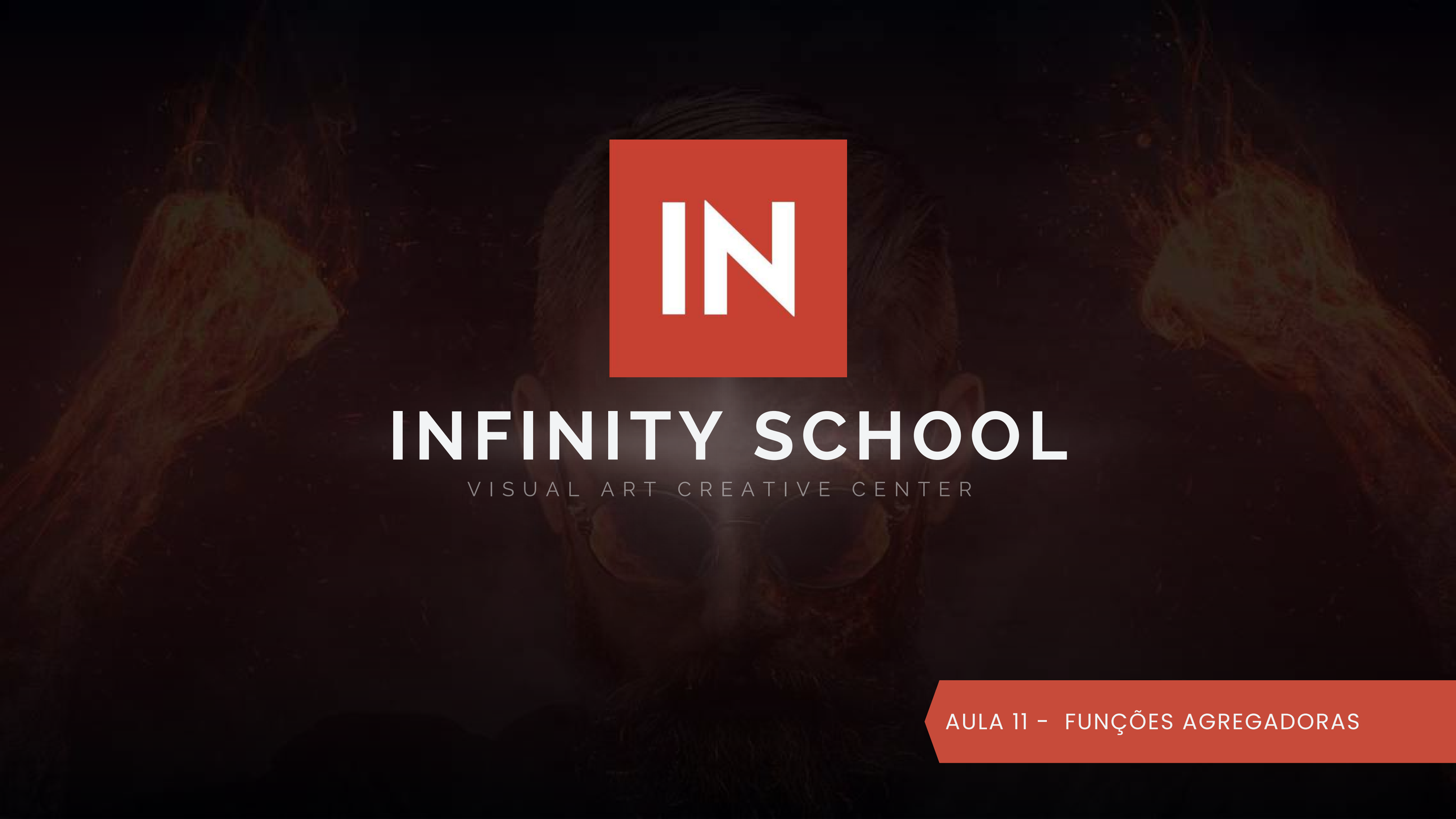


INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

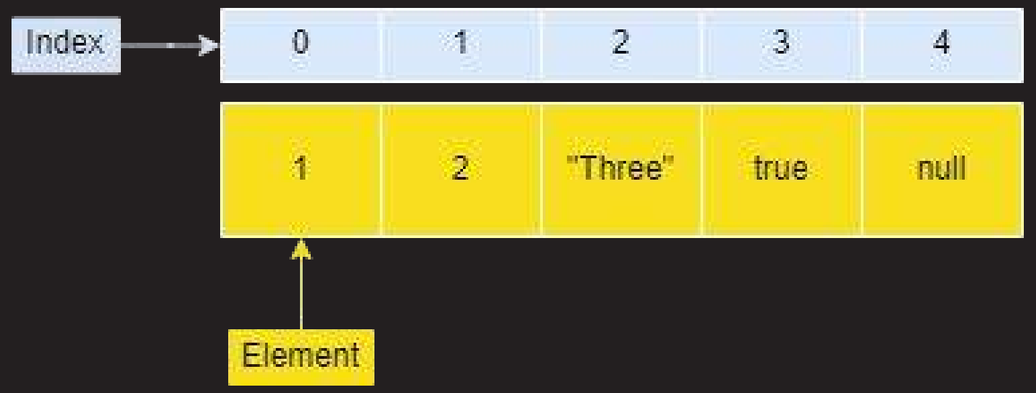
AULA 11 - FUNÇÕES AGREGADORAS

O QUE IREMOS APRENDER

01	ARRAYS
02	JOIN
03	SLICE
04	SPLICE
05	FOREACH
06	MAP
07	DIFERENÇA ENTRE MAP E FOREACH
08	FILTER
08	REDUCE
08	MÃOS AO CÓDIGO

ARRAYS

Você se lembra da aula 5? Não? Na aula 5 vimos o conceito de arrays. Arrays são estruturas de dados em programação que permitem armazenar vários valores em uma única variável. Cada valor em um array é chamado de elemento, e cada elemento tem um índice associado, que representa sua posição no array.



FUNÇÕES AGREGADORAS

Uma função de callback é uma função passada como argumento para outra função. Ela é executada após a conclusão de uma operação assíncrona ou como parte de um processo de maior ordem. Funções de callback são comuns em JavaScript, especialmente em operações assíncronas, como eventos, manipulação de arquivos. Elas permitem que você forneça um comportamento personalizado para ser executado quando uma determinada operação for concluída.

FUNÇÕES AGREGADORAS - JOIN

A função join é uma função que transforma nosso array em uma string. Esta função pode ser chamada de duas formas: Na primeira não são informados parâmetros e o retorno é uma string contendo todos os elementos do array separados por vírgula.
