



UNIPÊ
Centro Universitário
de João Pessoa

APLICAÇÕES PARA INTERNET

PROF. DANIEL BRANDÃO

@PROFDANIELBRANDAO

O QUE VIMOS ATÉ
AQUI?



O que vimos até aqui?

- Arquitetura e Protocolos de rede/internet
- Camadas de sistemas Web
- Tecnologias básicas da Web
- Primeiros passos com HTML
- TAGs de texto, link e imagem
- Listas, tabelas e Container
- Formulários e Inputs
- **Estilo e Layout**
- **Grid e Flexbox**

O QUE VAMOS
VER HOJE?



O que vamos ver HOJE?

Introdução a JavaScript e Bootstrap



The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on the left side is a large, semi-circular degree scale ranging from 150 to 260. Several concentric circles and dashed lines with arrows are scattered across the slide, creating a technical or scientific aesthetic.

MAS PRIMEIRO...

RESULTADO DA AULA PASSADA!

PÁGINA HTML

HTML Linguagem de Marcação de Hipertexto

Tutorial para Iniciantes



HTML (Linguagem de Marcação de HiperTexto) é o bloco de construção mais básico da web. Define o significado e a estrutura do conteúdo da web. Outras tecnologias do HTML geralmente são usadas...

Tabela de TAGs HTML

Tabela de TAGs HTML

Tag HTML	Para que Serve
HTML	define um documentos HTML
HEAD	Define a área de configuração de um documento HTML
BODY	Define a área de conteúdo de um documentos HTML
A	Tag de link (ancoragem) em elementos dentro de um doc HTML
IMG	Define uma imagem dentro de um doc HTML
Div	Elemento que divide e organiza conteúdos dentro de um doc HTML

Top 5 protocolos

1. HTTP - HyperText Transfer Protocol
2. FTP - File Transfer Protocol
3. POP - Post Office Protocol
4. SMTP - Simple Mail Transfer Protocol
5. IMAP - Internet Messaging Access Protocol

- HTTP - HyperText Transfer Protocol
- FTP - File Transfer Protocol
- POP - Post Office Protocol
- SMTP - Simple Mail Transfer Protocol
- IMAP - xxxxxxxx

Código Fonte disponível em:

<https://github.com/danielbrandao/appinternet>

JAVASCRIPT

Introdução

- JavaScript é uma linguagem de programação de alto nível amplamente utilizada para criar interatividade em páginas da web.
- Ela é uma linguagem de script leve e interpretada, o que significa que o código é executado linha por linha pelo navegador do usuário.

Introdução

- JavaScript é essencial para o desenvolvimento web moderno, permitindo a criação de funcionalidades dinâmicas, como validação de formulários, animações, manipulação do DOM e muito mais.

Paradigmas usados com JavaScript

JavaScript suporta diversos paradigmas de programação, incluindo:

- Programação Procedural: Foco na execução de instruções em sequência para resolver problemas.
- Programação Orientada a Objetos (POO): Baseada em objetos e classes, permitindo a encapsulação, herança e polimorfismo.

Paradigmas usados com JavaScript

JavaScript suporta diversos paradigmas de programação, incluindo:

- Programação Funcional: Concentra-se no uso de funções como elementos básicos de construção, evitando estados compartilhados e mutabilidade.
- Programação Orientada a Eventos: Baseada em responder a eventos (como cliques, teclas pressionadas) com funções específicas.

Paradigmas usados com JavaScript

JavaScript suporta diversos paradigmas de programação, incluindo:

- Programação Reativa: Foco na propagação de mudanças em dados ao longo do tempo, permitindo reações automáticas a essas mudanças.
- Esses paradigmas podem ser combinados e utilizados de forma flexível em JavaScript, permitindo que os desenvolvedores escolham a abordagem mais adequada para cada situação.

DOM (Document Object Model)

DOM (Document Object Model):

- O DOM é uma interface de programação que representa a estrutura de um documento HTML ou XML como uma árvore de objetos.
- Com o DOM, os desenvolvedores podem acessar, modificar e manipular os elementos de uma página web.

DOM (Document Object Model):

- Cada nó na árvore representa um objeto no documento (como elementos HTML, atributos, texto, etc.).
- O DOM é essencial para interações dinâmicas com o conteúdo da página, como adicionar, remover ou alterar elementos.

DOM (Document Object Model):

- Logo, o DOM é uma representação em forma de árvore que permite aos desenvolvedores acessar, modificar e manipular os elementos de uma página web de forma dinâmica. Vamos ver um exemplo simples de como manipular o DOM usando JavaScript
- Vamos a um primeiro exemplo:

DOM (Document Object Model):

Imprimindo um Olá, mundo na tela!

