

---

# Autoria Web

**Professor: Diego Oliveira**



**Aula 15 – JavaScript5**





# Argumentos de Funções

- Podemos passar valores para as funções
- Esses valores são os argumentos
- Como vimos na atividade da aula passada, argumentos possibilitam uma melhoria de código e evitam trabalho desnecessário:

```

```

```
function reservar(num) {  
    document.getElementById("cadeira"+num).src="ocupada.png";  
    cadeiras[num] = 1;  
}
```





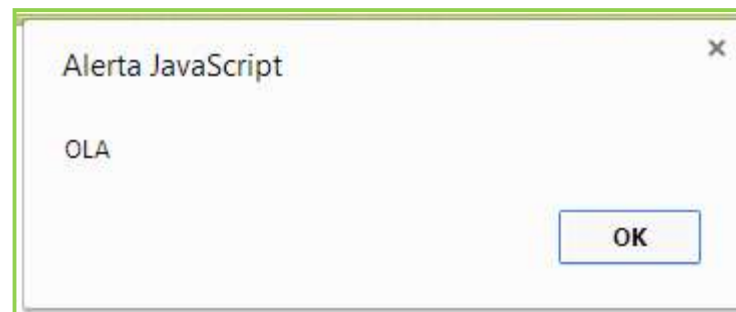
# Retorno de Funções

- Permitem que as funções retornem valores para outras
- Retornos podem ser de qualquer tipo:

```

```

```
<script type="text/javascript">  
  function cheocar() {  
    return "OLA";  
  }  
</script>
```





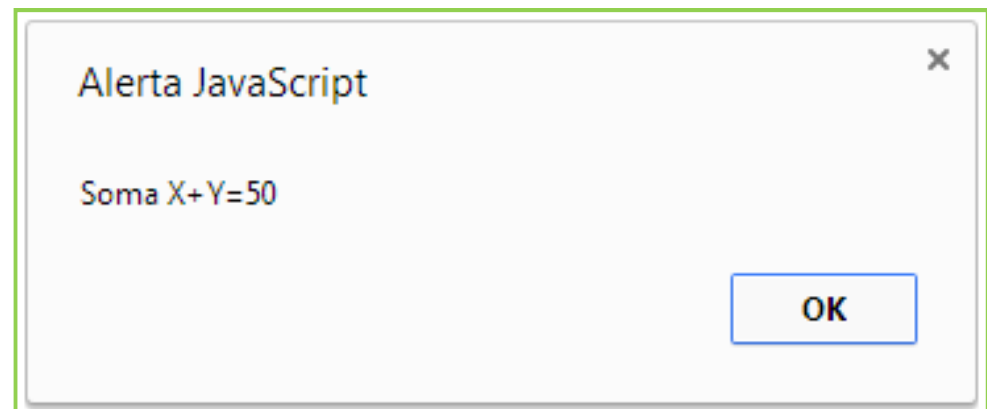
# Retorno de Funções

- Outros exemplos de retorno:

```

```

```
function getX() {  
    var x = 20;  
    return x;  
}  
function getY() {  
    var y = 30;  
    return y;  
}  
function soma() {  
    return getX() + getY();  
}
```





# Eventos

- Nós já aprendemos:
  - onclick()
  - onload()
- Há outros dois:
  - onchange()
  - onblur()

```
function mudou () {  
    alert ("ONCHANGE!");  
}  
  
function mudou2 () {  
    alert ("ONBLUR!");  
}
```

```
<input id="nome" type="text" value="nada"  
    .....  
    .....  
    onchange="mudou () " onblur="mudou2 () ">
```





# Validando Campos

- Campo vazio:

Sobrenome: `<input id="nome2" type="text" onblur="verificarVazio(this)">`

```
function verificarVazio(campo) {  
    if(campo.value.length == 0) {  
        alert("Preencha seu sobrenome");  
    }  
}
```

- Campos numéricos:

Salário: `<input id="salario" type="text" onblur="verificarNumero(this)">`

```
function verificarNumero(campo) {  
    if(isNaN(campo.value)) {  
        alert("Digite números para o salário!");  
    }  
}
```





# Atividade

- Crie um formulário de cadastro de cliente com:
  - Dados Pessoais
  - Endereço
  - Dados financeiros (salário, emprego, empresa)
- Para cada campo coloque validadores que não permitam campos vazios
- Nos campos de valores, coloque validadores para verificar se foram digitados números





