Aula 1 - Introdução à Programação com JavaScript

aTip Learn - Lógica de Programação com JavaScript

Objetivos da Aula

- Explicar o formato e objetivos do curso
- Compreender o que é programação e algoritmos
- Conhecer a história e aplicações da linguagem JavaScript
- Preparar o ambiente de desenvolvimento (VSCode e Node.js)
- Criar alguns exemplo de aplicações simples com JavaScript

Sobre o Curso

O curso será ministrado em 12 aulas, com duração de 2 horas cada. As aulas ocorreão semanalmente, às terças e quintas-feiras, das 19h às 21h.

Plataforma e Acessibilidade

As aulas ocorrerão de forma remota, através da plataforma Microsoft Teams. As aulas serão gravadas e disponibilizadas posteriormente.

O Teams possui recursos de acessibilidade como legendas e transcrição em tempo real. Caso você não saiba como ativar esses recursos consulte a documentação abaixo:

- Legendas no Microsoft Teams
- Transcrição em tempo real no Microsoft Teams
- Outras Dicas de Acessibilidade

Instrutor

- Eduardo P de Sousa
  - Mestre em Computação Aplicada
  - Professor do Instituto Federal de São Paulo
  - Contato: eduardo.sousa@atip.io
  - LinkedIn: https://www.linkedin.com/in/edupsousa/

O que é Lógica de Programação?

Lógica de programação é a disciplina que estuda a criação de algoritmos para a resolução de problemas computacionais. Ou de uma forma mais simples, é a disciplina responsável por ensinar computadores como resolver problemas.

Para isso a lógica de programação se utiliza de ferramentas chamadas algoritmos e estruturas de dados. Algoritmos são sequências de passos que devem ser seguidos para a resolução de um problema, como uma receita de bolo ou um manual de instruções que ensina como montar um móvel passo a passo. Já as estruturas de dados são formas de organizar e armazenar informações de forma eficiente, auxiliando na resolução de problemas.

Programação em nosso Dia a Dia

A programação está presente em diversos aspectos de nosso dia a dia, desde a criação de aplicativos de celular, sites e jogos até a automação de processos industriais, análise de dados e inteligência artificial.

Computadores, smartphones e dispositivos eletrônicos em geral seriam apenas peças de metal e plástico se não houvesse programação. A programação é o que dá vida a esses dispositivos, permitindo que eles executem tarefas e resolvam problemas.

Algoritmos

Algoritmos são sequências de passos que devem ser seguidos para a resolução de um problema. Eles são como receitas de bolo ou manuais de instruções que nos guiam para a solução de um problema. Aprender a programar é aprender a criar algoritmos que possam ser seguidos por computadores para resolução de problemas.

Entretanto, diferente de receitas ou manuais, que são escritos por pessoas para que outras pessoas possam seguir, algoritmos são escritos para que computadores possam seguir. E diferente das pessoas os computadores são incapazes de seguir instruções que não estejam claras e precisas.

Enquanto uma pessoa é capaz de substituir um parafuso ou um ingrediente por algo similar, um computador não é capaz de fazer isso. Se um algoritmo não estiver claro e preciso, o computador não será capaz de executá-lo corretamente.

Além disso, os computadores possuem limitações de memória e processamento que devem ser levadas em consideração ao se criar algoritmos. Imagine que você tem um conjunto de 10 post-its numerados na sua frente e que você deve ordenar esses post-its em ordem crescente. Provavelmente apenas olhando para esses post-its você seria capaz de imaginar a posição correta de cada um deles, e assim organizá-los rapidamente. Porém, quando se trata de um computador ele não é capaz de fazer isso, ele precisa de um algoritmo que o guie passo a passo para a resolução do problema, indicando como comparar os valores dos post-its e como trocá-los de lugar, um a um.

Material Complementar:

- Wikipedia Algoritmos
- Vídeo Introdução a Algoritmos - Gustavo Guanabara

Linguegens de Programação

Enquanto um algoritmo é uma sequência de passos que devem ser seguidos para a resolução de um problema, uma linguagem de programação é um conjunto de regras e símbolos que permitem a escrita de algoritmos de forma que computadores possam compreender. Assim, um algoritmo pode ser expresso em linguagem natural, como o português, porém, para que um computador possa compreeendê-lo e executá-lo é necessário traduzi-lo para uma linguagem de programação.

