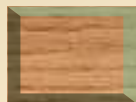


**Introdução**



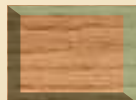
**React**



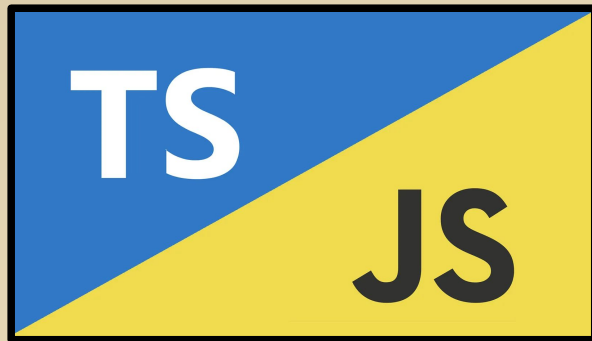
**HTML**



**CSS**



**Tableless**



**HTML e CSS**

Prof. Enzo  
Seraphim

Profa. Bárbara  
Pimenta Caetano



# Agenda

## Introdução

- ✓ Conceitos HTML e CSS
- ✓ Exemplificar html e css na arquitetura do React
- ✓ Linguagem JavaScript
- ✓ Ambiente Reat e os conceitos básico





# Introdução

## Ambiente

```
//mudar diretorio  
cd users\aluno\vscode  
//criar aplicação reatapp  
npx create-react-app pblic01-aula16
```

```
//abrir pasta no vscode  
c:\users\aluno\vscode\pblic01-aula16  
//iniciar o servidor react  
npm start  
//abrir browser  
http://localhost:3000
```



# Ambiente

```
▼ PBLC01-AULA16
  > node_modules
  > public
  ▼ src
    # App.css
    JS App.js
    JS App.test.js
    # index.css
    JS index.js
    logo.svg
    JS reportWebVitals.js
    JS setupTests.js
    .gitignore
    {} package-lock.json
    {} package.json
    ⓘ README.md
```

```
{
  "name": "pblc01-aula16",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/dom": "^10.4.0",
    "@testing-library/jest-dom": "^6.6.3",
    "@testing-library/react": "^16.3.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^19.1.0",
    "react-dom": "^19.1.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
}
```



## Introdução

# Marcação com JSX

- Permite escrever marcação semelhante a HTML dentro de um arquivo JavaScript.
- Antigamente, se separava: conteúdo em HTML; design em CSS; lógica em JavaScript
- A maior interativa fez com que a lógica determinava cada vez mais o conteúdo.
- JavaScript estava no comando do HTML!
- A lógica de renderização e a marcação vivem juntas no mesmo lugar : componentes React



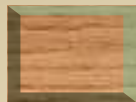


## Introdução

# Componente React

- É uma função JavaScript que pode conter alguma marcação que o React renderiza no navegador.
- Usam uma extensão de sintaxe chamada JSX para representar essa marcação.
- JSX se parece muito com HTML, mas é um pouco mais rigoroso e pode exibir informações dinâmicas.

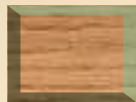




**Introdução**



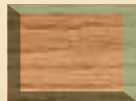
**React**



**HTML**



**CSS**



**Tableless**



**HTML e CSS**

Prof. Enzo  
Seraphim

Profa. Bárbara  
Pimenta Caetano



# O que é React?

React

- ✓ Framework JavaScript criado pelo Facebook (atual Meta) que é declarativo, eficiente e flexível, usado para criar interfaces de usuário (UI) em aplicativos web
- ✓ Conhecido por sua abordagem de componentes, que permite criar UI complexas a partir de pequenos blocos de código reutilizáveis e isolados







# Componente

React

- ✓ Componentes permitem dividir a UI em partes independentes, reutilizáveis para pensar em cada parte isoladamente.
- ✓ São como funções (ou classes) JavaScript que aceitam entradas arbitrárias (chamadas props) e retornam elementos React que descrevem o que deve aparecer na tela.
- ✓ Definido em uma Função ou em uma Classe





React

# Componente como Função

```
import './App.css';  
function App(props) {  
  return (<h1>Ola, {props.nome}</h1>);  
}  
export default App;
```