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplo 1</title>
</head>
<body>

  <button id="meuBotao">Clique Aqui</button>

  <script>
    document.getElementById("meuBotao").addEventListener("click", function() {
      alert("Olá, Mundo!");
    });
  </script>

</body>
</html>
```


DOM (Document Object Model):

- Num segundo exemplo, suponha que temos o seguinte código HTML com um botão e um parágrafo:

```
<html>
<head>
  <title>Exemplo de Manipulação do DOM</title>
</head>
<body>

<button id="meuBotao">Clique Aqui</button>
<p id="meuParagrafo">Texto Original</p>

<script>
  // Selecionando o botão e o parágrafo pelo ID
  const botao = document.getElementById("meuBotao");
  const paragrafo = document.getElementById("meuParagrafo");

  // Adicionando um evento de clique ao botão
  botao.addEventListener("click", function() {
    // Alterando o texto do parágrafo ao clicar no botão
    paragrafo.textContent = "Novo Texto!";
    // Alterando a cor do parágrafo ao clicar no botão
    paragrafo.style.color = "blue";
  });
</script>

</body>
</html>
```

DOM (Document Object Model):

No exemplo, ao clicar no botão "Clique Aqui", o texto do parágrafo será alterado para "Novo Texto!" e a cor do texto será alterada para azul. Isso é feito manipulando o conteúdo e o estilo do parágrafo diretamente no DOM usando JavaScript.

A seguir, um resumo do que está acontecendo no código:

DOM (Document Object Model):

1. O JavaScript seleciona o botão e o parágrafo usando seus IDs.
2. Um evento de clique é adicionado ao botão, e uma função é executada quando o botão é clicado.
3. Dentro da função do evento de clique, o texto do parágrafo é alterado para "Novo Texto!" usando `textContent`.
4. A cor do texto do parágrafo é alterada para azul alterando o estilo do parágrafo com `style.color`.

DOM (Document Object Model):

No próximo exemplo, vamos ver como alterar o texto de um parágrafo ao passar o mouse sobre ele.

Neste exemplo, ao passar o mouse sobre um parágrafo, o texto será alterado.


```
<html>
<head>
  <title>Exemplo 2</title>
</head>
<body>

<p id="meuParagrafo">Passe o mouse sobre mim</p>

<script>
  document.getElementById("meuParagrafo").addEventListener("mouseover", function() {
    document.getElementById("meuParagrafo").innerHTML = "Mouse passou por aqui!";
  });
</script>

</body>
</html>
```

DOM (Document Object Model):

Além de alterar conteúdo, podemos alterar características de estilo, como por exemplo o background-color de um elemento.

No próximo exemplo, ao clicar em um elemento, a cor de fundo será alterada.

```
<html>
<head>
  <title>Exemplo 3</title>
</head>
<body>

<div id="meuElemento" style="width: 100px; height: 100px; background-color: lightblue;"></div>

<script>
  document.getElementById("meuElemento").addEventListener("click", function() {
    this.style.backgroundColor = "lightcoral";
  });
</script>

</body>
</html>
```

DOM (Document Object Model):

E se quisermos manipular mais estilos em um container ou conteúdo em tela? Bem, podemos acionar classes e estilos CSS a partir de eventos do JS

Bora ver isso na prática?

```
<html>
<head>
  <title>Mudar Classe CSS com JavaScript</title>
  <style>
    .classeOriginal {
      background-color: lightblue;
    }
    .classeNova {
      background-color: lightcoral;
    }
  </style>
</head>
<body>

<div id="minhaDiv" class="classeOriginal">Div de Exemplo</div>
<button id="meuBotao">Mudar Classe</button>

<script>
  const minhaDiv = document.getElementById("minhaDiv");
  const meuBotao = document.getElementById("meuBotao");

  meuBotao.addEventListener("click", function() {
    minhaDiv.classList.toggle("classeNova");
  });
</script>