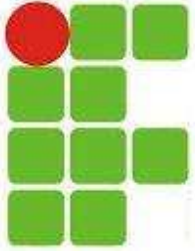
# Arrays

- As vezes precisamos utilizar várias variáveis para armazenar dados
- Exemplo:
  - Em um cinema, deseja-se armazenar quais das 10 poltronas estão ocupadas e quais não estão ->

```
<script type="text/javascript">  
    function testeArrays() {  
        var polt01 = true;  
        var polt02 = true;  
        var polt03 = false;  
        var polt04 = true;  
        var polt05 = false;  
        var polt06 = true;  
        var polt07 = false;  
        var polt08 = true;  
        var polt09 = true;  
        var polt10 = true;  
    }  
</script>
```







# Arrays

- Os Arrays são variáveis, que possuem apenas um nome, porém podem armazenar vários valores como se fosse um conjunto de variáveis
- Os Arrays são tuplas, do tipo chave-valor
- As chaves geralmente são números que iniciam com ZERO
- As chaves numéricas são conhecidas como índice, e os Arrays como Arrays Indexados





# Arrays

- Para criar um novo Array, fazemos assim:

```
var meuArray = new Array();
```

- Para armazenar valores no Array, basta utilizar os seus índices:

```
meuArray[0] = "teste";  
meuArray[1] = true;  
meuArray[2] = 20;
```





# Arrays

- Arrays podem ser inicializados no momento da sua criação:

```
var professoresInfo =  
    ["diego", "daniel", "adorilson",  
     "felipe", "igor", "augusto", "marcos"];
```

- Para acessar os valores de um array em um FOR, fazemos assim:

```
for(var j=0; j<professoresInfo.length; j++){  
    alert("professoresInfo["+j+"] = " + professoresInfo[j]);  
}
```





# Arrays Bidimensionais

- Basicamente é um Array de Arrays
- Muitas vezes é chamado de matriz
- Como criar:

```
var matriz = new Array(new Array(3), new Array(3), new Array(3));
```

- Como inicializar:

```
var matriz2 = [[1,2,3],[4,5,6],[7,8,9]];
```





# Arrays Bidimensionais

- Como percorrer um Array bidimensional:

```
var matriz2 = [[1,2,3],[4,5,6],[7,8,9]];

for(var x=0; x<matriz2.length; x++){
    for(var y=0; y<matriz2[x].length; y++){
        alert("matrix["+x+"]["+y+"] = " + matriz2[x][y]);
    }
}
```

- Array tridimensional:

```
var matriz3 = [[[1,2,3],[1,2,3],[1,2,3]],[[1,2,3],[1,2,3],[1,2,3]],[[1,2,3],[1,2,3],[1,2,3]]];

for(var a=0; a<matriz3.length; a++){
    for(var b=0; b<matriz3[a].length; b++){
        for(var c=0; c<matriz3[a][b].length; c++){
            alert("matrix["+a+"]["+b+"]["+c+"] = " + matriz3[a][b][c]);
        }
    }
}
```



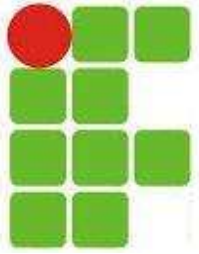


# Operadores Lógicos

- AND: é representado pelo operador `&&`
- Tabela verdade do AND:

p	q	p && q
true	true	true
true	false	false
false	true	false
false	false	false



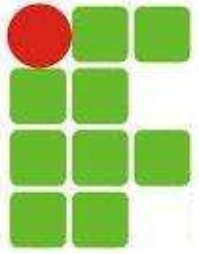


# Operadores Lógicos

- OR: é representado pelo operador ||
- Tabela verdade do OR:

p	q	p    q
true	true	true
true	false	true
false	true	true
false	false	false





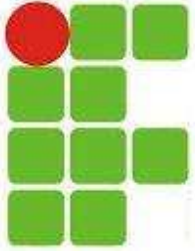
# Operadores Lógicos

- NOT: é representado pelo operador !
- Tabela verdade do NOT:

p	!p
true	false
false	true





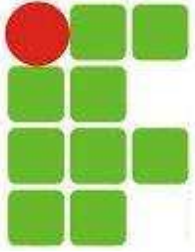


# Atividade

- Coloque 10 imagens de cadeiras livres no HTML, representando um cinema vazio
- Ao clicar em uma imagem, trocá-la por uma cadeira ocupada, representando uma reserva de lugar
- Coloque um botão para verificar quantos lugares vazios ainda existem e mostre em um alert();



- Opcionalmente, coloque um botão de reset



---

# Perguntas?

---

