

Aula – 2

Introdução a Web

Disciplina: XDES03 – Programação Web

Prof: Phyllipe Lima Francisco
phyllipe@unifei.edu.br

Universidade Federal de Itajubá – UNIFEI
IMC – Instituto de Matemática e Computação

Agenda



- ❑ O que é a Internet
- ❑ O que é a Web
- ❑ O ciclo Requisição e Resposta
- ❑ A Trinca HTML - CSS - JavaScript



Internet

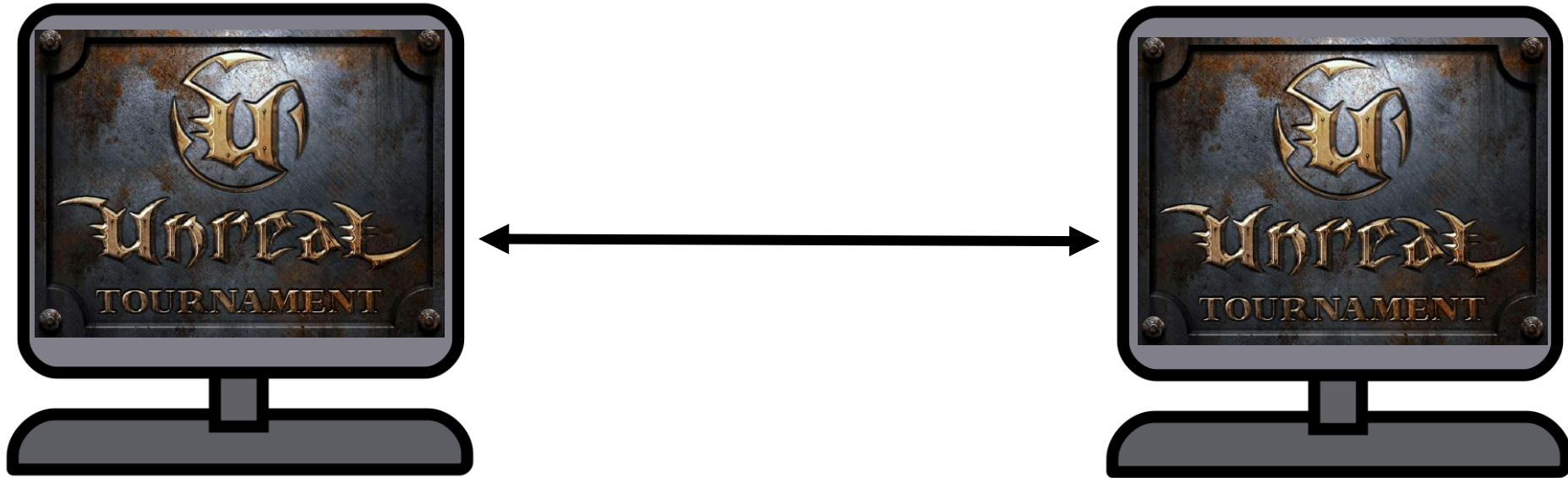


O que é a Internet?

“Uma rede de redes de computadores”

(uma quantidade grande demais da conta de computadores conectados...sô)

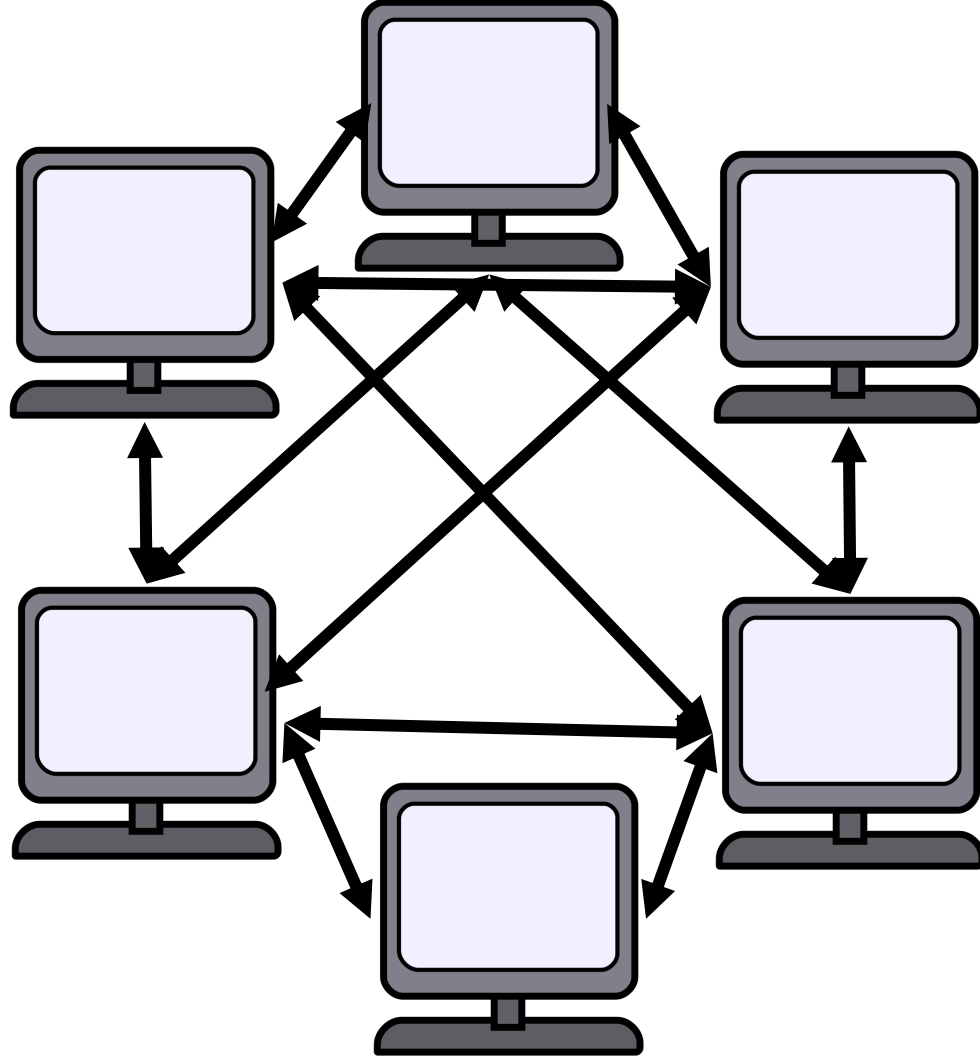
Para conectar dois computadores basta um cabo. Com isso eles conseguem trocar dados (e até jogar)





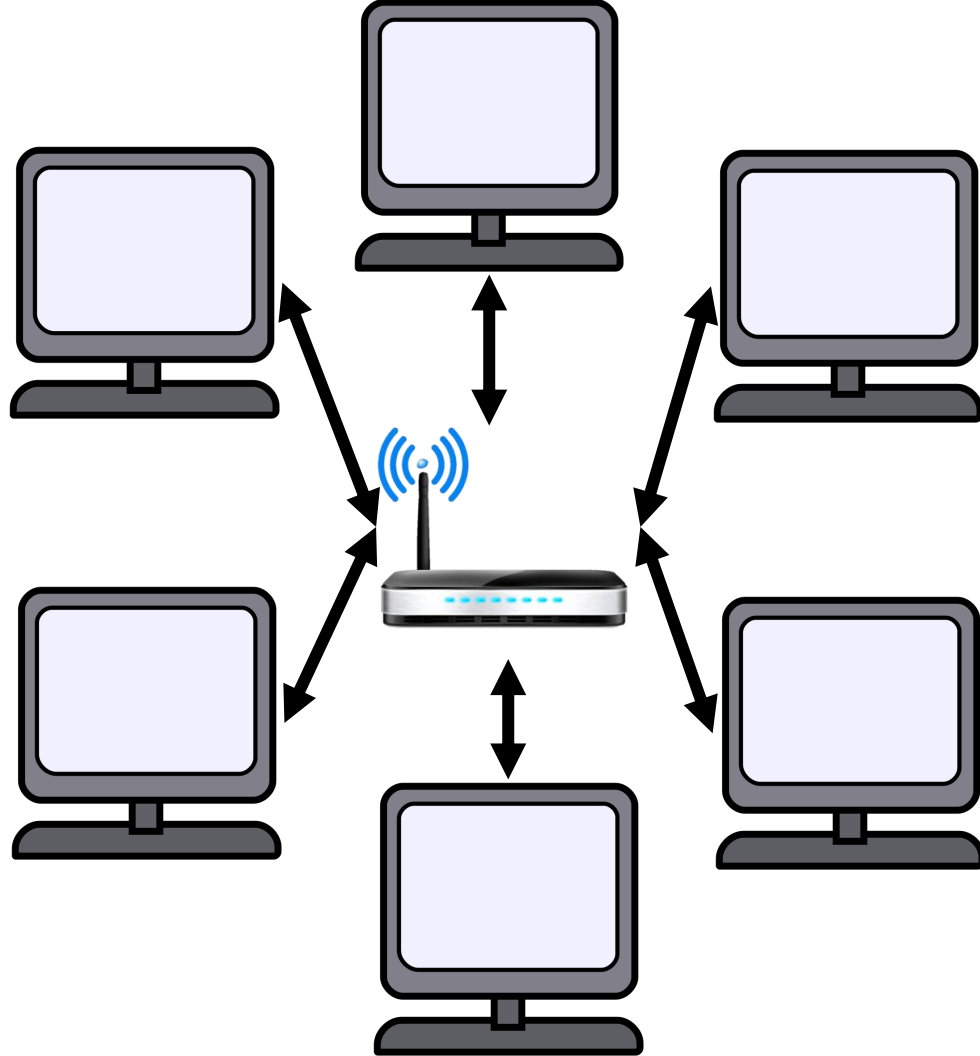
E se tivermos mais de
dois computadores?