</body>
</html>
```


The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on this are several white geometric elements: a large circular scale on the left with degree markings from 150 to 260, and several concentric circles with arrows indicating clockwise rotation, some solid and some dashed.

O QUE MAIS POSSO FAZER COM JS?

JAVASCRIPT E SUAS VARIAÇÕES

- JavaScript é uma linguagem padrão que se tornou tão poderosa, que passou a evoluir de maneira bastante rápida.
- Com isso, surgiram variantes dela mesma, como forma de aprimorar seu uso prático, indo além do front-end
- Entre elas, vemos termos como EcmaScript e TypeScript.

JAVASCRIPT E SUAS VARIAÇÕES

- JavaScript, TypeScript e ECMAScript são termos relacionados à linguagem de programação JavaScript, mas cada um possui um significado específico. Podemos resumir assim:
 - **JavaScript** é a linguagem de programação original, amplamente utilizada para desenvolvimento web.
 - **ECMAScript (ES)** é o padrão internacional que define a sintaxe e os recursos do JavaScript.
 - **TypeScript** é um superset do JavaScript que adiciona tipagem estática opcional e outros recursos avançados.

JAVASCRIPT E SUAS VARIAÇÕES

- O TypeScript é amplamente utilizado em frameworks e bibliotecas populares de JavaScript para fornecer vantagens como tipagem estática, autocompletar avançado e detecção de erros durante a fase de desenvolvimento.
- Veremos a seguir alguns dos recursos empregados com TypeScript para o front-end

JAVASCRIPT E SUAS VARIAÇÕES

- **Angular**: é um framework de desenvolvimento web criado pelo Google. Ele é escrito em TypeScript e oferece suporte total ao uso do TypeScript em projetos.
- O Angular aproveita as vantagens do TypeScript, como tipagem estática, interfaces, decorators e outros recursos avançados para criar aplicações web robustas e escaláveis.

JAVASCRIPT E SUAS VARIAÇÕES

- **React com TypeScript:** React é uma biblioteca JavaScript para construção de interfaces de usuário. Muitos desenvolvedores optam por usar TypeScript com React para garantir a segurança e robustez do código.
- O TypeScript é integrado facilmente com o React e é amplamente utilizado para desenvolver aplicações React com tipagem estática e autocompletar avançado.

JAVASCRIPT E SUAS VARIAÇÕES

- **Vue.js com TypeScript:**
- Vue.js é um framework progressivo de JavaScript para a construção de interfaces de usuário. Assim como o React, o Vue.js também oferece suporte ao uso do TypeScript em projetos. O TypeScript pode ser facilmente integrado ao Vue.js para fornecer uma experiência de desenvolvimento mais robusta, com tipagem estática e verificação de erros em tempo de compilação..

JAVASCRIPT E SUAS VARIAÇÕES

- **Node.js com TypeScript:**
- Node.js é um ambiente de execução de JavaScript do lado do servidor. O TypeScript é amplamente utilizado em projetos Node.js para garantir a segurança e a escalabilidade do código.
- Ele permite aos desenvolvedores escrever código mais seguro, com menos erros e uma melhor manutenção do código em projetos Node.js.

The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on this are several white geometric elements: a large circular scale with degree markings from 150 to 260, and several concentric circles of varying sizes, some with arrows indicating a clockwise direction.

UTILIZANDO BOOTSTRAP

BOOTSTRAP

O Bootstrap é um framework front-end de código aberto desenvolvido pelo Twitter que facilita a criação de interfaces web responsivas e visualmente atraentes.

Ele fornece um conjunto de estilos CSS, componentes JavaScript e recursos de design responsivo que permitem aos desenvolvedores criar rapidamente layouts e designs consistentes para seus projetos web.

BOOTSTRAP

A estrutura do Bootstrap é baseada em um sistema de grid responsivo, componentes reutilizáveis e classes CSS predefinidas que facilitam a criação de páginas web modernas e adaptáveis a diferentes dispositivos e tamanhos de tela.

A seguir vamos ver as principais características e vantagens de uso deste framework

BOOTSTRAP

Grid System:

O Bootstrap utiliza um sistema de grid responsivo baseado em 12 colunas. Isso permite que os desenvolvedores criem layouts flexíveis e adaptáveis, dividindo a largura da página em 12 colunas para posicionamento de elementos.

Os desenvolvedores podem combinar as classes CSS do Bootstrap (como container, row e col) para criar layouts complexos e responsivos.

BOOTSTRAP

Componentes Reutilizáveis:

O Bootstrap oferece uma variedade de componentes prontos para uso, como botões, formulários, navegação, alertas, modais, entre outros.

Esses componentes podem ser facilmente integrados em seu projeto, economizando tempo e esforço de desenvolvimento.

BOOTSTRAP

Classes CSS Predefinidas:

O Bootstrap inclui um conjunto abrangente de classes CSS predefinidas que podem ser aplicadas diretamente aos elementos HTML para estilizá-los.

Por exemplo, classes como `btn` para estilizar botões, `alert` para estilizar alertas e `card` para criar cartões de conteúdo.

BOOTSTRAP

Responsividade:

O Bootstrap é altamente responsivo e se adapta automaticamente a diferentes tamanhos de tela, como desktops, tablets e dispositivos móveis. Isso é alcançado através do uso de classes de grid responsivas e @media queries.

BOOTSTRAP

Para começar a usar o Bootstrap em um projeto web, basta incluir os arquivos CSS e JavaScript do Bootstrap em seu código HTML, definir a estrutura de layout usando o sistema de grid e aplicar as classes CSS do Bootstrap aos elementos conforme necessário para estilizar e adicionar funcionalidades aos componentes.

O site oficial getbootstrap.com é um ótimo primeiro passo.

W3schools.com também tem um tutorial inicial

DESAFIO MÃO NA MASSA!

- **Vamos importar o arquivo CSS do Bootstrap para um projeto local e passar a utilizar suas classes pré-definidas.**
- **OBS: Não precisa enviar a atividade, traga ela na próxima aula para finalizarmos.**

**Próximo passo?
Versionamento!**



Referências

- **Bootstrap.** Disponível em: < <http://getbootstrap.com> >.
- **W3C.** HTML5 Forms. Disponível em <https://www.w3schools.com/html/html_form_input_types.asp>.