ser implementados pelo hardware, software ou por uma combinação dos dois.



## Pela definição de protocolos (TI)

- Estabelecimento de ligação
- Negociação de várias características de uma conexão
- Inicializar, formatar, transmitir e finalizar uma mensagem



## Protocolos mais comuns da internet

- **IP** → Internet Protocol
- **TCP** → Transmission Control Protocol
- **HTTP(S)** → Hypertext Transfer Protocol → Porta 80/443
- **FTP** → File Transfer Protocol
- **SSH** → Secure shell
- **POP3, SMTP e IMAP** → Protocolos de E-mail





---

# Protocolo HTTP

- Hypertext Transfer Protocol - **HTTP**, ou Protocolo de Transferência de Hipertexto
- Protocolo de Comunicação via Aplicação, utilizado para distribuição de informação
- **Hipermídia** → Hipertexto (texto com links)
- É a base para a comunicação de dados da World Wide Web e roda na porta 80
- Este protocolo tem sido usado pela WWW desde 1990
- Primeira versão HTTP/0.9, era um protocolo simples para a transferência de dados no formato de texto, com um único método de requisição, chamado GET
- Entre 1992 e 1996 foi desenvolvida o HTTP/1.0, agora também com POST e HEAD
- Junho de 1999 definição do HTTP/1.1
- Março de 2015 lançamento do HTTP/2



---

## Mas e o HTTPS?

- De forma "básica" é o HTTP com uma camada extra de segurança
- E roda em outra porta 443



## Sessão HTTP(S)

- É uma sequência de transações de rede de requisição-resposta
- Para que o protocolo HTTP consiga transferir seus dados pela Web são necessários os protocolos TCP e IP
- Inicia no Cliente → estabelecendo uma conexão TCP para uma porta (normalmente 80 ou 443)
- Um servidor HTTP ouvindo naquela porta espera por uma mensagem de requisição de cliente
- Recebendo a requisição, o servidor retorna uma linha de estado, como "HTTP/1.1 200 OK" e uma mensagem particular própria
- O corpo desta mensagem normalmente é o recurso solicitado (HTML) ou também um erro de requisição (404 Page Not Found)



---

# Introdução ao Front-end: HTML e CSS



---

Começando com o rei:

**O HTML!**





# HTML

- HyperText Markup Language, ou linguagem de marcação
- É a principal linguagem utilizada na criação de sites e serviços WEB
- Documentos HTML normalmente são interpretados por Navegadores
- Foi criada por Tim Berners-Lee junto da WWW
- Teve sua primeira especificação oficial em 1993
- Tornou-se padronizada a partir do HTML 2.0 em 1995
- Em 1999 publicou-se a especificação do HTML 4.0 (amplamente usada até hoje)
- No ano 2000 a linguagem tornou-se uma norma internacional → ISO 15445:2000
- Em 2014 publicou-se o HTML 5.0 (versão atual)



---

## A linguagem (*de marcação*)

- Todo documento HTML possui marcadores
- Que são palavras entre parênteses angulares
- Esses marcadores são os comandos de formatação da linguagem
- Um elemento é formado por um nome de marcador (tag), atributos, valores e filhos

## Exemplo de elemento HTML com filho

```
<a href="http://www.satc.edu.br">Site da SATC</a>
```

## Exemplo de elemento sem filho

```
<br />  
<hr />  

```



# Estrutura Básica de um Documento HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="description" content="a descrição do seu site">
    <title>Título do Documento</title>
  </head>
  <body>
    <div>
      Tag para criar-se uma 'caixa, um bloco, mais utilizada
      com "Cascading Style Sheets (CSS ~ Folhas de Estilo em Cascata)
    </div>
    