Será viável utilizar cabos conectando
todos?



- ☐ Não parece adequado 😞
- ☐ E se precisarmos de mais computadores?



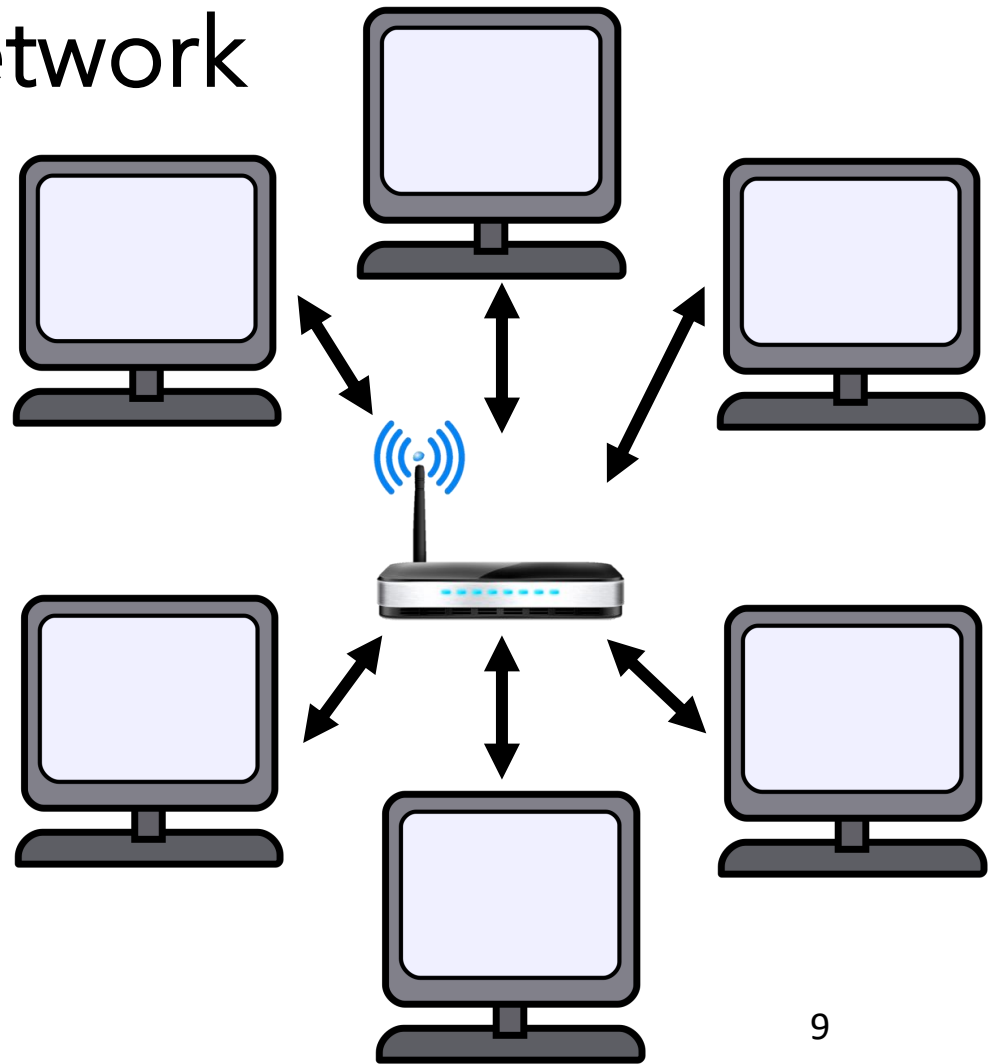


❑ Utilizando um ***roteador*** já melhora um pouco



LAN - Local Area Network

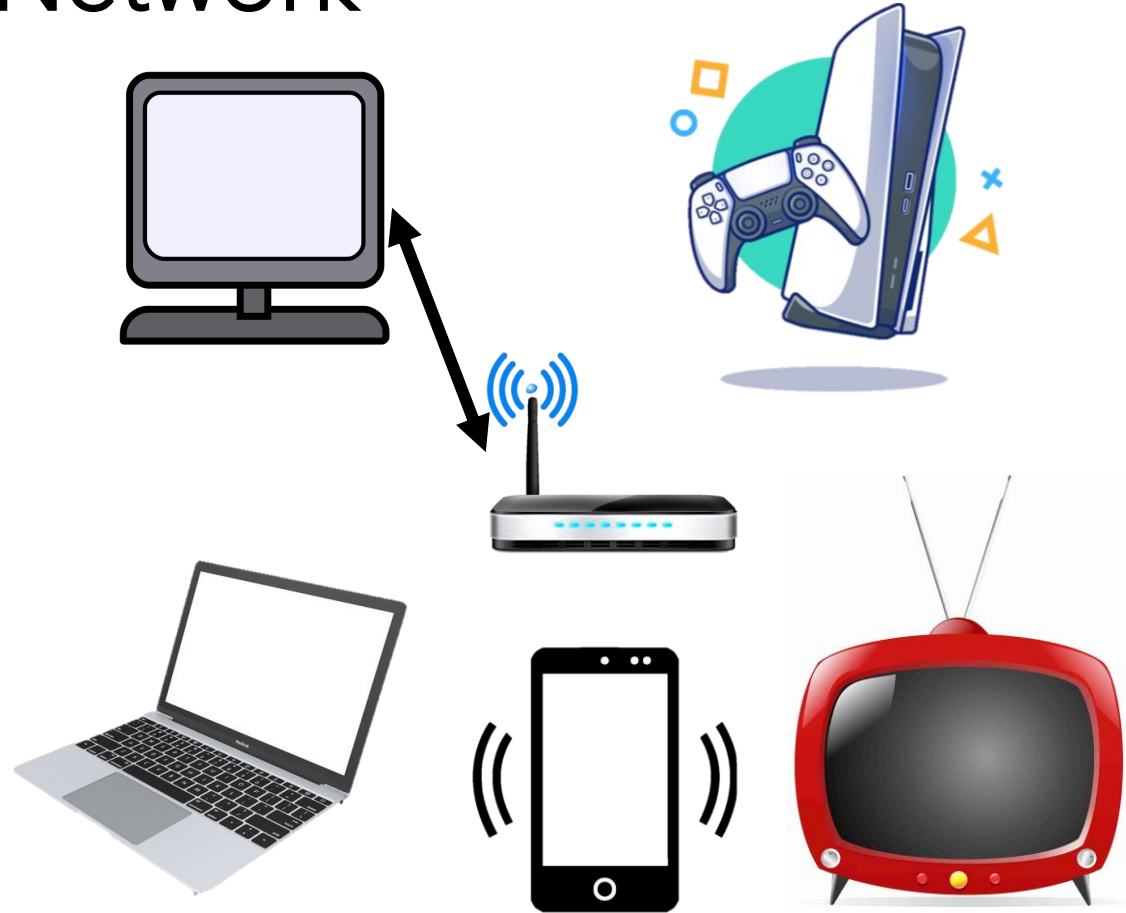
- ❑ Rede Local
- ❑ Dispositivos conseguem trocar dados apenas entre si
- ❑ Não consiste apenas em computadores pessoais



LAN - Local Area Network

❑ LAN pode conter:

- ❑ Televisores
- ❑ Videogames
- ❑ Dispositivos
móveis

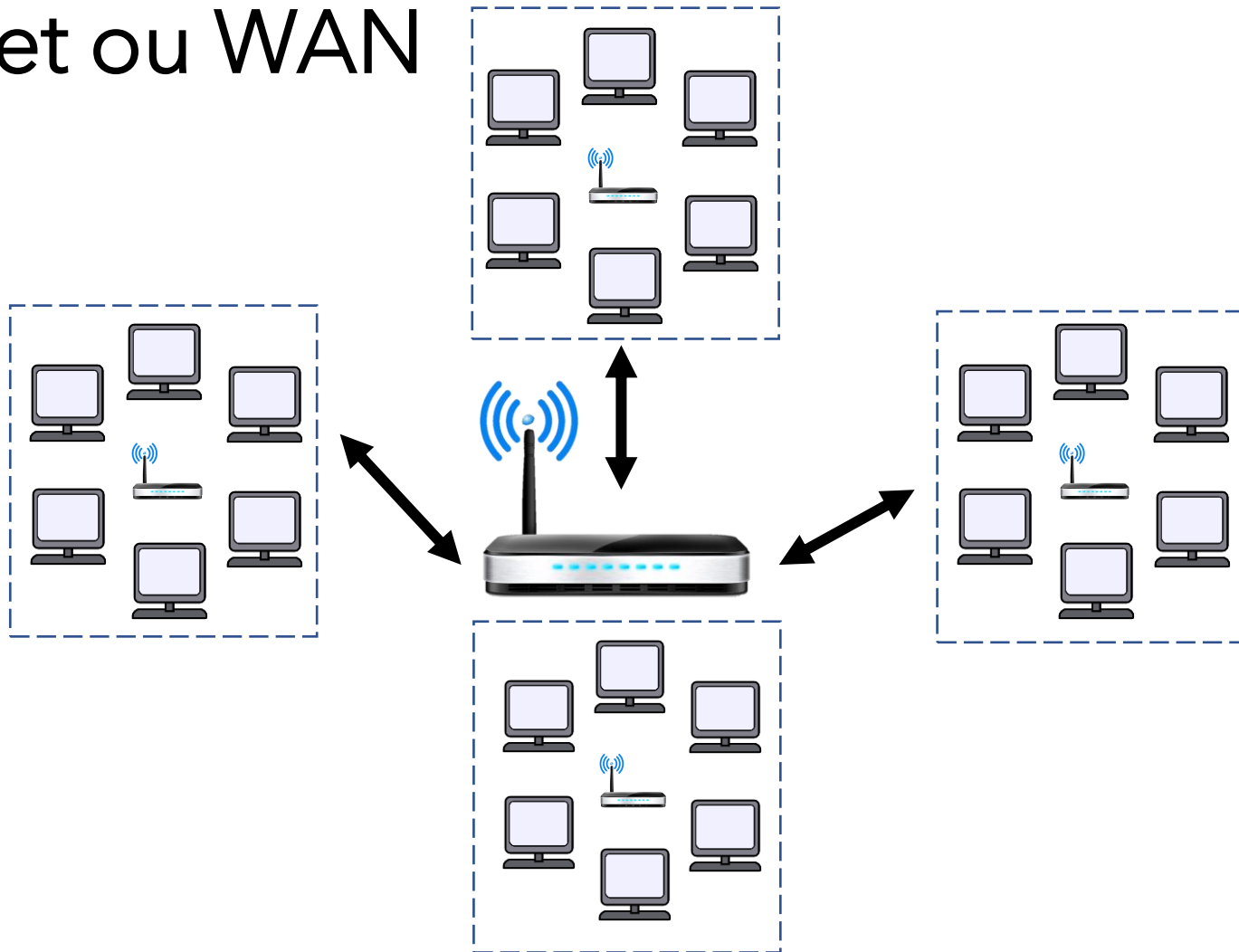




E a internet? A rede de redes?

A LAN só permite que os equipamentos comuniquem localmente. A internet permite qualquer dispositivo conversar com qualquer outro conectado na rede

Internet ou WAN





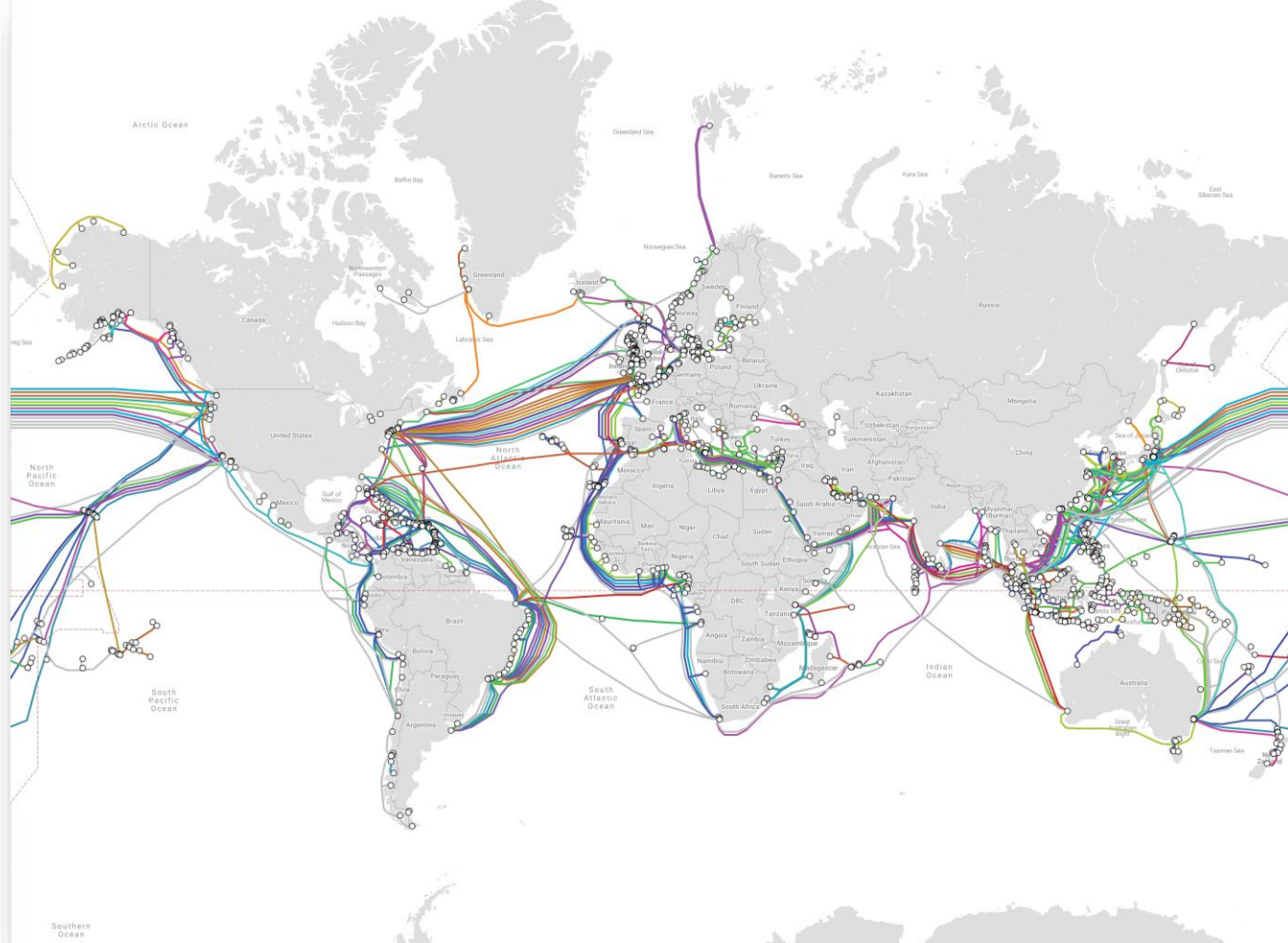
Endereço IP, Endereço MAC, Roteador?

Para nosso curso não precisamos nos preocupar com esses detalhes! O equipamento e termo "roteador" é o suficiente.



Como as redes se
conectam globalmente?

Temos grandes corporações responsáveis
pelos equipamentos e cabos submarinos





O que fazer agora com
todo mundo conectado?

As possibilidades são ilimitadas!



- ☐ Conectar no SIGAA e baixar o material da disciplina de Web.
- ☐ Entregar os exercícios dentro do prazo



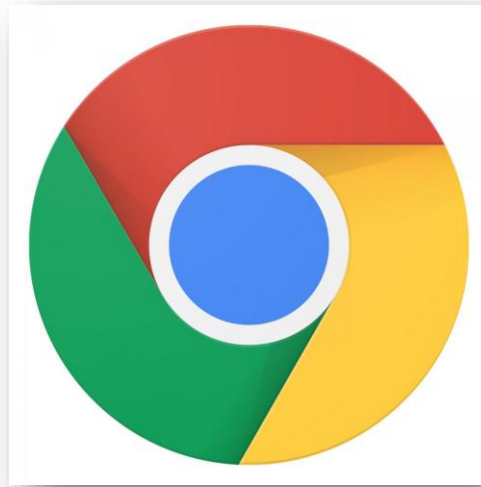
- ☐ Conectar no Spotify e ouvir suas músicas favoritas.
- ☐ Descobrir músicas novas no "mix diário"



- ☐ Acessar serviços de e-mail
- ☐ Limpar a caixa de spam



- ❑ Acessar serviços de vendas de jogos digitais, como a *steam*.
- ❑ Jogar com colegas em outros países



☐ Utilizar um navegador para requisitar páginas *web*.



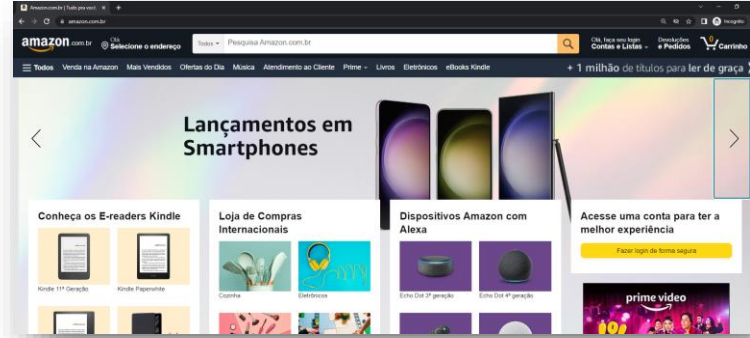
Web



O que é *web*?