**App.js**





React

# Componente como Classe

```
import './App.css';
import React from 'react';
class App extends React.Component {
  render(){
    return (<h1>Ola, {this.props.nome}</h1>);
  }
}
export default App;
```

**App.js**





# Importando/Exportando Componentes

React

- ✓ A reutilização é baseada no fato que um componente pode ser composto por outros
- ✓ Cada componente é armazenado em um arquivo
- ✓ Exportar é a prática de tornar o componente de um módulo (arquivo) disponível para uso em outros módulos → **export**
- ✓ Importar é a prática de usar um componente que está em outro arquivo → **import**





# Importando/Exportando Componentes

React

- ✓ Instanciar significa criar uma nova instância desse componente: uma nova cópia com seu próprio estado e comportamento.
- ✓ Garante que cada componente tenha seu próprio espaço na memória e que não haja interferência entre diferentes instâncias do mesmo componente.





React

# Exportando Componente

```
import './App.css';  
import React from 'react';  
class App extends React.Component {  
  render(){  
    return (<h1>Ola, {this.props.nome}</h1>);  
  }  
}  
  
export default App;
```

App.js



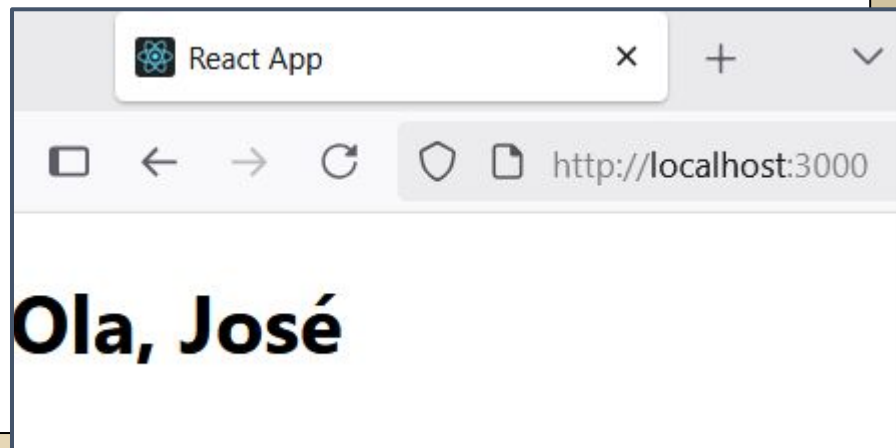


React

# Importando e Instanciando

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
const root = ReactDOM.createRoot(
  document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App nome="José"/>
  </React.StrictMode>
);
```

index.js





# JavaScript em JSX com chaves

React

- ✓ Chaves são utilizadas para inserir expressões JavaScript diretamente dentro do HTML ou de atributos de tags.
- ✓ Permitem incorporar variáveis, funções e outras expressões JavaScript no código JSX, convertendo-o para valores que podem ser renderizados no navegador
- ✓ Chaves são essenciais para mantendo a lógica de renderização e o conteúdo no mesmo lugar.







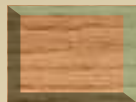
React

# JavaScript em JSX com chaves

```
import React from 'react';
const pessoa= {
  nome: "José",
};
class App extends React.Component {
  render(){
    return (<h1>Ola, {pessoa.nome}</h1>); }}
export default App;
```

App.js

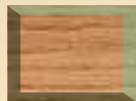




**Introdução**



**React**



**HTML**



**CSS**



**Tableless**



**HTML e CSS**

Prof. Enzo  
Seraphim

Profa. Bárbara  
Pimenta Caetano



# O que é HTML?

HTML

- ✓ HTML (HyperText Markup Language) é uma linguagem de marcação que estrutura e organiza conteúdo para páginas da web.
- ✓ Exibir conteúdo em websites, utilizando tags para definir elementos como texto, imagens, links, entre outros
- ✓ Usa tags (palavras-chave entre "<" e ">") para identificar/criar diferentes elementos de conteúdo e suas características.



# Partes do Elemento

- ✓ **Tag de abertura** → Nome do elemento envolvido entre símbolos ‘<’ e ‘>’
- ✓ **Atributo da tag** → Característica do elemento em que são atribuídos valor entre aspas duplas ""
- ✓ **Conteúdo** → Texto que está entre o tag de abertura e de fechamento.
- ✓ **Tag de fechamento** → Idem tag de abertura, com inclusão da barra ‘/’ antes do nome do elemento.
- ✓ **Exemplo:**