Existem diversas linguagens de programação, cada uma com suas características e aplicações. Algumas linguagens são mais adequadas para a criação de sites e aplicativos, como o JavaScript, outras são mais adequadas para a análise de dados e inteligência artificial, como o Python, e outras são mais adequadas para a criação de sistemas operacionais e drivers, como o C. Além da sua adequação ou popularidade em um determinado domínio, as linguagens de programação também podem ser categorizadas de outras formas, compiladas ou interpretadas, de propósito geral ou específicos, com tipagem estática ou dinâmica, entre outras.

Em nosso curso vamos utilizar a linguagem JavaScript, uma linguagem de programação interpretada, de propósito geral e com tipagem dinâmica. O JavaScript é uma das linguagens de programação mais populares e amplamente utilizadas no mundo, sendo a linguagem de programação mais utilizada para o desenvolvimento de sites e aplicações para Web.

História do JavaScript

A linguagem JavaScript, ou simplesmente JS, foi criada por Brendan Eich em 1995, enquanto trabalhava na Netscape Corporation. O objetivo da linguagem era permitir a criação de páginas web dinâmicas, que respondessem a eventos e interações do usuário, sem a necessidade de recarregar a página a cada clique ou ação.

O nome JavaScript foi escolhido por questões de marketing, para capitalizar o sucesso da linguagem Java na época. Porém, apesar do nome, JavaScript e Java são linguagens completamente diferentes, com propósitos e

características bastante distintas. Em comum, além do nome, as duas linguagens possuem apenas a sintaxe similar, derivada da linguagem C.

Com o sucesso e rápida adoção da linguagem, a Netscape submeteu o JavaScript à ECMA International, uma organização que padroniza tecnologias de informação, para que a linguagem fosse padronizada e pudesse ser implementada por outros navegadores. A padronização da linguagem resultou na criação da especificação ECMAScript, que é a base da linguagem JavaScript até hoje. Quando falamos de termos como ES5, ES6 ou ES7, na verdade estamos nos referindo a diferentes versões da especificação **E**CMA**S**cript.

Atualmente o JavaScript é uma das linguagens de programação mais populares e amplamente utilizadas no mundo, e seu uso foi muito além das páginas da Web, sendo utilizada para o desenvolvimento de aplicativos móveis, servidores, jogos, robótica, IoT, entre outros. Muito desses novos usos da linguagem JavaScript se devem ao desenvolvimento do Node.js, um ambiente de execução de JavaScript fora do navegador, que permitiu a criação de aplicações JavaScript independentes do navegador.

O Node.JS foi criado por Ryan Dahl em 2009, para permitir a criação de aplicações de rede escaláveis e de alta performance. O Node.js é baseado no motor de JavaScript V8, criado pela Google para o navegador Chrome. O papel do V8 é interpretar e executar o código JavaScript, transformando as instruções da linguagem em código de máquina que pode ser executado pelo processador. Unindo essa capacidade, a bibliotecas e APIs que permitissem que a linguagem interagisse com o sistema operacional, o Node.js se tornou uma popular plataforma para o desenvolvimento de aplicações de rede, servidores e APIs.

Aplicações do JavaScript

O JavaScript é uma linguagem de programação versátil e amplamente utilizada, com aplicações em diversos domínios. Alguns exemplos de aplicações do JavaScript são:

- Desenvolvimento de Sites e Aplicações de Frontend Web:
  - Esse é o propósito original da linguagem, e ainda hoje é uma das suas principais aplicações. O JavaScript é capaz de interagir com o navegador Web para adicionar interatividade a páginas Web, como validação de formulários, animações, efeitos visuais, entre outros. Hoje em dia é muito comum a utilização de bibliotecas e frameworks como React, Angular e Vue.js para o desenvolvimento dessas aplicações.
- Desenvolvimento de Aplicações de Backend Web:
  - Com o desenvolvimento do Node.js, o JavaScript passou a ser utilizado também no desenvolvimento de aplicações de backend, servidores e APIs. O Node.js permite a criação de aplicações de rede escaláveis e de alta performance, que podem ser utilizadas para a criação de APIs REST, servidores Web, aplicações de tempo real, entre outros. O JavaScript também deu origem a uma categoria especial de linguagens transpiladas (ou seja, que são traduzidas para outra linguagem para que possam ser executadas), como o TypeScript, que adiciona tipagem estática e outras funcionalidades à linguagem.
- Desenvolvimento de Aplicações Móveis:
  - O JavaScript é amplamente utilizado no desenvolvimento de aplicativos móveis, através de frameworks como React Native, Ionic e NativeScript. Esses frameworks permitem a criação de aplicativos móveis multiplataforma, que podem ser executados em dispositivos Android e iOS, utilizando uma única base de código.
- Desenvolvimento de Jogos:
  - O JavaScript é amplamente utilizado no desenvolvimento de jogos, através de bibliotecas e frameworks como Phaser, Three.js e Babylon.js. Essas bibliotecas permitem a criação de jogos 2D e 3D, que podem ser executados diretamente no navegador, sem a necessidade de plugins ou instalações adicionais.
- Desenvolvimento de Aplicações Desktop:
  - O JavaScript também é utilizado no desenvolvimento de aplicações desktop, através de frameworks como Electron e Tauri. Esses frameworks permitem a criação de aplicações desktop multiplataforma, que podem ser executadas em Windows, macOS e Linux, utilizando tecnologias Web como HTML, CSS e JavaScript.

