# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 05 – ARRAY

# O QUE IREMOS APRENDER

01 RESUMO DA AULA PASSADA

02 VARIÁVEIS SIMPLES E COMPOSTAS

03 ARRAYS

04 MANIPULANDO ARRAYS

05 MÃOS NO CÓDIGO

# RESUMO DA AULA PASSADA

O loop for é uma estrutura de controle de fluxo. Ele é usado para repetir um bloco de código várias vezes com base em uma condição específica. O formato básico do for consiste em três partes:

Inicialização: Uma expressão que é usada para inicializar uma variável de controle.

Condição: Uma expressão avaliada antes de cada iteração do loop.

Incremento/Decremento: Uma expressão executada após cada iteração do loop.

# VARIÁVEIS SIMPLES E COMPOSTAS

Vimos em aulas anteriores o conceito de uma variável, e essas variáveis são variáveis simples, ou seja, elas conseguem armazenar apenas um valor por vez.

## Variáveis Simples:

Por exemplo, temos uma variável "nome" que guarda a string "Infinity", se atribuímos a ela outra string, como "Infinity School", a variável "nome" perde o valor "Infinity" e ganha o valor "Infinity school", já que a variável nome é uma variável simples.

# VARIÁVEIS SIMPLES E COMPOSTAS

## Variáveis Compostas:

Já as variáveis compostas possuem a capacidade de armazenar mais de um valor em uma mesma estrutura. Ou seja, se nessa variável composta atribuímos o valor "Infinity" e atribuírmos outro valor "Infinity School", essa variável será capaz de armazenar ambos os valores.

# ARRAYS

No JavaScript, o primeiro tipo de variável composta que vamos estudar é chamado de Array.
Para criar um array no JavaScript, utilizamos pares de colchetes []. E então, cada valor dentro dos [] é separado por vírgula.

# ARRAYS

```
let nome = "Pedro" // Variável Simples

let nome2 = "Marcela" // Variável Simples

let nomes = ["Marcela", "Pedro"] // Variável Composta

/*
Perceba que tanto a string Marcela quanto a string
Pedro estão armazenas na variável "nomes"
*/
```

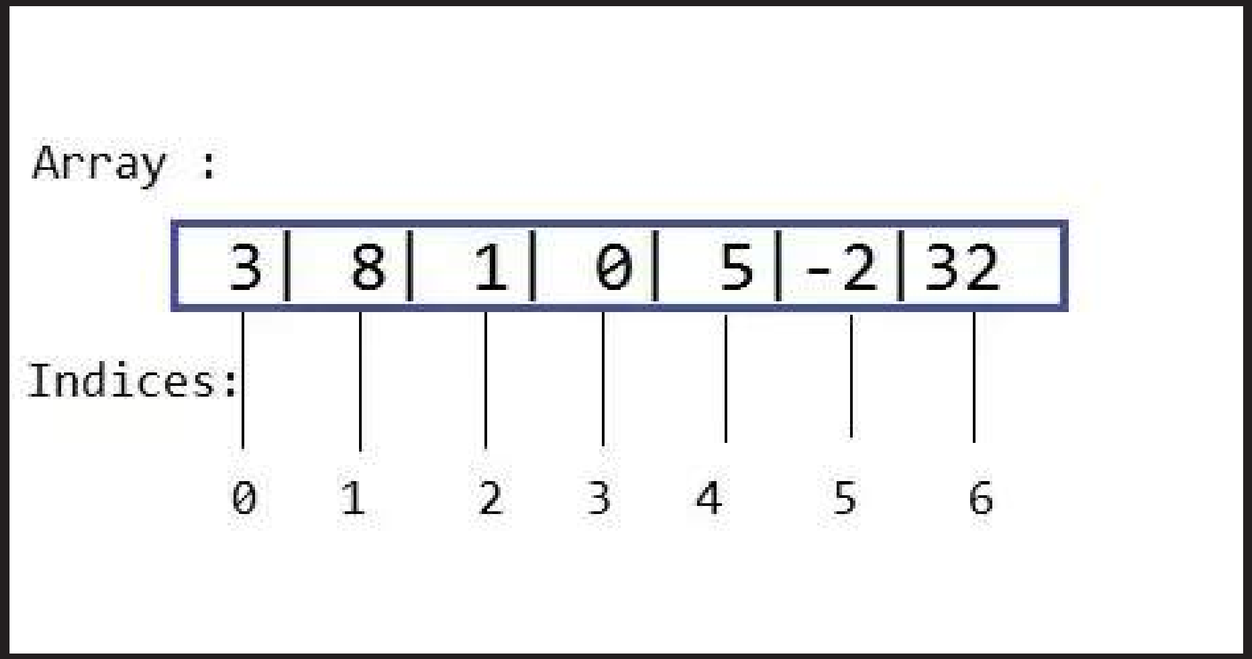
# ARRAYS

Dentro das variáveis compostas, cada valor salvo está ocupando um espaço, e cada espaço desse é identificado por um número que chamamos de índice.