```
<a href="unifei.edu.br">UNIFEI</a>
```



# Principais Elementos



## Estrutura

- `<html>`, `<head>`, `<body>`



## Layout

- `<header>`, `<main>`, `<footer>`
- `<section>`, `<article>`, `<nav>`, `<aside>`



## Conteúdo

- `<h1>` - `<h6>`, `<p>`, `<strong>`, `<em>`, `<span>`, `<div>`



## Mídia & Links

- `<a>`, `<img>`, `<video>`, `<iframe>`



## Listas & Tabelas

- `<ul>`, `<ol>`, `<li>`, `<table>`, `<tr>`, `<td>`

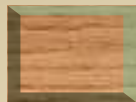


## Formulários

- `<form>`, `<input>`, `<textarea>`, `<button>`

HTML





**Introdução**



**React**



**HTML**



**CSS**



**Tableless**



**HTML e CSS**

Prof. Enzo  
Seraphim

Profa. Bárbara  
Pimenta Caetano



# O que é CSS?

- ✓ Acrônimo de Cascading Style Sheets → Folhas de Estilo em Cascata usada para definir o layout de HTML, XML e XHTML;



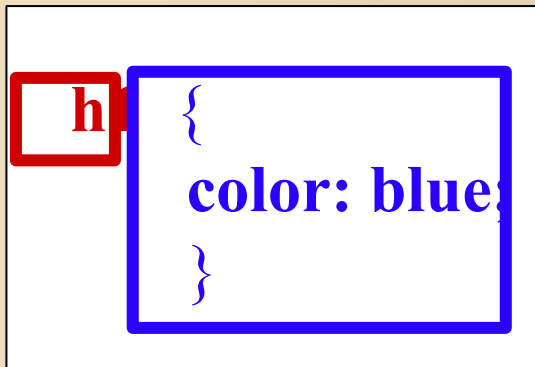
- ✓ Separa a definição das tags (HTML) da definição de formatação/layout (CSS);
- ✓ Controlar o estilo/layout de múltiplas páginas
- ✓ Contém informação do layout (cores, fontes, posicionamento, tamanhos, margens, etc.);



# Seletor e bloco declaração

- ✓ Sintaxe simples baseada em um conjunto de regras de estilo
- ✓ Tem um seletor e um bloco, propriedade e valor

CSS

**Seletor** →← **Bloco  
Declaração**

Elementos `<h1>` no HTML foi estilizado com a cor azul



# Propriedades

## Propriedades

- ✓ Propriedades para background
- ✓ Propriedades para borda
- ✓ Propriedades para contorno
- ✓ Propriedades para margem
- ✓ Propriedades para padding
- ✓ Propriedades para largura e altura
- ✓ Propriedades para textos
- ✓ Propriedades para listas
- ✓ Propriedades para layout
- ✓ Propriedades para transparência



# Folha de Estilo em Linha

App.js

```
import React from 'react';
class App extends React.Component {
  render(){
    const mystyle = {
      color: "blue",
    };
    return (<h1 style={mystyle}>Ola, {this.props.nome}</h1>); }}
export default App;
```

html puro

```
<html><body>
<h1 style="color:blue;">Ola, José</h1>
</body></html>
```





CSS

# Folha de Estilo Externa

App.js

```
import './App.css';
import React from 'react';
class App extends React.Component {
  render(){
    return (<h1>Ola, {this.props.nome}</h1>); }}
export default App;
```

App.css

```
h1 {
  color: blue; /*comentário*/
}
```

html puro

```
<html><head>
<link rel="stylesheet" href="App.css" type="text/css">
</head>
<body><h1>Ola, José</h1></body>
</html>
```





CSS

# Tipos de Seletores

- 1) Elementos do tipo
- 2) Elementos da classe
- 3) Elementos com id
- 4) Elementos com atributo
- 5) Pseudoclasses
- 6) Pseudoelementos



# 1 – Elementos do tipo

- ✓ Especifica os elementos HTML (tags) sobre o quais se devem aplicar o efeito.

- ✓ Exemplo:

```
table { ... }
```

- ✓ Aplicará o efeito a todas as tags do tipo table.