    <a href="http://www.satc.edu.br">Site da SATC</a>
  </body>
</html>
```



---

## Tags importantes (*e quase obrigatórias*)

- `<html>` : define o início de um documento HTML
- `<head>` : define o cabeçalho de um documento HTML
- `<body>` : define o conteúdo principal, o corpo do documento e esta é a parte do documento HTML que é exibida no navegador



# Cabeçalho

## <HEAD>

- Inclui meta informações sobre documento
- Campos de configuração do site (*responsividade, favicons, preload e etc*)
- Usados por mecanismos de busca ou leitores de tela
- Elementos externos (assets) carregados com prioridade
- Tags mais usadas: `<title>`, `<meta>`, `<link>`, `<style>` e `<script>`
- Não confundir com o topo visual do site (HEADER)
- Site com dicas e possibilidades de elementos a serem usados dentro da tag `<head>`: <https://htmlhead.dev/>



# Corpo

## <BODY>

Qualquer elemento real, visual (ou não) a ser disposto em um site/aplicação web.

- `<header>` : topo do site
- `<footer>` : rodapé
- `<h1>`, `<h2>`, ... `<h6>` : tags de título por ordem de relevância
- `<p>` : parágrafo
- `<br />` : quebra de linha
- `<img />` : imagem
- `<table>`, `<tr>`, `<td>` : tabela (dados tabulares)
- `<div>` : divisões ou blocos
- `<b>`, `<i>`, `<u>` e `<s>` : negrito, itálico, sublinhado e riscado
- `<a>` : hiper-ligação, link ou âncora
- `<input>`, `<select>` ou `<textarea>` : campos de formulário
- `<button>` : botões

entre outras várias tags!



# HTML5

- É a quinta versão da linguagem HTML
- Traz consigo importantes mudanças quanto ao papel do HTML no mundo da Web
- Novas API's e funcionalidades sintáticas, semânticas e foco na acessibilidade
- Possibilita o uso de novos recursos antes possíveis apenas com a aplicação de outras tecnologias
- Suporte para as mais recentes multimídias
- Novas tags: `<video>`, `<audio>`, `<canvas>`, `<section>`, `<main>`, `<article>`, `<header>`, `<menu>` e `<nav>`
- Lista completa de novas tags:  
[https://pt.wikipedia.org/wiki/HTML5#Novos\\_elementos\\_do\\_HTML5](https://pt.wikipedia.org/wiki/HTML5#Novos_elementos_do_HTML5)



---

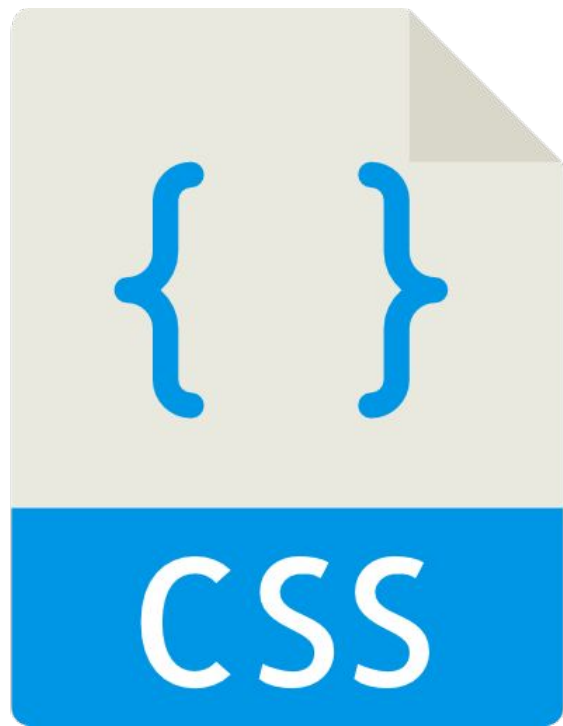
UM POUCO DE  
CÓDIGO...





—  
E para melhorar o sabor

O CSS!





## CSS

- Cascading **Style Sheets** ou folha de estilos em cascata
- É um mecanismo para adicionar estilo (cores, fontes, espaçamento, etc.) a um documento web
- Proposto por Håkon Wium Lie (Opera Foundation) em 1994
- Primeira versão oficial em 1996
- Sintaxe simples baseada em seletores e propriedades
- Documentos inline ou externos





# Comofas CSS?

- Uma instrução CSS consiste em um seletor e um bloco de declaração
- Cada declaração contém uma propriedade e um valor, separados por dois pontos (:)
- Cada declaração finalizada por ponto e vírgula (;)
- Seletores são usados para declarar a quais elementos de marcação um estilo se aplica

```
selector [, selector2, ...][:pseudo-class] {  
  property: value;  
}
```



## Seletores mais comuns

- elemento do tipo:

```
element_name { style definition; }
```

- todos os elementos com a classe:

```
.class_name { style definition; }
```

- o elemento com o id:

```
#id_of_element { style definition; }
```



—  
+ UM POUCO DE  
CÓDIGO...

**DO YOU....**



**WANNA PLAY A GAME**  
memegenerator.net



# Coisas importantes para estudar em CSS

- Propriedade **display**: <https://developer.mozilla.org/pt-BR/docs/Web/CSS/display>
- Propriedade **position**: <https://developer.mozilla.org/pt-BR/docs/Web/CSS/position>
- Display => Flex:  
<https://flexboxfroggy.com/>  
[https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS layout/Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox)
- Estados interativos (:hover, :focus):  
[https://developer.mozilla.org/en-US/docs/Learn/CSS/Building\\_blocks/Selectors/Pseudo-classes and pseudo-elements#user-action pseudo classes](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors/Pseudo-classes_and_pseudo-elements#user-action_pseudo_classes)
- Transições e Animações:  
[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS Transitions/Using CSS transitions](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions)  
[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS Animations/Using CSS animations](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Animations/Using_CSS_animations)



---

# BÔNUS ROUND





# Acessibilidade

*"Acessibilidade na web significa que pessoas com deficiência podem usar a web. Mais especificamente, a acessibilidade na web significa que pessoas com deficiência podem perceber, entender, navegar, interagir e contribuir para a web. E mais. Ela também beneficia outras pessoas, incluindo pessoas idosas com capacidades em mudança devido ao envelhecimento"*



---

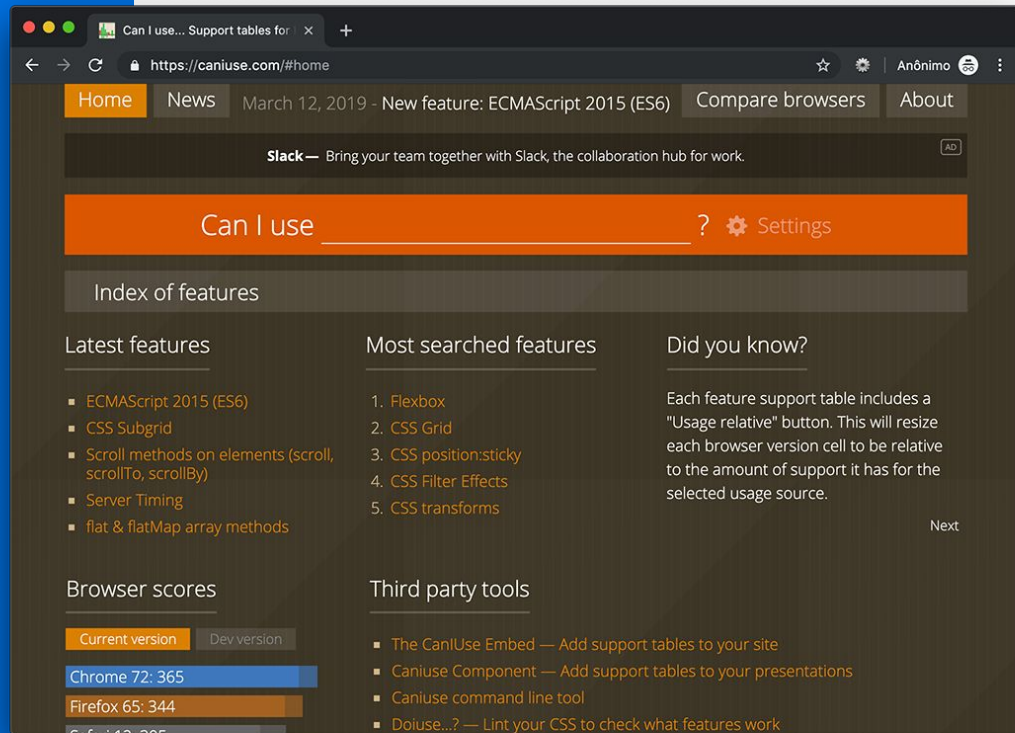
## HTML semântico

- Utilizar os elementos e tags corretas para cada função, destacando o conteúdo e utilidade de cada elemento de acordo com sua estrutura.
- Não usar tags fracas para necessidades fortes, nem substituir elementos de consistência real por outros "falsos"
- Trazer a devida importância aos leitores de telas e indexadores de sites de busca



# Can I use this?

<https://caniuse.com>







## Conteúdo para *review* & estudos

- <https://www.w3schools.com/html/default.asp>
- <https://www.w3schools.com/css/default.asp>
- <https://developer.mozilla.org/en-US/docs/Web/HTML>
- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- <http://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-I.html>



---

E por fim...

**Nosso 1º Exercício!**

valendo nota evidentemente



# Template de CHAT

Atendimento on-line

Atendente diz:

Blabla blabla blabla

Atendente diz:

Blabla blabla blabla

Você diz:

Blabla blabla blabla

Você diz:

Blabla blabla blabla

Digite sua mensagem...

ENVIAR



# 1º Exercício / Atividade

- Vamos praticar um pouco de HTML + CSS!
- Criar uma breve tela, única, em HTML, que junto de CSS chegue nos resultados da imagem anterior, criando um layout de atendimento on-line/chat "fake" (*fake pq não vai mandar mensagem de verdade*)
- **A entrega até a nossa próxima aula às 20h**
- **Como Entregar:** Criar um repositório no github (público) e inserir o exercício lá (por enquanto o mínimo, html, css e imagens apenas); OU...
- Para quem não se sentir confortável em mexer com github neste momento, zipar os arquivos do exercício e postar no Portal do Aluno / AVA ~ ainda vou abrir o espaço, aviso vocês no grupo quando já estiver liberado para entrega
- **Spoiler/Aviso:** Não percam esse exercício, em algumas aulas iremos dar mais uma "melhorada" nesse chat



—  
obrigado 🚀