Na segunda maneira de ser chamada é informado um texto para servir como **separador**, e o retorno é semelhante à primeira forma, mas no lugar das vírgulas é exibido o separador indicado.

FUNÇÕES AGREGADORAS - JOIN

```
1 const arr = [1,2,3,4,5]
2 const arrManiulado = arr.join()
3 console.log(arrManiulado)
```

```
1,2,3,4,5
```

```
1 const arr = [1,2,3,4,5]
2 const arrManiulado = arr.join('-')
3 console.log(arrManiulado)
```

```
1-2-3-4-5
```

```
1 const fruits = ["Banana","Orange","Apple","Mango"]
2 let text = fruits.join(" and ")
```

```
Banana and Orange and Apple and Mango
```

FUNÇÕES AGREGADORAS - SLICE

A função slice é a função que vai "fatiar" nosso array, ela retorna uma parte do nosso array com base nos limites que foram passados.

A função recebe como parâmetros dois valores inteiros: O primeiro parâmetro indica a posição (índice) do primeiro elemento do array a fazer parte da seleção.

FUNÇÕES AGREGADORAS - SLICE

O segundo parâmetro indica a **posição** (índice) do último elemento a ser selecionado, porém, este último elemento não faz parte da seleção, mas, sim, o elemento que está uma posição atrás.

```
1 const arr = [1,2,3,4,5]
2 const arrFatiado = arr.slice(0,3)
3 console.log(arrFatiado)
```

```
[1, 2, 3]
```

```
1 const arr = [1,2,3,4,5]
2 const arrFatiado = arr.slice(0,4)
3 console.log(arrFatiado)
```

```
[1, 2, 3, 4]
```