```
<table> </table>
```



## 2 – Elementos da classe

- ✓ Especifica a classe dos elementos sobre os quais se devem aplicar o efeito. Ideal para aplicar um efeito sobre vários elemento na mesma página.

- ✓ Exemplo:

```
.dados { ... }
```

- ✓ Aplicará o efeito descrito todas as tags que apresentem a classe 'dados':

```
<table class="dados">
```

- ✓ Pode aplicar a mais de uma classe a uma tag separando-as por espaço.



## 3 – Elementos com id

CSS

- ✓ Especifica o id dos elementos sobre os quais se devem aplicar o efeito.

- ✓ Exemplo:

```
#listagem { ... }
```

- ✓ Aplicará estilo a todas as tags que apresentem o id botao\_ok, exemplo:

```
<table id="listagem">
```

- ✓ Na prática o id seja único para cada elemento.



## 4 – Elementos com atributo

- ✓ Especifica que o efeito deve ser aplicado a elementos que declarem aquele atributo.
- ✓ Exemplo1:

```
[name] { ... }
```

- ✓ Aplicará o efeito descrito entre chaves a todas as tags que apresentem o atributo name

```
[type="text"] { ... }
```

- ✓ Aplicará o estilo descrito entre chaves a todas as tags input que apresentem o atributo type="text".





## 5 – Pseudoclasses (:)

CSS

- ✓ Uma pseudoclassee é usada para definir a formatação para um estado especial de um elemento.
- ✓ Para um elemento quando o cursor do mouse passa sobre ele (hover);
- ✓ Para hyperlinks não visitados (link) ou já visitados (visited);
- ✓ Para um elemento que recebeu o foco (focus).



## 5 – Pseudoclasses (:) – Exemplo

CSS

- ✓ **a:link** → todos os links que ainda não foram visitados
- ✓ **a:visited** → todos os links já visitados
- ✓ **a:hover** → o link sob o cursor do mouse
- ✓ **a:active** → o link ativo
- ✓ **input:checked** → todo elemento input “checado”



## 6 – Pseudoelementos (::)

- ✓ Um pseudoelemento é usado para definir a formatação para uma parte especificada de um elemento
- ✓ Para a primeira letra (first-letter) de um elemento
- ✓ Para a primeira linha (first-line) de um elemento
- ✓ Para definir conteúdo antes (before) ou depois (after) do conteúdo do elemento;
- ✓ Para a porção do elemento selecionada pelo usuário (selection).

CSS



## 5 – Pseudoelementos (::) – Exemplo

CSS

- ✓ **p::after** → Insere algo depois do conteúdo de cada elemento p (usando a propriedade content)
- ✓ **p::before** → Insere algo antes do conteúdo de cada elemento p (usando a propriedade content)
- ✓ **p:first-letter** → Seleciona a primeira letra de cada elemento p
- ✓ **p:first-line** → Seleciona a primeira linha de cada elemento p
- ✓ **p::selection** → Seleciona a porção de um elemento p selecionada pelo usuário



# Combinação de seletores

```
p, h1, div {  
    color: #FF0000;  
    text-align: center;  
}
```

Em vez de criar três blocos (p, h1 e div) com as mesmas declarações, podemos agrupar todas elas, separadas por vírgula.

```
div a {  
    color: #333333;  
    text-align: left;  
}
```

Se desejarmos aplicar um efeito somente às tags “a” dentro de tag “div”, citamos as tags na ordem pai-filho, separadas por espaço.



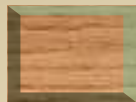
# Combinação de seletores

```
div#conteudo {  
    color: #333333;  
    text-align: left;  
}
```

**Combinando tag e id(#)** – podemos selecionar somente as tags div que apresentem o id conteudo.