Coleção de recursos (textos, imagens, vídeos) que são trafegados pela *internet* e renderizados em navegadores como uma *página web*.

URL



- ❑ Os recursos são localizados através de URL (*Uniform Resource Locator*)
- ❑ Exemplo: www.amazon.com.br (URL)

Recursos ficam armazenados nos servidores



- ❑ Esses recursos trafegam pela internet e são exibidos por navegadores.
- ❑ Navegadores são chamados também de "web browser", -> "navegar na web"

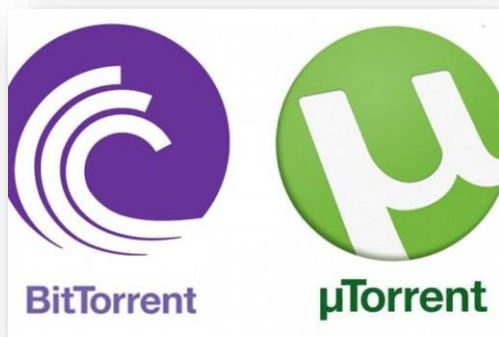
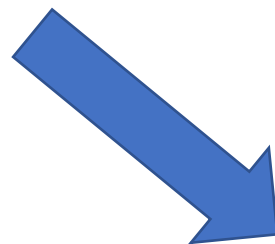
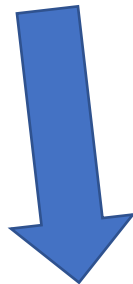
Internet



Web

- ❑ A “web” é um dos tipos de serviços que trafegam pela Internet.
- ❑ A internet é a estrutura física de equipamentos e conexões

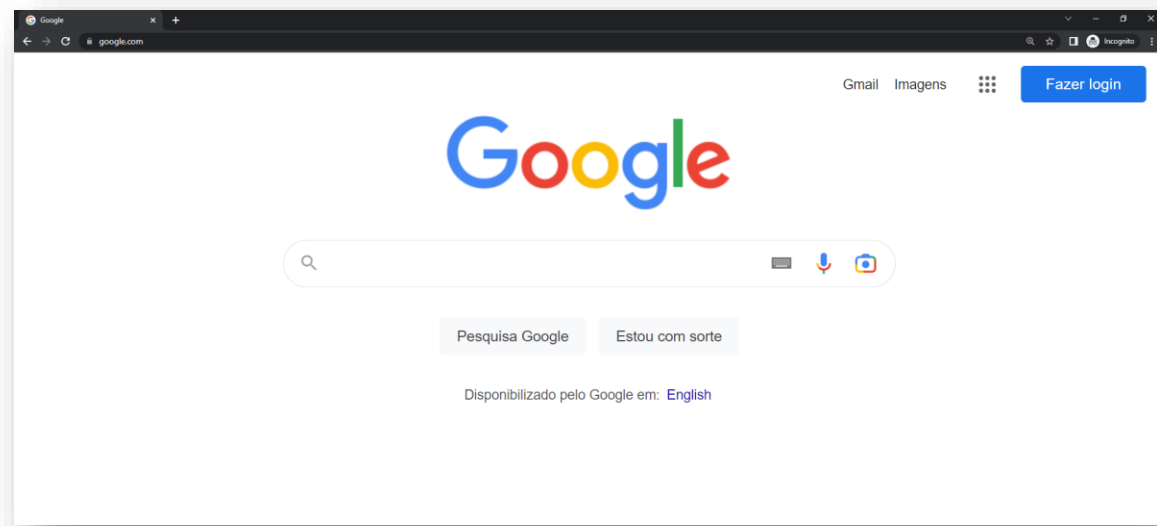
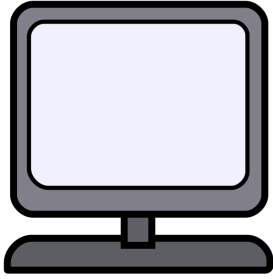
Internet





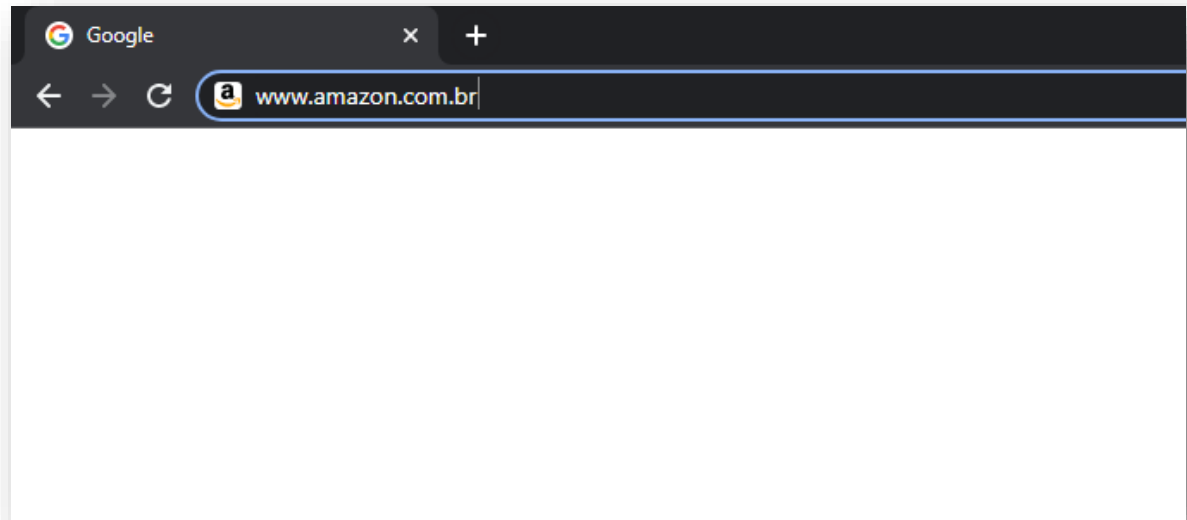
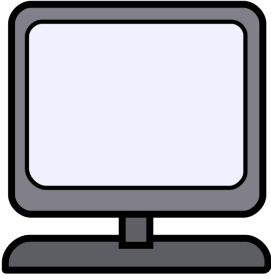
Requisição e Resposta

CLIENTE



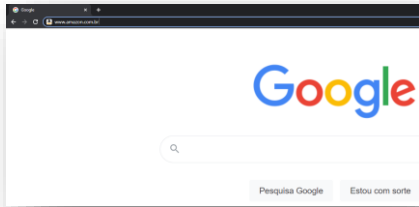
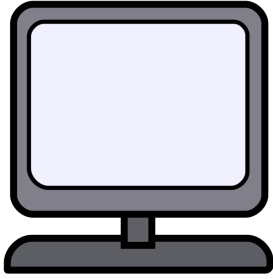
- ☐ Executa a ação de abrir o navegador
- ☐ Deseja requisitar uma página

CLIENTE



- ☐ No campo de busca do navegador digita o endereço:
www.amazon.com.br
- ☐ Ao pressionar *enter* uma requisição será enviada

CLIENTE



→
requisição



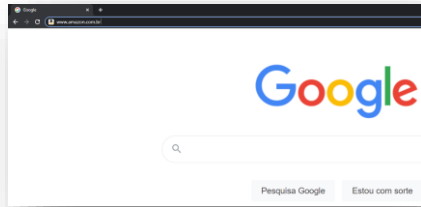
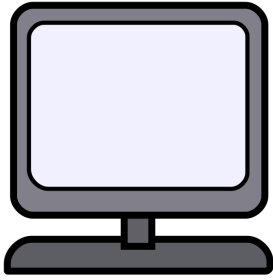
→
requisição

SERVIDOR



- ❑ A **requisição**, através da Internet, chegará no computador de destino.
- ❑ Este computador tem o nome de “servidor” ou “servidor web”

CLIENTE



→
requisição



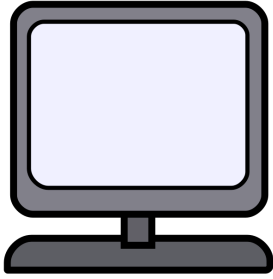
→
requisição

SERVIDOR



- ❑ A **requisição** contém um pedido com a página desejada.
- ❑ “Olá Sr. Servidor, tudo bem? Pode me devolver a página www.amazon.com.br?”

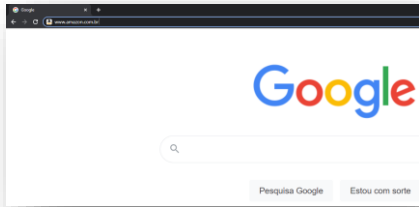
CLIENTE



requisição

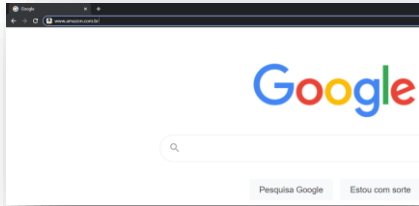
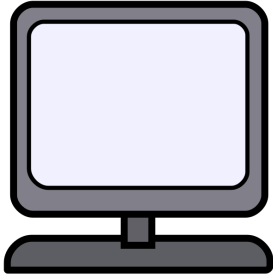


requisição



- ❑ A partir desse momento o servidor irá processar a requisição e irá devolver uma resposta para o cliente.