FUNÇÕES AGREGADORAS - SPLICE

A função splice permite remover e/ou adicionar elementos em um array. O primeiro parâmetro indica a posição do elemento a ser removido, o segundo indica a quantidade de elementos a serem removidos, e para adicionar elementos, basta informar a posição e os elementos a serem adicionados.

```
1 const fruits = ["Banana","Orange","Apple","Mango","Kiwi"]
2 // Na osição 2, remove 2 elementos:
3 fruits.splice(2,2)
```

```
Banana,Orange,Kiwi
```

```
1 const fruits = ["Banana","Orange","Apple","Mango","Kiwi"]
2 // Na osição 2, adicione 2 elementos:
3 fruits.splice(2,0,"Lemon","Kiwi")
```

```
Banana,Orange,Lemon,Kiwi,Apple,Mango
```

IN

FUNÇÕES AGREGADORAS - FOREACH

A função “forEach” permite executar uma função para cada item do array.

A função “forEach” precisa da função “callback” como argumento obrigatório, também o argumento “this” que é um valor opcional a ser usado como “this” no momento que executar a função de “callback”.

FUNÇÕES AGREGADORAS - FOREACH

Para a função callback, temos como parâmetro:

1º parâmetro: É o valor da posição atual sendo percorrida no array. Parâmetro obrigatório na declaração a função.

2º parâmetro: É o índice, posição do array que está sendo lida. Parâmetro opcional da função.

3º parâmetro: É o array a ser percorrido no forEach(). Parâmetro opcional da função.

FUNÇÕES AGREGADORAS - FOREACH

```
1  const numbers = [65,44,12,4]
2  numbers.forEach(numbers)
3
4  function myFunction(item,index,arr) {
5      arr[index] = item * 10;
6  }
```

```
650,440,120,40
```

FUNÇÕES AGREGADORAS - MAP

A função map permite executar uma função para cada item do array.

A função map, assim como a função forEach precisa da função callback como argumento obrigatório, também o argumento this, opcional.

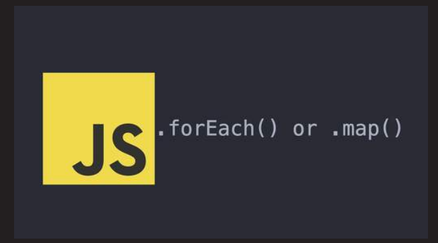
```
// Array de números
const numeros = [1, 2, 3, 4, 5];

// Função que multiplica cada número por 2
const multiplicarPorDois = numeros.map(numero => numero * 2);

// Função que calcula o quadrado de cada número
const calcularQuadrado = numeros.map(numero => numero ** 2);

// Exibindo os resultados
console.log("Array original:", numeros);
console.log("Cada número multiplicado por dois:", multiplicarPorDois);
console.log("Quadrado de cada número:", calcularQuadrado);
```

FUNÇÕES AGREGADORAS - MAP

```
1  const arr = ['a','b','c','d','e','f']
2
3  function tornarMaiuscula(elemento, indice, array){
4      return elemento.toUpperCase()
5  }
6
7  const newArr = arr.map(tornarMaiuscula)
8  // Retorna um novo array com todos os elmentos maiusculos
9
10 const newArr2 = arr.map(elemento=> elemento.toUpperCase())
11 // Retorna um novo array com todos os elmentos maiusculos
```

DIFERENÇA ENTRE FOREACH E MAP

Em JavaScript, as funções forEach e map são usadas para iterar sobre arrays, mas têm propósitos ligeiramente diferentes.



FOREACH

A função forEach **não retorna um novo array**, simplesmente executa a função callback fornecida em cada elemento do array, também não modifica o array original, é útil para realizar uma ação em cada elemento do array, como registrar informações, atualizar valores, etc.

MAP

A função map **cria e retornar um novo array** com os resultados da aplicação da função callback a cada elemento, também não modifica o array original, é útil quando você deseja transformar cada elemento de um array e coletar os resultados em um novo array.

FUNÇÕES AGREGADORAS - FILTER

A função filter() cria um novo array com todos os elementos que passaram no teste implementado pela função fornecida.

