

# JavaScript

=D

# O que é?

- Linguagem de programação interpretada
- **Extremamente versátil** - faz de tudo!
- Predominantemente client-side
- Orientada a objetos
- Tipagem fraca

# Para que serve?

- Manipular elementos de um layout HTML
- Detectar eventos do usuário
- Permite carregamento dinâmico
- Scripts offline

# Onde fica o JavaScript?

O JavaScript é carregado junto com o HTML, sendo executado no momento da leitura do código pelo navegador, ou conforme definido pelo desenvolvedor.

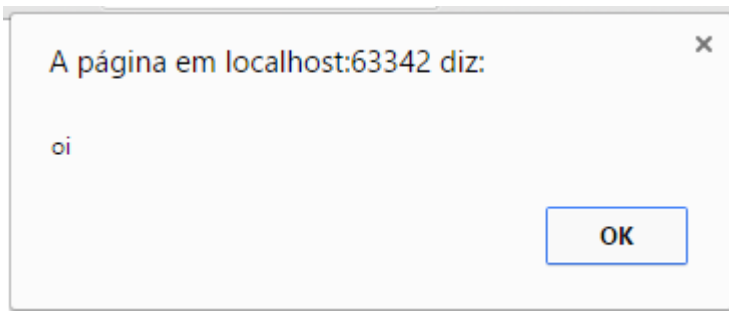
Existem 2 modos de incluir JavaScript à um documento HTML:

- Internamente (via tag)
- Externamente (via arquivo .js)

# JavaScript Interno

O script JS é envolto por uma tag `<script>`, dentro do documento HTML.

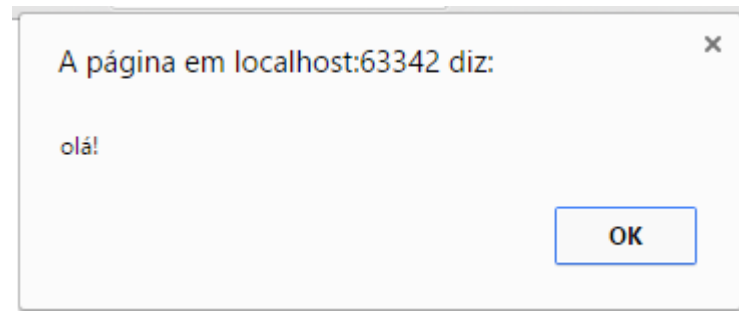
```
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Exemplo JS</title>
  <script>
    alert('oi');
  </script>
</head>
<body>
</body>
</html>
```



# JavaScript Externo

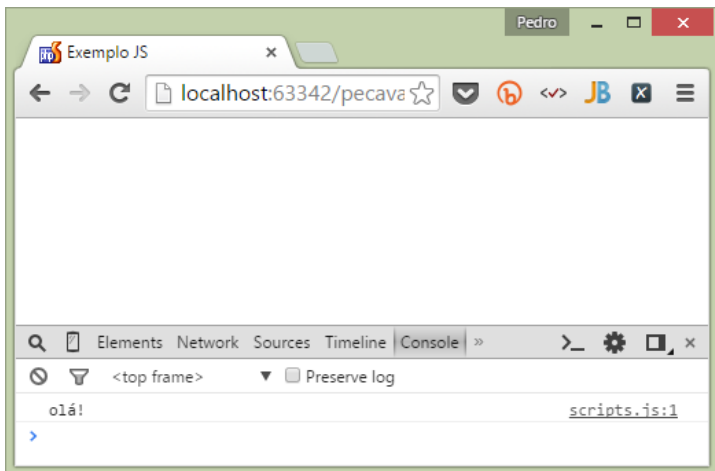
Um arquivo com scripts JS é vinculado ao documento HTML, isolando as linguagens. É o modo recomendado.

```
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>Exemplo JS</title>
  <script src="js/scripts.js" type="application/javascript"></script>
</head>
<body>
</body>
</html>
```

