## Capacitação HTML + CSS + JavaScript Dia 03



O1 Sobre o JS

02 Conceitos fundamentais da linguagem

03 API DOM

#### 1. Sobre o JS

JavaScript é uma linguagem de programação que insere interatividade ao site.

Se trata de uma das mais populares ferramentas para web! O domínio dessa tecnologia pode dar uma experiência totalmente nova ao usuário JavaScript é versátil e amigável para iniciantes. Com mais experiência, você será capaz de criar jogos, gráficos animados 2D e 3D, aplicativos abrangentes baseados em banco de dados e muito mais.

As Interfaces de Programação de Aplicativos (APIs) de aplicativos de navegador são incorporadas aos navegadores da web, fornecendo funcionalidades como criar dinamicamente HTML e definir estilos CSS, coletar e manipular uma transmissão de vídeo da webcam do usuário ou gerar gráficos 3D e amostras de áudio.

# 2. Conceitos fundamentais da linguagem

#### Atribuições e contas no console

Uma variável, em qualquer linguagem de programação, é o nome dado a um espaço de memória destinado a armazenar um valor que pode variar durante a execução do programa

A atribuição é utilizada para alterar o valor de uma variável ou inicializá-la com um determinado valor, podemos dar como exemplo o trecho abaixo:

let a = 120 a = 30 a = a+a a fica valendo 60

#### Vinculando o JS no html

Podemos referenciar um arquivo .js em uma tag script usando o src. É recomendado criar a pasta scripts

```
index.html
  EXPLORER
                         Js main.js
                         (> index html > ...
V DIA-03
                                <!DOCTYPE html>

✓ assets

                                <html lang="en">
   > images
                                <head>

√ scripts

                                    <meta charset="UTF-8">
   JS main.js
                                    <meta http-equiv="X-UA-Compatible" content="IE=edge">
   > styles
                                    <meta name="viewport" content="width=device-width, initial-scale=1.0">
 () index.html
                                    <title>Document</title>
                                </head>
                                <body>
                                    <script src="assets/scripts/main.js"></script>
                          10
                                </body>
                          11
                                </html>
                          12
```

#### console.log()

Função utilizada para imprimir no console, que é acessado a partir da ferramenta de desenvolvedor (inspecionar).

#### Criando variáveis

Uma variável, em qualquer linguagem de programação, é o nome dado a um espaço de memória destinado a algum dado que pode ser alterado durante a execução do programa

#### Tipos primitivos no JavaScript

**String:** é uma sequência de zero ou mais caracteres. Uma string literal começa e termina com aspas simples ou duplas;

**Undefined:** por padrão, quando uma variável é declarada, mas não inicializada, ela recebe o valor undefined;

**Null:** é o segundo tipo de dado primitivo que tem somente um valor e define que é um ponteiro de objeto vazio;

**Number:** o JS usa o formato padrão IEEE-754 para representação de inteiros e pontos flutuantes;

**Boolean:** possui os valores true ou false;

#### **Principais Operadores**



#### Aritmético

São usados para operações de soma, subtração, multiplicação, divisão, módulo e exponenciação.



#### Comparação

Os operadores de comparação envolvem dois operandos e retornam um resultado lógico.



#### Atribuição

É o sinal de =, que atribui um valor a variável.

#### **Principais Operadores**

#### Lógico



São usados para realizar operações lógicas, do tipo AND, OR e NOT.

#### Concatenação



É usado para unir strings como uma soma.

#### Condicional



É uma contração de um operador de atribuição e duas expressões lógicas.

#### **Condicional: if e else**

- Executa um bloco de código SE EXPRESSÃO for verdadeira
- Caso contrário executa o segundo bloco

```
if (EXPRESSAO1) {
    //EXPRESSAO1 é verdadeira
} else {
    //EXPRESSAO1 é falsa
}
```

#### Condicional: if, else e else if

```
if (EXPRESSA01) {
    //EXPRESSA01 é verdadeira
  } else if (EXPRESSA02) {
    //EXPRESSA02 é verdadeira
  } else if (EXPRESSAO3) {
    //EXPRESSAO3 é verdadeira
  } else {
    //nenhuma das expressões são verdadeiras
```

#### **Condicional: Switch case**

```
switch (op) {
   case 1:
      //executa quando op vale 1
      break;
   case 2:
      //executa quando op vale 2
     break;
   case 3:
      //executa quando op vale 3
     break;
   default:
      //executa quando op vale algo diferente de 1, 2 ou 3
      break;
```

## Vamos praticar

#### **Arranjos / Array**

- Um arranjo/array é uma forma de armazenar diversos valores em somente uma variável.
- Podemos criar um vetor/array e já atribuir valores em suas posições.

```
const frutas = ["Maçã", "Banana"];
const nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0];
const pessoa = ["Márcio", "Victor", 42];
```