A função filter chama a função callback fornecida, uma vez para cada elemento do array, e constrói um novo array com todos os valores para os quais a função callback retornou o valor true ou um valor que seja convertido para true.

```
// Array de números
const numeros = [10, 25, 4, 30, 15, 8];

// Função que filtra números maiores que 10
const numerosMaioresQueDez = numeros.filter(numero => numero > 10);

// Função que filtra números pares
const numerosPares = numeros.filter(numero => numero % 2 === 0);

// Exibindo os resultados
console.log("Array original:", numeros);
console.log("Números maiores que 10:", numerosMaioresQueDez);
console.log("Números pares:", numerosPares);
```

FUNÇÕES AGREGADORAS - FILTER

Os elementos do array que não passaram no teste estabelecido na função callback são simplesmente ignorados, e não são incluídos no novo array.

```html
<!DOCTYPE html>
<html>
<head>
 <title>Página de Teste</title>


</head>
<body>
<p><input type="number" id="ageToCheck" value="30"></p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>

<script>
    const ages = [32,33,12,40]

    function checkAge(age){
        return age > document.getElementById("ageToCheck").value;
    }

    function myFunction(){
        document.getElementById("demo").innerHTML = ages.filter(checkAge);
    }
</script>
</html>
```

FUNÇÕES AGREGADORAS - REDUCE

A função reduce() é uma função built-in disponível para arrays.

Ela é usada para iterar sobre os elementos de um array e acumular um único valor aplicando uma função fornecida a cada elemento no array.
Isso pode ser útil para tarefas como: Somar todos os elementos, encontrar o valor máximo ou mínimo, ou qualquer outra operação que envolva a agregação de valores.

FUNÇÕES AGREGADORAS - REDUCE

A sintaxe básica do reduce possui 2 parâmetros, a função callback e o valor inicial.

callback: Uma função que é chamada para cada elemento no array. Ela recebe quatro argumentos: acumulador, elemento atual, índice atual e o próprio array.

```
// Array de números
const numeros = [5, 10, 15, 20];

// Função que realiza a soma dos elementos do array
const soma = numeros.reduce((acumulador, numero) => acumulador + numero, 0);

// Exibindo o resultado da soma
console.log("Array original:", numeros);
console.log("Soma dos números:", soma);
```

FUNÇÕES AGREGADORAS - REDUCE

Valor inicial: Um valor inicial opcional para o acumulador. Se não fornecido, o primeiro elemento do array é usado como valor inicial e a iteração começa a partir do segundo elemento.

```html
<body>
    <p id="demo"></p>

    <script>
        const numbers = [175, 50, 25];
        const result = numbers.reduce(myFunc); // Armazena o resultado da redução em uma variável

        function myFunc(total, num) {
            return total - num;
        }

        document.getElementById("demo").innerText = result; // Exibe o resultado na página
    </script>
</body>
```

100

ATIVIDADE PRÁTICA

Atividade 01

Faça um programa que mostre na tela o dobro de cada número do seguinte array [50, 45, 67, 89, 10, 5].

Atividade 02

Faça um Programa que mostre na tela apenas os número pares do seguinte array [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15].

ATIVIDADE PRÁTICA

Atividade 03

Faça um Programa que calcule a soma de todos os números do seguinte array [1, 2, 3, 4, 5, 6].

Atividade 04

Dada uma lista de números imprima cada elemento elevado ao quadrado.

ATIVIDADE PRÁTICA

Atividade 05

Dada uma lista de strings, use a função map para criar uma nova lista onde cada string é convertida para maiúsculas.

Atividade 06

Dada uma lista de strings, use a função filter para criar uma nova lista contendo apenas as palavras que têm mais de 5 caracteres.

DESAFIO PRÁTICO

Crie um sistema simples de blog onde os usuários podem visualizar postagens existentes, adicionar novas postagens e ver detalhes sobre cada postagem.

Funcionalidades:

Visualização de Postagens:

- Mostre ao usuário uma lista de postagens existentes com títulos e resumos.

Detalhes da Postagem:

- Ao clicar em uma postagem, exiba o conteúdo completo da postagem, incluindo a data de publicação.

DESAFIO PRÁTICO

Adicionar Nova Postagem:

- Permita que os usuários adicionem novas postagens. Cada postagem deve ter um título, conteúdo e data de publicação.

Comentários:

- Adicione a capacidade de os usuários deixarem comentários nas postagens.