- Desenvolvimento de Aplicações de IoT:
  - O JavaScript é utilizado no desenvolvimento de aplicações de Internet das Coisas, através de plataformas como Johnny-Five e Espruino. Essas plataformas permitem a criação de aplicações que interagem com sensores, atuadores e outros dispositivos eletrônicos, utilizando JavaScript.

Ambiente de Desenvolvimento

Para estudarmos e desenvolvermos pequenas aplicações com JavaScript podemos utilizar ferramentas online, simples e gratuítas como os editores de código:

- Programiz
- PlayCode
- OneCompiler
- JSFiddle

Porém, para aplicações mais complexas é recomendado que utilizemos um ambiente de desenvolvimento integrado (IDE), como o Visual Studio Code (VSCode) instalado em nosso computador. O VSCode é um editor de código gratuito, leve e altamente customizável, que possui suporte a diversas linguagens de programação, incluindo JavaScript.

Além disso é recomendado que instalemos o Node.js em nosso computador, para que possamos executar aplicações JavaScript fora do navegador, para que possamos executar e testar nossos códigos JavaScript fora do navegador.

Instalação do Visual Studio Code

A instalação do VSCode é bastante simples, porém, o processo pode variar de acordo com o sistema operacional utilizado. Para instalar o VSCode em seu computador siga os passos abaixo:

1. Acesse o site oficial do Visual Studio Code em code.visualstudio.com
2. Clique no botão de download para o seu sistema operacional
3. Execute o instalador e siga as instruções na tela
4. Após a instalação, abra o VSCode para confirmar que a instalação foi bem sucedida.

Em caso de dúvidas você pode consultar a documentação oficial do Visual Studio Code em code.visualstudio.com/docs.

Instalação do Node.js

A instalação do Node.js também é bastante simples, porém, o processo pode variar de acordo com o sistema operacional utilizado. Para instalar o Node.js em seu computador siga os passos abaixo:

1. Acesse o site oficial do Node.js em nodejs.org
2. Clique no botão de download para o seu sistema operacional, certifique-se de baixar a versão LTS (Long Term Support) do Node.js.
3. Execute o instalador e siga as instruções na tela
4. Após a instalação, abra o terminal (prompt de comando ou power-shell) e execute o comando `node -v` para verificar se o Node.js foi instalado corretamente, caso a instalação tenha sido bem sucedida a versão do Node.js será exibida no terminal.

Em caso de dúvidas você pode consultar a documentação oficial do Node.js em nodejs.org/docs.

Material Complementar

- Tutorial de Instalação do Node.js em Vídeo

Primeiros Exemplos de Código JavaScript

Utilizando o VSCode + Node.js

Vamos criar nosso primeiro exemplo de código JavaScript, um programa que exibe uma mensagem de boas-vindas no console. Para isso siga os passos abaixo:

1. Abra o Visual Studio Code
2. Crie um novo arquivo clicando em `File > New File`
3. Digite o seguinte código no arquivo:

```
console.log('Olá, Mundo!');
```

4. Salve o arquivo clicando em `File > Save As...` e escolha um nome para o arquivo, por exemplo `ola-mundo.js`
5. Abra o terminal no VSCode clicando em `Terminal > New Terminal`
6. Execute o arquivo JavaScript no terminal digitando o comando `node ola-mundo.js`
7. A mensagem `Olá, Mundo!` deve ser exibida no terminal

Utillizando um Editor Online

Se você não conseguiu ou não teve tempo de instalar o VSCode e o Node.js em seu computador, você pode utilizar um editor online para executar o código JavaScript. Para isso siga os passos abaixo:

1. Abra o navegador e acesse o endereço playcode.io/javascript.
2. No editor você deve ver duas abas abertas, uma com o arquivo index.html e outra com o arquivo script.js, certifique-se que a aba script.js está selecionada.
3. Substitua o código existente no arquivo script.js pelo código abaixo:

```
console.log('Olá, Mundo!');
```