#### **Arranjos / Array**

```
const nums = [];
nums[0] = 1;
nums[1] = 2;
nums[2] = 3;
nums[3] = 4;
nums[4] = 5;
nums[5] = 6;
nums[6] = 7;
nums[7] = 8;
nums[8] = 9;
nums[9] = 0;
```

#### Métodos de Array

Arrays são objetos semelhantes a listas que vêm com uma série de métodos embutidos para realizar operações de travessia e mutação.

```
const nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0];
const alfabeto = ['c', 'd', 'a', 'b', 'j'];
let frutas = ["Maçã", "Banana"];
.length
console.log(nums.length); // imprime 10
Sort
alfabeto.sort(); // ['a', 'b', 'c', 'd', 'j']
Push e Pop
frutas.push("Laranja"); // adiciona elementos ao final do array e retorna o novo comprimento desse array
// ["Maçã", "Banana", "Laranja"]
frutas.pop(); // remove o último elemento de um array e retorna aquele elemento
// ["Maçã", "Banana"]
```

#### Laço de Repetição: For

- Realiza uma operação inicial
- Executa o bloco enquanto a expressão for verdadeira
- Realiza uma operação ao fim de cada execução

```
for(let i = 0; i < 10; i++) {
    //executa o bloco 10 vezes
}</pre>
```

#### Laço de Repetição: While

• Executa enquanto a expressão for verdadeira

```
let i = 0;
while(i < 10) {
    //executa o bloco 10 vezes
    i++;
}</pre>
```

#### Laço de Repetição: Do While

- Executa o bloco uma vez
- Depois enquanto a expressão for verdadeira o bloco é executado

```
let i = 0;
do {
    //executa o bloco 10 vezes
    i++;
} while(i < 10);</pre>
```

#### **Funções**

- Um conjunto de instruções que executa uma tarefa ou calcula um valor.
- Se você está repetindo diversas vezes um mesmo bloco de código, muito provavelmente poderá transformá-lo em uma função.

```
function nomeFuncao(parametro1, parametro2, ..., parametroN) {
    //bloco de instrução da função
    //pode ou não retornar algo
}
```

## Vamos praticar

#### 3. API DOM

e detalhes avançados

#### **DOM**

- O DOM define métodos que permitem acesso à árvore, para que eles possam alterar a estrutura, estilo e conteúdo do documento.
- Embora o DOM seja frequentemente acessado usando JavaScript, não é uma parte da linguagem JavaScript. Ele também pode ser acessado por outras linguagens.

#### DOM

• Para pegarmos o primeiro parágrafo encontrado no documento, podemos utilizar o código abaixo:

```
const paragrafo = document.querySelector('p');
```

#### **DOM**

- O document diz respeito ao próprio documento HTML.
- O método querySelector retorna o primeiro elemento do documento que for igual ao passado como parâmetro para ele.
- Existem muitos métodos no DOM, alguns deles são:

```
document.querySelectorAll('seletor');
document.getElementById('id');
document.createElement('div');
elemento.setAtributte(nome, valor);
elemento.addEventListener('tipo', função);
```

#### Métodos internos do DOM: JSON

- JSON (JavaScript Object Notation) é, como o próprio nome já diz, uma notação de objetos baseada em JavaScript.
- Apesar de ser derivado do JavaScript, também é suportado em diversas outras linguagens de programação (C#, Java, Python).
- É comumente usado para trocar informações/dados entre clientes e servidores web.
- Para seres humanos é fácil de ler e escrever e para máquinas é fácil de interpretar e gerar.

#### Métodos internos do DOM: JSON

• Exemplo:

```
"nome": "Cássio Ramos",
"idade": 35,
"time": "Sport Club Corinthians Paulista"
}
```

#### Métodos internos do DOM: JSON

- Possui dois métodos:
- JSON.parse()

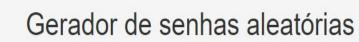
Recebe uma string JSON e a transforma em um objeto JavaScript.

JSON.stringify()

Recebe um objeto JavaScript e o transforma em uma string JSON.

### Desafio

Criar uma página parecida com a imagem a seguir



Clique no botão abaixo para gerar uma senha totalmente aleatória para você

Gerar senha



#### Gerador de senhas aleatórias

Clique no botão abaixo para gerar uma senha totalmente aleatória para você

Gerar senha

ceVIMPSzUX 🗗

## Obrigado(a)!

Dúvidas?

e-mail: **pet@din.uem.br** instagram: **@petinfouem** 

#### Referências Bibliográficas

CSS. MDN Web Docs, c.1998-2023. Disponível em: <a href="https://developer.mozilla.org/pt-BR/docs/Web/JavaScript">https://developer.mozilla.org/pt-BR/docs/Web/JavaScript</a> Acesso em 10 de março de 2023.