CLIENTE

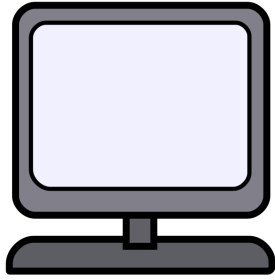


SERVIDOR

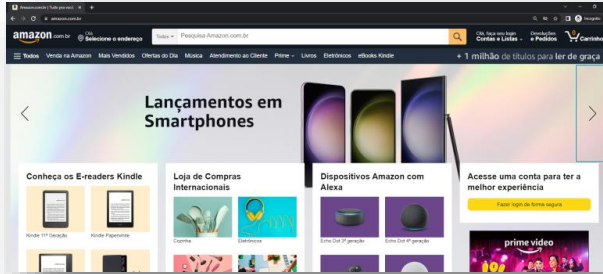


❑ A resposta é uma página web e um código indicando o *status*

CLIENTE

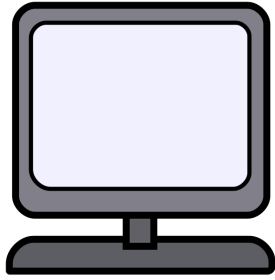


SERVIDOR



- ❑ Se a requisição for processada corretamente o servidor devolverá, como reposta, a página web da Amazon e um código 200.

CLIENTE



SERVIDOR

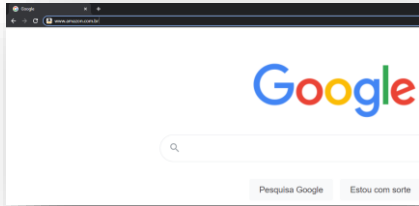
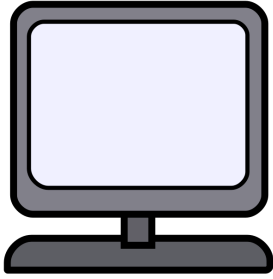


- ☐ Porém diversos erros podem ocorrer. Se a página solicitada não existir é retornada uma outra página informando o erro
- ☐ O famoso código 404 é retornado

Pra fechar!



CLIENTE



requisição



www.amazon.com.br



requisição



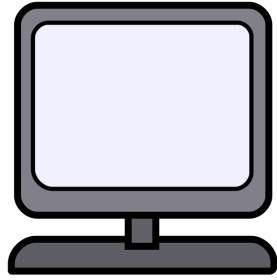
www.amazon.com.br

SERVIDOR



- ❑ Cliente envia uma requisição, que será roteada pela internet

CLIENTE

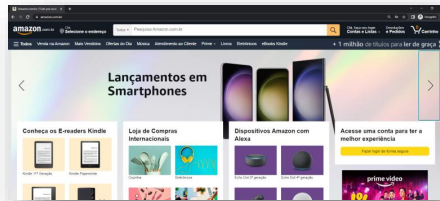


requisição

www.amazon.com.br



resposta



SERVIDOR



requisição

www.amazon.com.br



resposta

- ❑ Cliente envia uma requisição, que será roteada pela internet
- ❑ Servidor processa a requisição e devolve uma resposta em formato de página web.

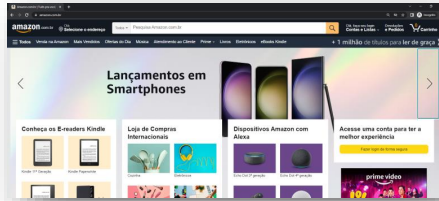
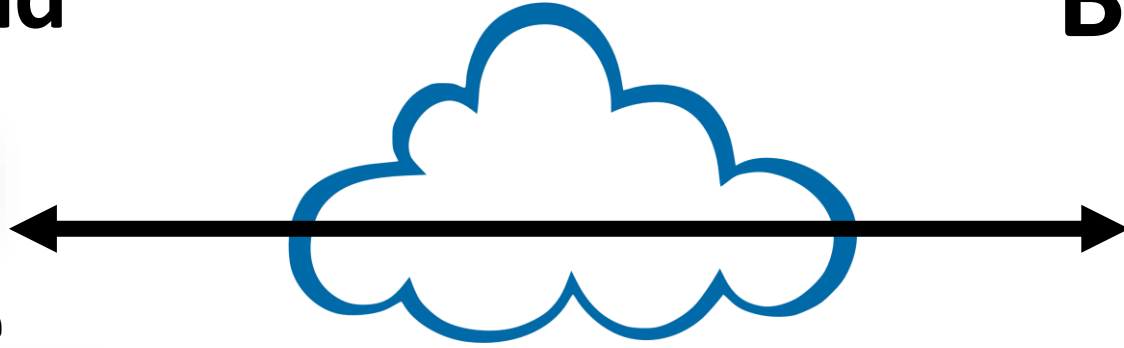


Frontend e Backend

Frontend



Backend

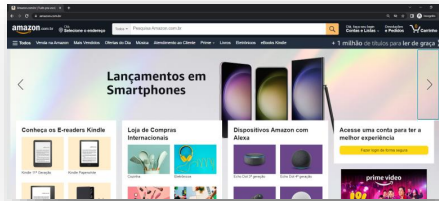


- ❑ *Frontend* é a parte da web renderizada para o cliente
- ❑ *Backend* é a parte da web que executa do lado do servidor

Frontend



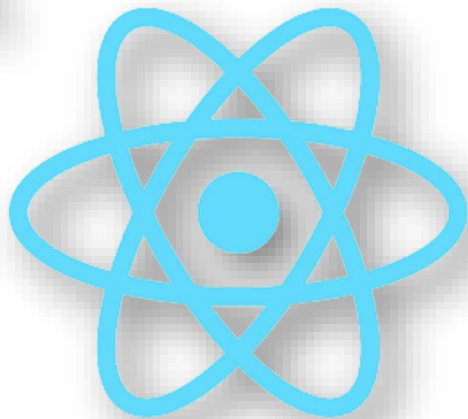
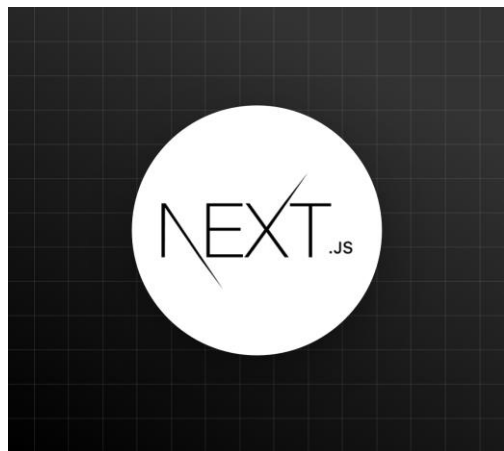
Backend



- ☐ A equipe responsável pelo frontend cuida de toda a interação/ações e lógica da página web visualizada na máquina cliente. Não necessariamente a estética.
- ☐ Eventos, submissão de formulários, ações



- ☐ A principal linguagem de programação utilizada no frontend é o JavaScript.
- ☐ Recentemente o TypeScript vem ganhando espaço. Mas o código final é convertido para JavaScript antes de ser interpretado pelo navegador.