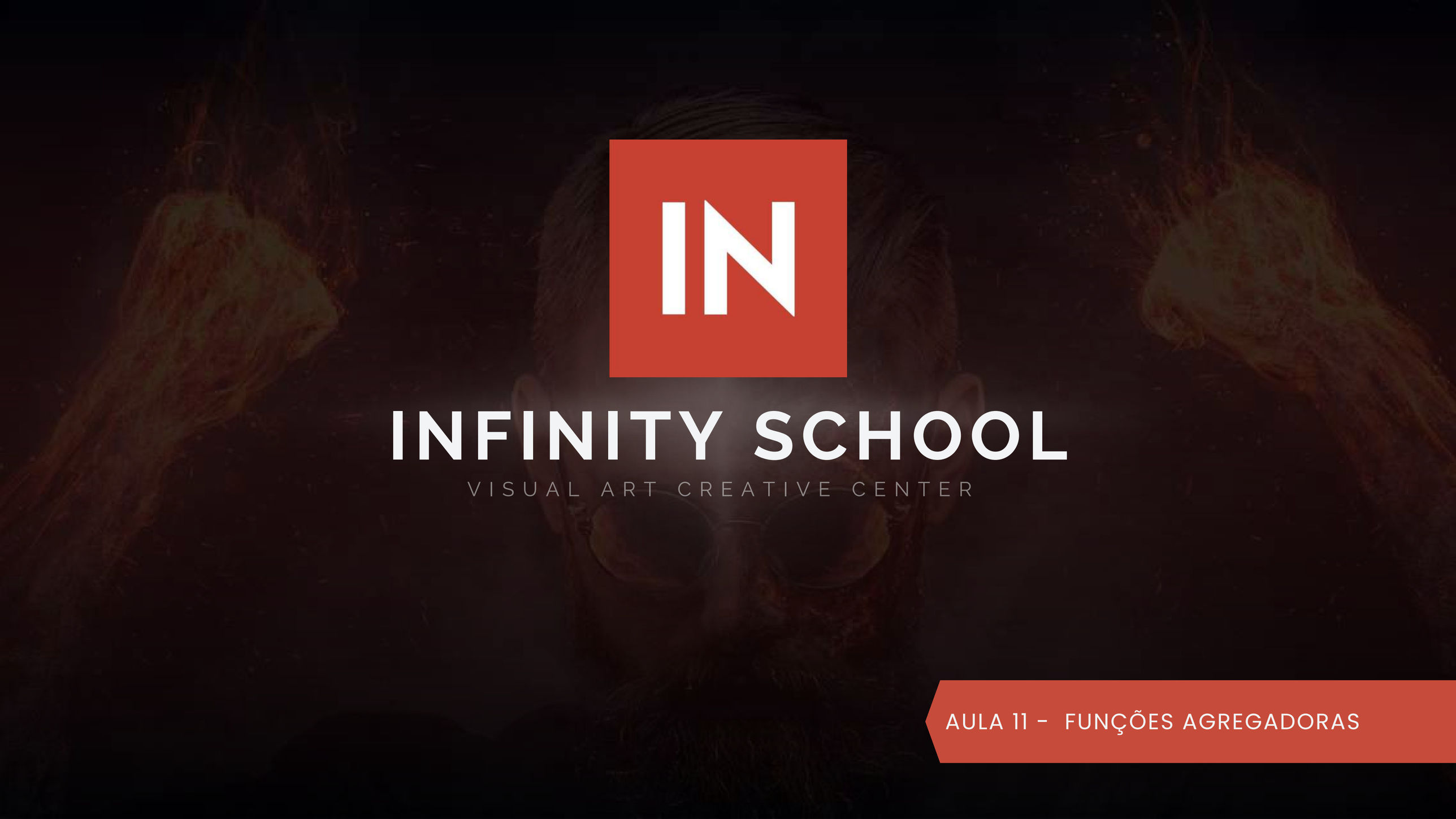
SE LIGA NO CONTEÚDO DA PRÓXIMA AULA!

AULA 12 DE JAVASCRIPT.
PROJETO

IN

INFINITY SCHOOL

VISUAL ART CREATIVE CENTER



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 11 - FUNÇÕES AGREGADORAS