```
div.conteudo {  
    color: #333333;  
    text-align: left;  
}
```

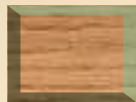
**Combinando tag e classe(.)** – podemos selecionar somente as tags div que apresentem a classe (.) conteudo.



**Introdução**



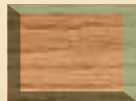
**React**



**HTML**



**CSS**



**Tableless**



**HTML e CSS**

Prof. Enzo  
Seraphim

Profa. Bárbara  
Pimenta Caetano



# Tableless

## Tableless

- ✓ Metodologia de desenvolvimento de layouts sem o uso de tabelas
- ✓ Substituindo-as pela tag `<div>` para montar os “blocos” do layout
- ✓ Nomear os blocos com ID

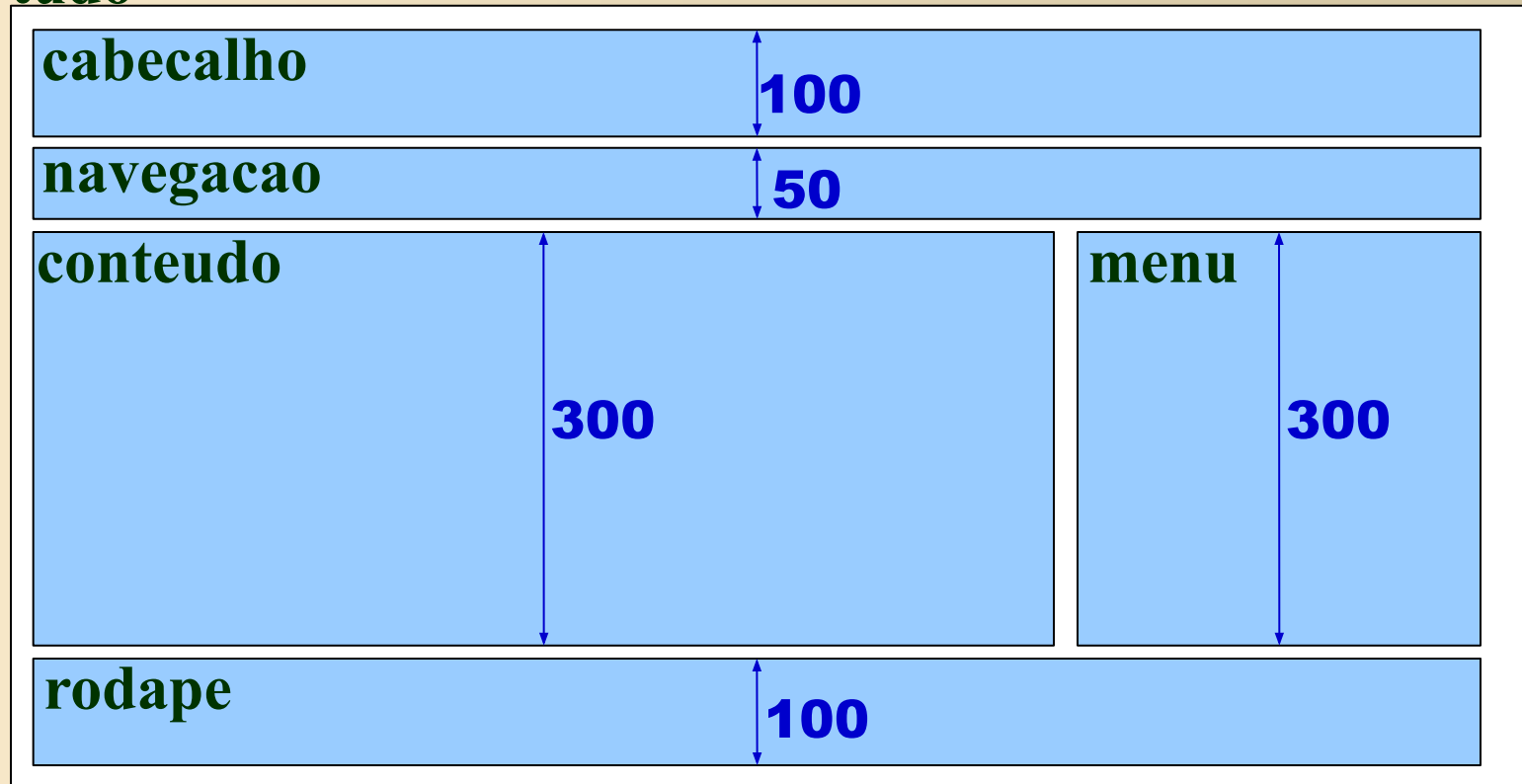




# Escolha os “blocos” do layout

tudo

Tableless



450 650





# Tableless

# HTML