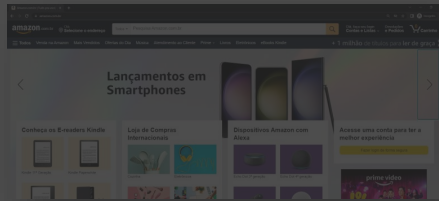


React



❑ Alguns frameworks/bibliotecas auxiliam esse desenvolvimento

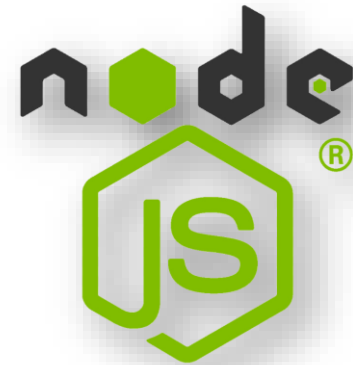
Frontend



Backend



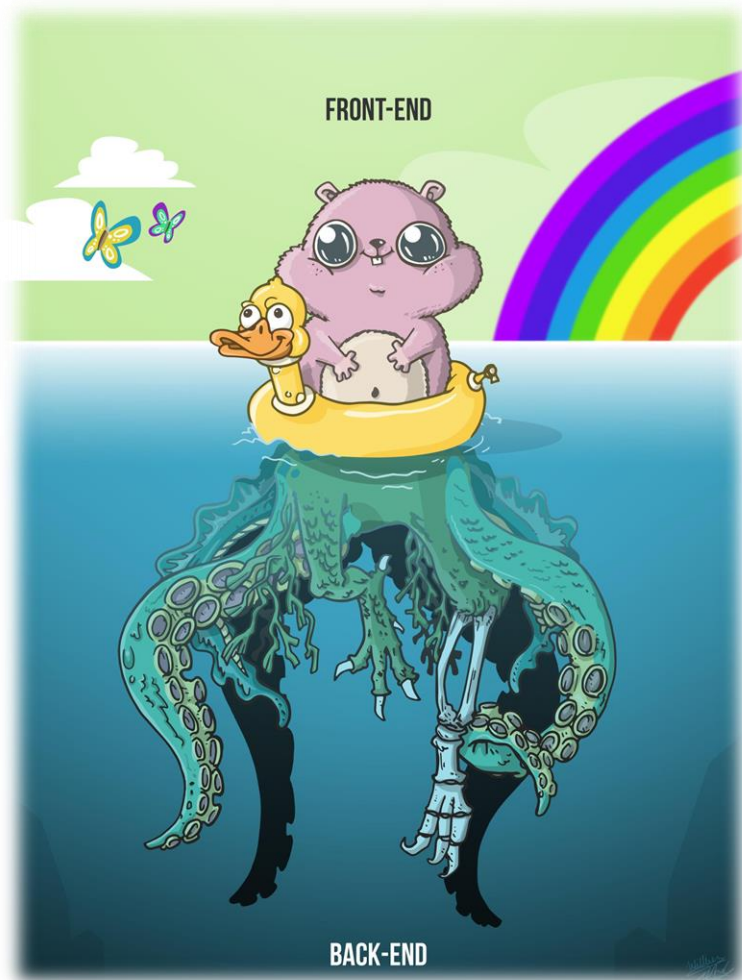
- ☐ A equipe responsável pelo backend cuida de todo o código que executa do lado do servidor.
- ☐ Acesso ao banco de dados, autenticação, autorização, processamento



- ☐ Diferente do frontend, existem diversas tecnologias e linguagens que podem ser utilizadas no backend, ou lado do servidor.

Pra fechar!





HTML



CSS



JS



EU
TOMEI JS – Verbo
CAFÉ HTML - Substantivo
EXPRESSO CSS - Adjetivo



HTML



CSS



JS



HTML – HyperText Markup Language

CAFÉ HTML - Substantivo

HTML – HyperText Markup Language

□ HyperText

- Remete a ideia de um texto que “*linkável*”. Isto é, o texto pode ser um caminho para outro elemento. O objetivo inicial era navegar por páginas através desses “*hiper-links*”

HTML – HyperText Markup Language

❑ *Markup Language*

- ❑ Significa marcação. Aqui entra a argumentação que HTML não é linguagem de programação e sim **marcação**. A marcação é informar o que é aquele elemento. Por exemplo, podemos marcar um texto dizendo que é um botão, ou um cabeçalho.
- ❑ A marcação é feita através de etiquetas (*tag*) HTML
- ❑ Temos *tag* específicas para botões, imagens, cabeçalhos, formulários, tabelas.
- ❑ Temos *tag* para organização/agrupamento
- ❑ Temos para semântica: Barra de navegação, rodapé, seção
 - ❑ Essenciais para acessibilidade

HTML – HyperText Markup Language

❑ *Language*

- ❑ Concluindo que HTML é uma “Linguagem para a marcação de hipertexto”
- ❑ Sua extensão é “.html”

HTML



CSS



JS



CSS – Cascading Style Sheet

❑ *Cascading*

- ❑ Em cascata. Significa que a estilização é aplicada “em cascata”.
- ❑ Veremos que é possível de definir um “identificador” para elementos HTML.
- ❑ A estilização pode ser aplicada para elementos de determinada *tag* e/ou para elementos com um identificador específico.
- ❑ Se o elemento possuir a *tag* e o identificador a estilização será aplicada “em cascata”.
- ❑ Mas devemos tomar cuidado com casos concorrentes, que veremos ao longo do curso.

CSS – Cascading Style Sheet

❑ *Style Sheet*

- ❑ Folhas de estilização. As “folhas de estilização” podem ser vistas como os arquivos “.css” que contém as regras de estilização
- ❑ Essas regras podem definir diversos aspectos dos elementos HTML, tais como:
 - ❑ Cor, fonte, margem, posição, animação, ..., etc.

HTML



CSS



JS



JavaScript (JS)

- ❑ Linguagem de programação multiparadigma que contém o comportamento (ações/verbos) de uma página com elementos HTML
- ❑ Possui sintaxe e comandos presentes em outras linguagens tais como:
 - ❑ if-else, for, switch, variáveis, constantes, expressões lambda, etc

JavaScript (JS)

- ❑ Navegadores são capazes de interpretar os comandos descrito por código JS
- ❑ Utilizando o NodeJS é possível executar código JS fora do navegador.

JavaScript (JS)

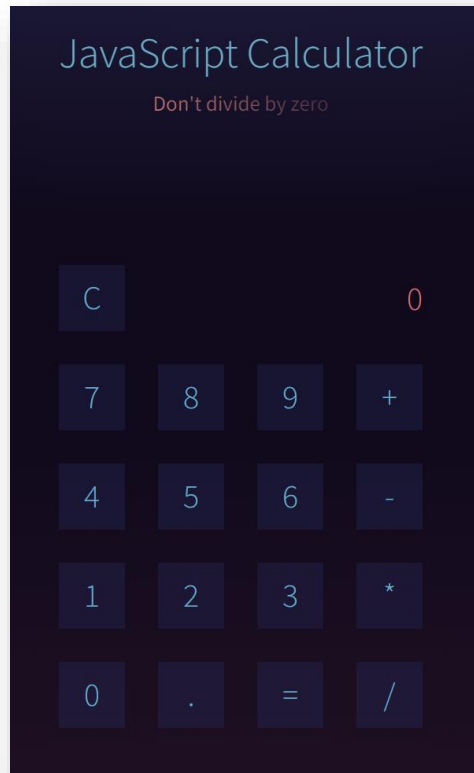
- ❑ Apesar de não ter nenhuma relação com a linguagem Java, seu nome foi pensando para fins comerciais.
- ❑ Na época, a linguagem Java estava popular

Mão na Massa – HTML/CSS/JS

❑ Acesse o link:

<https://codepen.io/giana/pen/GJMBEv>

❑ É uma calculadora escrita em HTML/CSS/JS



JavaScript calculatorGiana 700 0 Follow

HTML

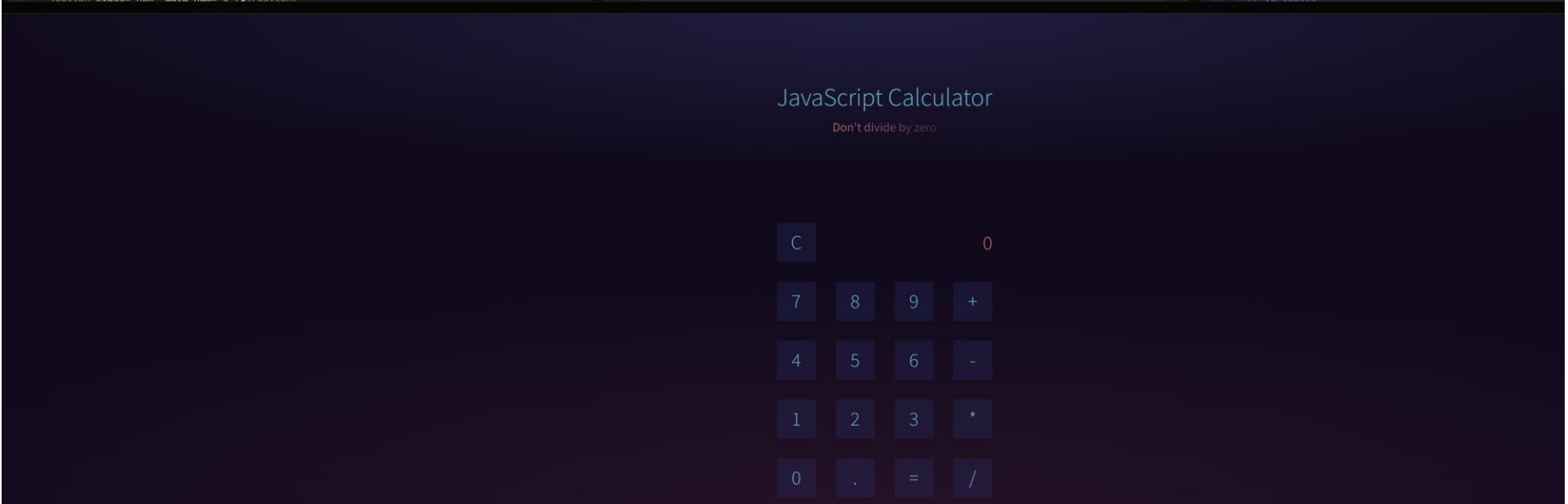
CSS (SCSS)