É como se tivéssemos uma lista de compra e cada item da lista a gente colocasse um número:
1. Arroz, 2. Feijão, 3. Sal
Dizemos que o índice do item arroz é 1, o índice do item feijão é 2, o índice do item sal é 3.

# ARRAYS

Lembrando que no JavaScript, o índice se inicia sempre pelo valor 0. Ou seja, se temos 10 elementos dentro de um array, os índices desses elementos serão de 0 a 9 e não de 1 a 10.



# ARRAYS

Imagine que você está desenvolvendo um sistema que necessite guardar o nome de 5 pessoas, para fazer isso, criaríamos 5 variáveis para guardar esses nomes, porém, pense agora que fosse necessário guardar o nome de 1000 pessoas, criaríamos 1000 variáveis...
Com as variáveis compostas, podemos armazenar quantos nomes fossem necessários em apenas uma linha de código.

```
let vetor = [] // Array vazio

let vetor2 = [2 , 0 , 3 , 2] // Array com 4 elementos, índices de 0 a 3

let vetor3 = ['r', 'a', 'a', 'm', 'a'] // Array com 5 elementos, índices de 0 a 4
```

# ARRAYS

O array possui uma característica importante, ele é mutável, ou seja, conseguimos, por exemplo, adicionar valores a ele depois de criado.
Para fazer isso, podemos explicitar a posição (o índice) que queremos adicionar aquele valor.

É importante salientar que, caso adicionarmos valor em um índice que já está ocupado por outro, o valor antigo se perde e o novo o substituirá.

# ARRAYS

## Exemplo

```
let array = [1, 2, 3] // Array com 3 elementos, índices de 0 a 2

array[3] = 4
/*
Adicionando o número 4 no índice 3
*/

// O array atualizado: [1, 2, 3, 4]

array[0] = 7

// O valor 1 será substituído pelo valor 7
```

# Manipulando Arrays

O JavaScript nos dá também a função push, que adiciona um valor ao final do array.
A própria função identifica qual é a última posição para adicionar esse elemento.

```
let array = [10, 20, 30, 40] // Array com 4 elementos, índice de 0 a 3

array.push(50) //Adicionando o número 50 no índice 4
```

# Manipulando Arrays

O JavaScript nos dá também a função unshift, que adiciona um valor na primeira posição do array.

Passamos o valor a ser adicionado.

```
let array = [20, 30, 40] // Array com 3 elementos, índice de 0 a 2

array.unshift(10) // Agora o array possui os elementos: [10,20,30,40]
```

# Manipulando Arrays

Temos também o comando pop, que remove o último elemento do array, e esse comando sim, altera o tamanho do array, ou seja, ele exclui o elemento e também exclui a "vaga".

```
let array = [20, 30, 40] // Array com 3 elementos, índice de 0 a 2

array.pop() // o array atualizado: [20,30]
```

# Manipulando Arrays

O método splice é comumente utilizado em muitas linguagens de programação, especialmente em JavaScript. Ele é usado para modificar o conteúdo de um array, removendo ou substituindo elementos existentes e/ou adicionando novos elementos.

```
array.splice(start, deleteCount, item1, item2, ...)
```

start: Índice no qual começar a modificar o array.

deleteCount: Número de elementos a serem removidos a partir do start.
item1, item2, ...: Elementos a serem adicionados ao array no start.

# Manipulando Arrays

## Exemplo

Substituir elementos de um array

```
let array = [1, 2, 3, 4, 5];
array.splice(2, 2, 6, 7);  // Remove 2 elementos a partir do índice 2 e adiciona 6 e 7
console.log(array);  // Resultado: [1, 2, 6, 7, 5]
```

Remover elementos de um array

```
let array = [1, 2, 3, 4, 5];
array.splice(2, 2);  // Remove 2 elementos a partir do índice 2
console.log(array);  // Resultado: [1, 2, 5]
```

Adicionar elementos de um array

```
let array = [1, 2, 3, 4, 5];
array.splice(2, 0, 6, 7);  // Adiciona 6 e 7 no índice 2 sem remover nenhum elemento
console.log(array);  // Resultado: [1, 2, 6, 7, 3, 4, 5]
```

IN

# Manipulando Arrays

O JavaScript também nos dá uma maneira de saber o tamanho daquele array, ou seja, quantos elementos existem dentro daquele array. Ele faz isso através do atributo length.

```
let array = [20, 30, 40] // Array com 3 elementos, índice de 0 a 2

array.length //Esse comando retorna o valor 3
```