```
<html><head>
<link rel="stylesheet" href="site.css"
type="text/css">
</head><body>
<div id="tudo">
<div id="cabecalho">cabeçalho</div>
<div id="navegacao">navegação</div>
<div id="menu">menu</div>
<div id="conteudo">conteúdo</div>
<div id="rodape">rodapé</div>
</div>
</body></html>
```





# Tableless

## Em React

```
import './App.css';

import React from 'react';

class App extends React.Component {
  render(){
    return (<body><div id="tudo"></div>
      <div id="cabecalho">cabeçalho</div>
      <div id="navegacao">navegação</div>
      <div id="menu">menu</div>
      <div id="conteudo">conteúdo</div>
      <div id="rodape">rodapé</div></body>);
  }
}

export default App;
```

App.js





# Tableless

## CSS

## App.css

```
#cabecalho {  
  background: #CF3;  
  height: 100px;  
}  
#navegacao {  
  background: #9F3;  
  height: 50px;  
}  
#menu {  
  float: right;  
  height: 300px;  
  width: 200px;  
  background: #6F9;  
}
```

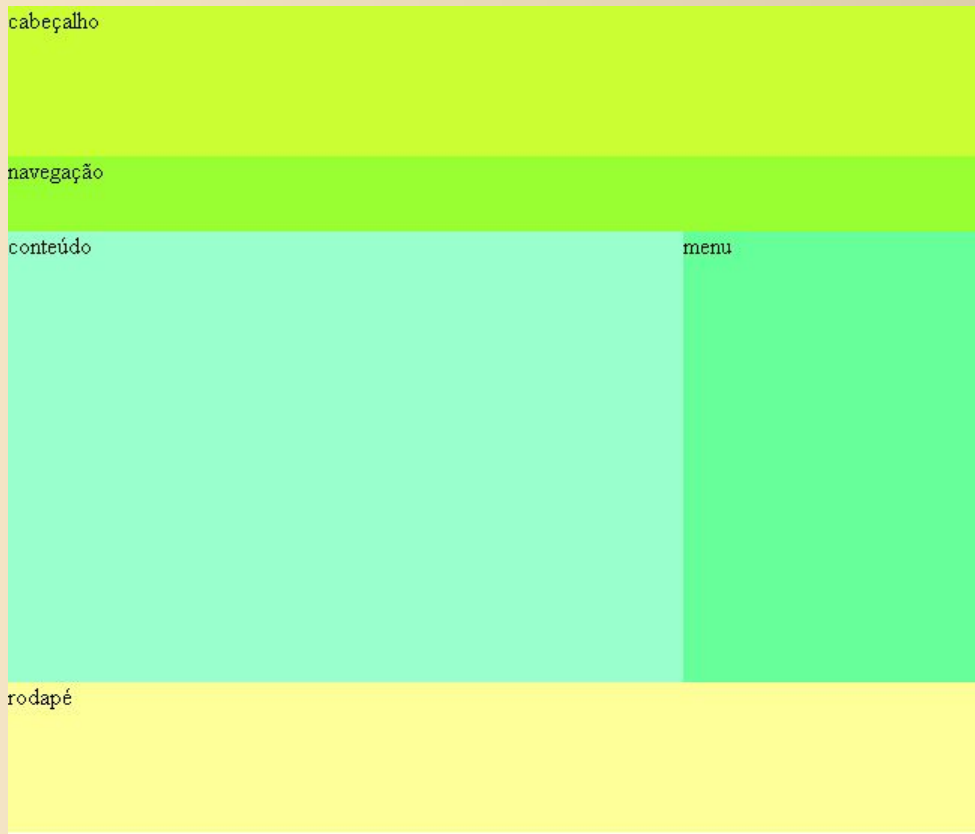
```
#conteudo {  
  float: left;  
  height: 300px;  
  width: 450px;  
  background: #9FC;  
}  
#rodape {  
  clear: both;  
  background: #FF9;  
  height: 100px;  
}  
#tudo {  
  margin: 1em auto;  
  width: 650px;  
}
```





# Tableless

## Browser



**Prof. Enzo  
Seraphim**

**Profa. Bárbara  
Pimenta Caetano**

Os logotipos, marcas comerciais e nomes de produtos citados nesta publicação tem apenas o propósito de identificação e podem ser marcas registradas de suas respectivas companhias.



**HTML e CSS**