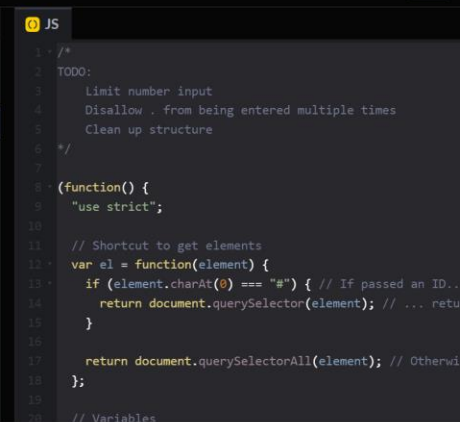
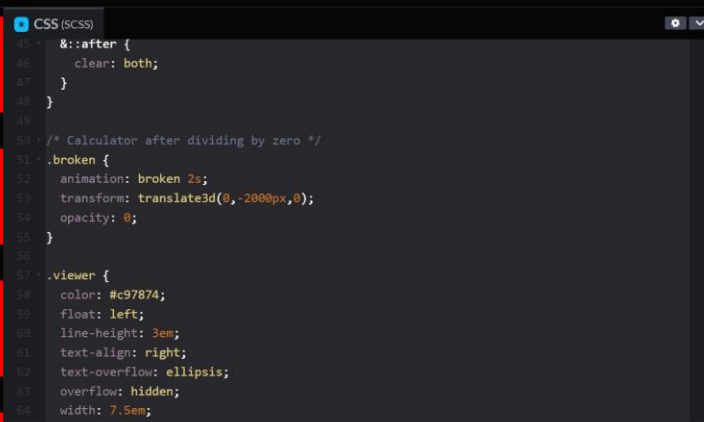
JS

```
1 <h1>JavaScript Calculator</h1>
2 <p class="warning">Don't divide by zero</p>
3
4 <div id="calculator" class="calculator">
5
6   <button id="clear" class="clear">C</button>
7
8   <div id="viewer" class="viewer">0</div>
9
10  <button class="num" data-num="7">7</button>
11  <button class="num" data-num="8">8</button>
12  <button class="num" data-num="9">9</button>
13  <button data-ops="plus" class="ops">+</button>
14
15  <button class="num" data-num="4">4</button>
16  <button class="num" data-num="5">5</button>
17  <button class="num" data-num="6">6</button>
18  <button data-ops="minus" class="ops">-</button>
19
20  <button class="num" data-num="1">1</button>
```

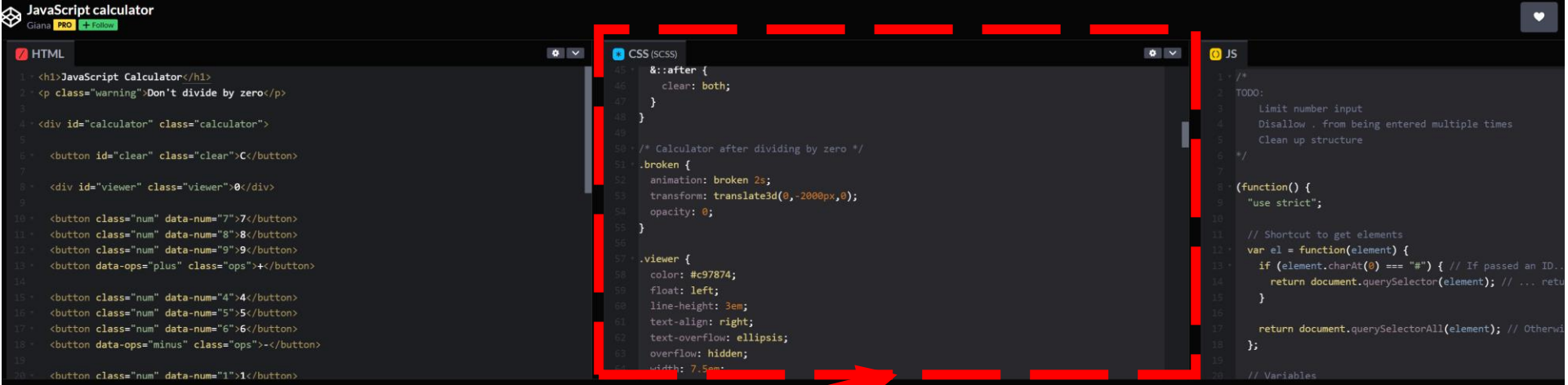
```
45 &::after {
46   clear: both;
47 }
48 }
49
50 /* Calculator after dividing by zero */
51 .broken {
52   animation: broken 2s;
53   transform: translate3d(0,-200px,0);
54   opacity: 0;
55 }
56
57 .viewer {
58   color: #c97874;
59   float: left;
60   line-height: 3em;
61   text-align: right;
62   text-overflow: ellipsis;
63   overflow: hidden;
64   width: 7.5em;
```

```
1 /*
2 TODO:
3   Limit number input
4   Disallow . from being entered multiple times
5   Clean up structure
6 */
7
8 (function() {
9   "use strict";
10
11 // Shortcut to get elements
12 var el = function(element) {
13   if (element.charAt(0) === "#") { // If passed an ID..
14     return document.querySelector(element); // ... return
15   }
16
17   return document.querySelectorAll(element); // Otherwise
18 };
19
20 // Variables
```





- ❑ Nessa aba temos o código HTML
- ❑ Determina o que temos na tela
- ❑ A maioria são botões nesse exemplo

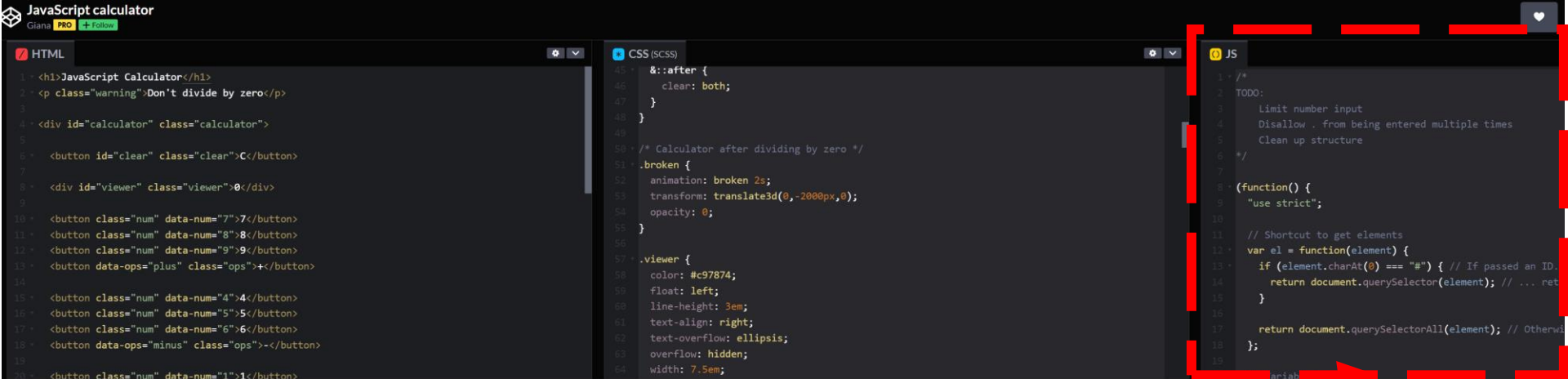


- ☐ A segunda aba é o CSS
- ☐ Ela determina a estilização dos elementos.
- ☐ Cores, posição, animação, opacidade, etc.

JavaScript Calculator

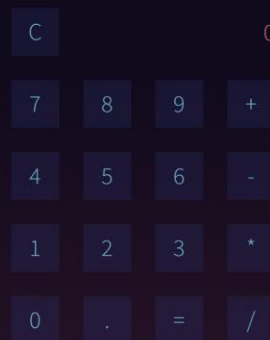
Don't divide by zero

C	0		
7	8	9	+
4	5	6	-
1	2	3	*
0	.	=	/



JavaScript Calculator

Don't divide by zero



- ☐ A terceira aba é o JavaScript(JS)
- ☐ Ela determina as ações que podemos executar na página
- ☐ Nesse exemplo as ações são operações aritméticas