4. Feito isso você já deve visualizar a mensagem `Olá, Mundo!` no console do editor.

Explicação do Código

O código que acabamos de criar é um exemplo bastante simples que apenas exibe uma mensagem de boas-vindas no console. Vamos analisar esse código parte a parte:

- `console`: é um objeto global do JavaScript que está disponível tanto no navegador quanto no Node.js. Esse objeto possui diversos métodos que permitem exibir informações no console. Quando utilizamos o navegador, o console é uma ferramenta de desenvolvimento onde são exibidas mensagens de log, erros e avisos. Para visualizar o console no Google Chrome você pode pressionar `F12` e clicar na aba `Console`. Já no Node.js o console refere-se ao terminal onde o programa está sendo executado.
- `log`: é um método do objeto `console` que exibe uma mensagem no console. O método `log` aceita um ou mais argumentos, que podem ser de qualquer tipo, e exibe esses argumentos no console.
- `'Olá, Mundo!'`: é uma string, ou seja, um texto, nesse caso delimitado por aspas simples. Essa string é o argumento que estamos passando para o método console.log, e que será exibido no console. Na linguagem JavaScript as strings podem ser delimitadas por aspas simples (`'`), aspas duplas (`"`) ou crases (`` ` ``). A escolha entre aspas simples ou duplas é uma questão de preferência, porém, é importante manter a consistência ao longo do código. Já o crase possui uma funcionalidade adicional, de interpolação de strings, sobre a qual falaremos mais adiante.

Introdução às Variáveis em JavaScript

Vamos reescrever o código anterior utilizando uma variável para armazenar a mensagem de boas-vindas:

```
let mensagem = 'Olá, Mundo!';
console.log(mensagem);
```

Executando esse código você verá o mesmo resultado do código anterior. Abaixo vamos analisar as mudanças que fizemos no código:

- `let mensagem = 'Olá, Mundo!';`: declaramos uma variável chamada `mensagem` utilizando a palavra-chave `let`. A palavra-chave `let` é utilizada para declarar variáveis na linguagem JavaScript. Nesse caso estamos declarando uma variável chamada `mensagem` e atribuindo a ela o valor `'Olá, Mundo!'`.
- `console.log(mensagem);`: diferente do código inicial onde passamos a mensagem diretamente para o método `console.log`, nesse código passamos a variável `mensagem` como argumento para o método `console.log`. O resultado é o mesmo do código anterior.

Perceba que em ambos os casos a string *Olá, Mundo!* precisa ser armazenada na memória para que seja utilizada pelo método `console.log`. Utilizando uma variável nós estamos apenas identificando o espaço de memória onde essa string é armazenada. Isso permite que possamos reutilizar essa string em outros lugares do código, sem a necessidade de repetir o texto. Por exemplo, ao invés de escrevermos:

```
console.log('Olá, Mundo!');
console.log('Olá, Mundo!');
console.log('Olá, Mundo!');
```

Podemos escrever:

```
let mensagem = 'Olá, Mundo!';
console.log(mensagem);
console.log(mensagem);
console.log(mensagem);
```

Esse é um exemplo bastante simplista, porém, ilustra uma das funcionalidades que o uso de variáveis traz para o código, a reutilização de valores.

Tipos de Dados em JavaScript

Nos exemplos anteriores falamos sobre strings, ou seja, textos, que são um dos tipos de dados básicos da linguagem JavaScript. Além das strings, o JavaScript possui outros tipos de dados básicos, como números, booleanos, arrays e objetos. Abaixo vamos falar um pouco sobre cada um desses tipos de dados:

- `String`: é um tipo de dado que representa um texto. As strings podem ser delimitadas por aspas simples (`'`), aspas duplas (`"`) ou crases (`` ` ``). Exemplos de strings são `'Olá, Mundo!'`, `"JavaScript"` e `` `Aula 1` ``.
- `Number`: é um tipo de dado que representa um número, inteiro ou decimal. Exemplos de números são `42`, `3.14` e `-1`.
- `Boolean`: é um tipo de dado que representa um valor lógico, verdadeiro ou falso. Os valores booleanos são `true` e `false`.
- `Array`: é um tipo de dado que representa uma coleção de valores, que podem ser de qualquer tipo. Os valores de um array são acessados através de um índice, que começa em zero. Exemplos de arrays são `[1, 2, 3]`, `['a', 'b', 'c']` e `['JavaScript', 11, true]`.
- `Object`: é um tipo de dado que representa um objeto, que pode possuir propriedades e métodos. Os objetos são coleções de pares chave-valor, onde a chave é uma string que identifica a propriedade e o valor é o valor da propriedade. Exemplos de objetos são `{nome: 'Alice', idade: 30}`, `{cor: 'azul', tamanho: 'grande'}`, `{}` e `null`.
- `undefined`: é um tipo de dado que representa um valor indefinido. O valor `undefined` é retornado quando uma variável é declarada mas não inicializada. Por exemplo, `let x; console.log(x);` irá exibir `undefined` no console.

```
let texto = 'Olá, mundo!';
let numero = 42;
```

```
let booleano = true;
let objeto = {
  chave: 'valor'
};
let array = [1, 2, 3];
let indefinido = undefined;

console.log(texto);
console.log(numero);
console.log(booleano);
console.log(objeto);
console.log(array);
console.log(indefinido);
```