# Manipulando Arrays

O JavaScript também nos dá uma maneira de ordenar um array. Ele nos dá esse recurso através da função sort.

```
let array = [6, 8, 3, 1 , 4, 0]

array.sort() //Essa função retorna um array ordenado
```

# Manipulando Arrays

Temos uma função em JavaScript que retorna o índice de determinado elemento. Chamamos a função indexOf e passamos o elemento, e ela retorna pra gente o índice daquele elemento. Caso esse elemento não exista dentro do array, a função retorna pra gente o valor -1.

```
let array = [6, 8, 3, 1 , 4, 0]

array.indexOf(3) //Essa função retorna 2

array.indexOf(18) //Essa função retorna -1
```

# Manipulando Arrays

Vimos na aula passada que podemos utilizar a estrutura de repetição for of para percorrer uma string, em JavaScript podemos percorrer um array também utilizando a estrutura for of.

```
let array = ['i', 'n', 'f', 'i', 'n', 'i', 't', 'y']

for (item of array){
    console.log(item)
    console.log('Passei por mais um item')
}
// A cada iteração, a variável "item" recebe um elemento do array como valor
```

# ATIVIDADE PRÁTICA

**Atividade 01**

Faça um programa que leia 20 números inteiros e armazene-os em um vetor. Armazene os números pares no vetor 'PAR' e os números ímpares no vetor 'ímpar'. Imprima os três vetores.

**Atividade 02**

Faça um Programa que leia um vetor de 5 números inteiros e mostre-os.

# ATIVIDADE PRÁTICA

**Atividade 03**

Utilizando listas, faça um programa que realize 5 perguntas para uma pessoa sobre um crime, ela deve responder com prompt S - Sim e N - Não. As perguntas são:

1 - "Telefonou para a vítima?"
2 - "Esteve no local do crime?"
3 - "Mora perto da vítima?"
4 - "Devia para a vítima?"
5 - "Já trabalhou com a vítima?"

Ao final, o programa deve emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões, ela deve ser classificada como "Suspeita"; entre 3 e 4 respostas positivas como "Cúmplice"; e se responder positivamente a 5 questões, deve ser classificada como "Assassino". Caso contrário, será classificada como "Inocente".

# ATIVIDADE PRÁTICA

**Atividade 04**

Faça um Programa que leia um vetor de 10 números reais e mostre-os na ordem inversa.

**Atividade 05**

Faça um Programa que leia 4 notas, mostre as notas e a média na tela.

# DESAFIO PRÁTICO

Uma empresa de pesquisas precisa tabular os resultados da seguinte enquete feita a um grande quantidade de organizações:
"Qual o melhor Sistema Operacional para uso em servidores?" As possíveis respostas são:

1- Windows Server
2- Unix
3- Linux
4- Netware
5- Mac OS
6- Outro

# DESAFIO PRÁTICO

Você foi contratado para desenvolver um programa que leia o resultado da enquete e informe ao final o resultado da mesma. O programa deverá ler os valores até ser informado o valor 0, que encerra a entrada dos dados. Não deverão ser aceitos valores além dos válidos para o programa (0 a 6). Os valores referentes a cada uma das opções devem ser armazenados num vetor. Após os dados terem sido completamente informados, o programa deverá calcular a percentual de cada um dos concorrentes e informar o vencedor da enquete. O formato da saída foi dado pela empresa, e é o seguinte:

# DESAFIO PRÁTICO

| Sistema Operacional | Votos | % |
| --- | --- | --- |
| Windows Server 1500 | | 17% |
| Unix | 3500 | 40% |
| Linux | 3000 | 34% |
| Netware | 500 | 5% |
| Mac OS | 150 | 2% |
| Outro | 150 | 2% |
| | Total : | 8800 |

O Sistema Operacional mais votado foi o Unix, com 3500 votos, correspondendo a 40% dos votos.

# SE LIGA NO CONTEÚDO DA PRÓXIMA AULA!

AULA 06 DE JAVASCRIPT.
FUNÇÔES I

IN

INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

# FUNÇÕES

Funções em JavaScript são blocos de código que realiza uma tarefa, uma operação, sendo executada quando é chamada por alguém ou invocada.

Elas permitem que você agrupe um conjunto de instruções em uma única unidade lógica, tornando seu código mais organizado, modular e fácil de manter.

# FUNÇÕES

Exemplos de **Funções** em JavaScript:

```
<script>
    /*
    Para criar uma função basta colocar a palavra reservada function,
    e posteriormente o nome da função. Dentro dos parênteses colocamos
    os parâmetros(os parâmetros são opcionais), abre chaves e dentro das
    chaves eu coloco meu código da função.
    */

    function funcaoExemplo(parametro1,parametro2){
        //codigo
    }
</script>
```

# INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 05 – ARRAY