JavaScript calculator

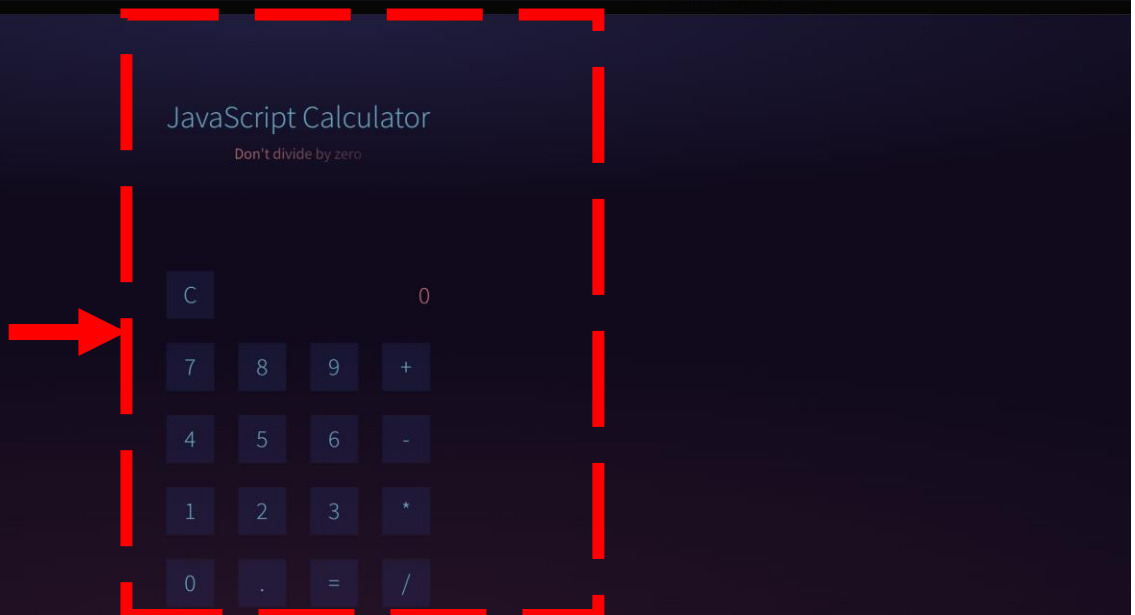
Giana 720 4 Follow

```
HTML
1 <h1>JavaScript Calculator</h1>
2 <p class="warning">Don't divide by zero</p>
3
4 <div id="calculator" class="calculator">
5
6   <button id="clear" class="clear">C</button>
7
8   <div id="viewer" class="viewer">0</div>
9
10  <button class="num" data-num="7">7</button>
11  <button class="num" data-num="8">8</button>
12  <button class="num" data-num="9">9</button>
13  <button data-ops="plus" class="ops">+</button>
14
15  <button class="num" data-num="4">4</button>
16  <button class="num" data-num="5">5</button>
17  <button class="num" data-num="6">6</button>
18  <button data-ops="minus" class="ops">-</button>
19
20  <button class="num" data-num="1">1</button>
```

```
CSS (SCSS)
45 &::after {
46   clear: both;
47 }
48
49
50 /* Calculator after dividing by zero */
51 .broken {
52   animation: broken 2s;
53   transform: translate3d(0,-200px,0);
54   opacity: 0;
55 }
56
57 .viewer {
58   color: #c97874;
59   float: left;
60   line-height: 3em;
61   text-align: right;
62   text-overflow: ellipsis;
63   overflow: hidden;
64   width: 7.5em;
```

```
JS
1 /*
2 TODO:
3   Limit number input
4   Disallow . from being entered multiple times
5   Clean up structure
6 */
7
8 (function() {
9   "use strict";
10
11   // Shortcut to get elements
12   var el = function(element) {
13     if (element.charAt(0) === "#") { // If passed an ID...
14       return document.querySelector(element); // ...return
15     }
16
17     return document.querySelectorAll(element); // Otherwise
18   };
19
20   // Variables
```

- ❑ Nessa área temos o simulador de um navegador
- ❑ O resultado da interpretação dos códigos HTML/CSS/JS



Removendo CSS e JS

- ❑ Vamos entender melhor o HTML e o que ele representa na página.
- ❑ Para isso vamos remover o código CSS e JS.
- ❑ Não se preocupe se achar o código confuso. Temos o semestre inteiro ainda.



JavaScript Calculator

Don't divide by zero

C

0

7 8 9 + 4 5 6 - 1 2 3 * 0 . = /

Reset Universe?

- ☐ Observe os códigos CSS e JS comentados.
- ☐ Cada linha possui // (duas barras) no início

The screenshot shows a code editor with three panels: HTML, CSS (SCSS), and JS. The HTML panel contains the structure of a calculator with buttons for numbers, operations, and a display. The CSS panel contains styling rules for the calculator's layout and elements. The JS panel contains the logic for the calculator's functionality, including input validation and calculations.

```
HTML
1 <h1>JavaScript Calculator</h1>
2 <p class="warning">Don't divide by zero</p>
3
4 <div id="calculator" class="calculator">
5
6   <button id="clear" class="clear">C</button>
7
8   <div id="viewer" class="viewer">0</div>
9
10  <button class="num" data-num="7">7</button>
11  <button class="num" data-num="8">8</button>
12  <button class="num" data-num="9">9</button>
13  <button data-ops="plus" class="ops">+</button>
14
15  <button class="num" data-num="4">4</button>
16  <button class="num" data-num="5">5</button>
17  <button class="num" data-num="6">6</button>
18  <button data-ops="minus" class="ops">-</button>
19
20  <button class="num" data-num="1">1</button>

CSS (SCSS)
45 // &:after {
46 //   clear: both;
47 // }
48 // }
49
50 // /* Calculator after dividing by zero */
51 // .broken {
52 //   animation: broken 2s;
53 //   transform: translate3d(0,-200px,0);
54 //   opacity: 0;
55 // }
56
57 // .viewer {
58 //   color: #c97874;
59 //   float: left;
60 //   line-height: 3em;
61 //   text-align: right;
62 //   text-overflow: ellipsis;
63 //   overflow: hidden;
64 //   width: 7.5em;

JS
1 // /*
2 // TODO:
3 //   Limit number input
4 //   Disallow . from being entered multiple times
5 //   Clean up structure
6 // */
7
8 // (function() {
9 //   "use strict";
10
11 //   // Shortcut to get elements
12 //   var el = function(element) {
13 //     if (element.charAt(0) === "#") { // If passed an ID...
14 //       return document.querySelector(element); // ... returns single element
15 //     }
16
17 //     return document.querySelectorAll(element); // Otherwise, returns a nodelist
18 //   };
19
20 //   // Variables
```

JavaScript Calculator

Don't divide by zero

0
7 8 9 + 4 5 6 - 1 2 3 * 0 . = /
Reset Universe?

- ☐ O resultado final é uma página HTML pura.
- ☐ Os elementos continuam existindo
- ☐ São botões e textos
- ☐ Não possuem nenhuma estilização (CSS)
- ☐ Não executam nenhuma ação (JS)

Adicionando o CSS

- ❑ Vamos voltar com o código CSS
- ❑ Basta repetir os mesmos passos usados pra "comentar"
- ❑ Ao selecionar um código comentado e repetir os passos o código será descomentado.
- ❑ Faça isso apenas para o CSS (segunda aba)



HTML

```
1 <h1>JavaScript Calculator</h1>
2 <p class="warning">Don't divide by zero</p>
3
4 <div id="calculator" class="calculator">
5
6 <button id="clear" class="clear">C</button>
7
8 <div id="viewer" class="viewer">0</div>
9
10 <button class="num" data-num="7">7</button>
11 <button class="num" data-num="8">8</button>
12 <button class="num" data-num="9">9</button>
13 <button data-ops="plus" class="ops">+</button>
14
15 <button class="num" data-num="4">4</button>
16 <button class="num" data-num="5">5</button>
17 <button class="num" data-num="6">6</button>
18 <button data-ops="minus" class="ops">-</button>
19
20 <button class="num" data-num="1">1</button>
```

CSS (SCSS)

```
45 &::after {
46   clear: both;
47 }
48
49
50 /* Calculator after dividing by zero */
51 .broken {
52   animation: broken 2s;
53   transform: translate3d(0,-200px,0);
54   opacity: 0;
55 }
56
57 .viewer {
58   color: #c97874;
59   float: left;
60   line-height: 3em;
61   text-align: right;
62   text-overflow: ellipsis;
63   overflow: hidden;
```

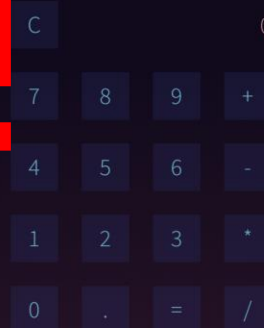
JS

```
1 // ?
2 // TODO:
3 //   Limit number input
4 //   Disallow . from being entered multiple times
5 //   Clean up structure
6 // ?
7
8 // (function() {
9 //   "use strict";
10
11 // // Shortcut to get elements
12 // var el = function(element) {
13 //   if (element.charAt(0) === "#") { // If passed an ID
14 //     return document.querySelector(element); // ... r
15 //   }
16
17 //   return document.querySelectorAll(element); // Other
18 // };
19
20 // // Variables
```

JavaScript Calculator

Don't divide by zero

9



- ☐ O código CSS não está comentado
- ☐ O código JS deverá permanecer comentado
- ☐ Linhas comentadas possuem // (duas barras)

Página com HTML e CSS

- ❑ O resultado consiste dos elementos HTML, isto é, o *que* temos na tela e a estilização fornecida pelo CSS.
- ❑ A página está "agradável" pois agora os elementos estão posicionados adequadamente, com cores e fonte.
- ❑ Mas nada funciona ☹

JavaScript são as ações

- ❑ Como não há código JS, não temos nenhuma ação.
- ❑ O JS que consiste nos “verbos” e permite executarmos ações.
 - ❑ Operações aritméticas
 - ❑ Enviar formulários
 - ❑ Fazer login
 - ❑ Jogar
 - ❑ ...

Voltando com o JavaScript

- ❑ Agora descomente o código JS e verá que a página voltará a funcionar normalmente.
- ❑ Assim, temos:
 - ❑ Elementos na página (botões) : HTML
 - ❑ Estilização (cores, fonte, posicionamento): CSS
 - ❑ Ações (operações aritméticas): JS

Dúvidas?





Aula – 2

Introdução a Web

Disciplina: XDES03 – Programação Web

Prof: Phyllipe Lima Francisco
phyllipe@unifei.edu.br

Universidade Federal de Itajubá – UNIFEI
IMC – Instituto de Matemática e Computação