Futuramente iremos discutir esses tipos de dados com mais detalhes, porém, por enquanto é importante que você tenha uma noção básica sobre eles para que possa compreender os exemplos de código que iremos criar.

Tipagem Dinâmica em JavaScript

Dizemos que o JavaScript é uma linguagem de tipagem dinâmica, ou seja, as variáveis em JavaScript não possuem um tipo de dado fixo, e podem armazenar valores de qualquer tipo. Isso significa que uma variável pode armazenar um número em um momento e uma string em outro momento, sem a necessidade de declarar explicitamente o tipo da variável. Vimos isso em prática nos exemplos anteriores quando declaramos e inicializamos a variávem `mensagem` com uma string. Em momento nenhum precisamos informar explicitamente que a variável `mensagem` é do tipo string, o JavaScript inferiu isso a partir do valor que atribuímos à variável.

A tipagem dinâmica traz flexibilidade e simplicidade para a linguagem, porém, também pode trazer confusão e erros se não utilizada corretamente. Por isso é importante que as variáveis sejam utilizadas de forma consistente, evitando que uma mesma variável armazene valores de tipos diferentes ao longo do código, o que torna o código confuso e mais propenso a bugs.

Conclusão

Nessa aula aprendemos o que é programação e algoritmos, conhecemos a história e aplicações da linguagem JavaScript, preparamos o ambiente de desenvolvimento (VSCode e Node.js) e criamos alguns exemplos de aplicações simples com JavaScript. Aprendemos também sobre variáveis e tipos de dados em JavaScript, e vimos como utilizar variáveis para armazenar e reutilizar valores em nossos programas.

Nas próximas aulas iremos aprofundar nossos conhecimentos em JavaScript, aprendendo mais sobre variáveis, operadores e estrutras de controle para criar programas mais complexos e interessantes. Até lá!

Referências

Lógica de Programação e Algoritmos:

- **Khan Academy - Algoritmos:** https://pt.khanacademy.org/computing/computer-science/algorithms
- **Curso em Vídeo - Algoritmos:** https://www.cursoemvideo.com/curso/curso-de-algoritmo/

História do JavaScript:

- **Mozilla Developer Network - A re-introduction to JavaScript:** https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_overview

Aplicações do JavaScript:

- **JavaScript.com - What is JavaScript?:** https://www.javascript.com/

Ambiente de Desenvolvimento:

- **Visual Studio Code - Download:** https://code.visualstudio.com/download
- **Node.js - Download:** https://nodejs.org/en/download/
- **Tutorial de Instalação do Node.js em Vídeo:** https://www.youtube.com/watch?v=-jft_9PlffQ

Primeiros Exemplos de Código JavaScript:

- **W3Schools - JavaScript Tutorial:** https://www.w3schools.com/js/
- **MDN Web Docs - JavaScript Basics:** https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

Outras Referências:

- **Legendas no Microsoft Teams:** https://tinyurl.com/legendas-teams
- **Transcrição em tempo real no Microsoft Teams:** https://tinyurl.com/transcricao-teams
- **Outras Dicas de Acessibilidade:** https://tinyurl.com/dicas-acessibilidade
- **LinkedIn (Eduardo P de Sousa):** https://www.linkedin.com/in/edupsousa/
- **Wikipedia Algoritmos:** https://pt.wikipedia.org/wiki/Algoritmo
- **Vídeo Introdução a Algoritmos - Gustavo Guanabara:** https://www.youtube.com/watch?v=8mei6uVttho
- **Programiz (Editor de código online):** https://www.programiz.com/javascript/online-compiler/
- **PlayCode (Editor de código online):** https://playcode.io/javascript
- **OneCompiler (Editor de código online):** https://onecompiler.com/javascript
- **JSFiddle (Editor de código online):** https://jsfiddle.net/