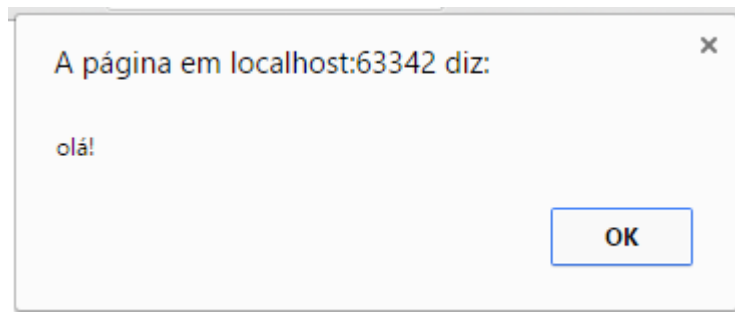


# Auxiliares para debug

`console.log('olá!');`



`alert('olá!');`



# Variáveis

As variáveis são declaradas com a palavra chave `var`, como segue:

```
var i;  
var sum;
```

Também é possível declarar varias variáveis com a mesma palavra chave `var`:

```
var i, sum;
```

E pode-se combinar a declaração da variável com sua inicialização:

```
var i = 0, j = 0, k = 0;  
var nome = "Fulano";
```

Antes de usar uma variável em um programa JavaScript, você deve declará-la.



# Tipos

Os tipos em JavaScript podem ser divididos em duas categorias:

- tipos primitivos
- tipos de objeto

Os tipos primitivos incluem números, strings e valores booleanos.

Os valores especiais `null` e `undefined` são valores primitivos mas não são números, nem string e nem booleanos.

Qualquer valor em JavaScript que não seja número, string, booleano, `null` ou `undefined` é um objeto.

# Condicional - if...else

```
if (n == 1) {  
    // executa este bloco if  
} else if (n == 2) {  
    // executa este bloco else if  
} else {  
    // executa este bloco else  
}
```

# Condicional - switch

```
switch(sentimento) {  
    case 'fome':  
        comer();  
        break;  
    case 'sono':  
        dormir();  
        break;  
    default:  
        vadiar();  
}
```

# For

```
for (var i = 0; i < 5; i++) {  
    // será executado 5 vezes  
}
```

# Arrays

Em JavaScript, **arrays** são um tipo especial de objeto que representam um conjunto ordenado de valores numerados.

```
var a = new Array();  
a[0] = "dog";  
a[1] = "cat";  
a[2] = "hen";  
a.length // 3
```

Uma forma mais conveniente de utilização de um array, mais utilizada:

```
var a = ["dog", "cat", "hen"];  
a.length // 3
```

# Funções

Uma **função** é um objeto que tem código executável associado. Uma função pode ser chamada para executar esse código executável e retornar um valor calculado.

```
function soma(x, y) {  
    var total = x + y;  
    return total;  
}
```

# jQuery

- jQuery é uma biblioteca JavaScript cross-browser desenvolvida para simplificar os scripts client side que interagem com o HTML
- facilita \*horrores\* a vida do desenvolvedor
- coleção de funções e métodos prontos para serem utilizados
- mantra “Write less, do more” (Escreva menos, faça mais)
- Usa os mesmos seletores usados no CSS
- <https://jquery.com/>

# Como usar?

## Inserindo o jQuery local

```
<script type="text/javascript" src="js/jquery-X.X.X.min.js"></script>
```

## Inserindo o jQuery hospedado no Google (Google Hosted Libraries)

```
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/X.X.X/jquery.min.js"></script>
```



# Carregar quando...

```
// Quando o site terminar de carregar...  
$(document).ready(function() {  
    alert('carregou!');  
    // Outros scripts  
});
```

# Capturar eventos - mouse

```
$( 'button.enviar' ).click(function() {  
    form.submit();  
});
```

```
$( 'img' ).hover(function() {  
    $(this).fadeOut();  
});
```

```
$( 'input.cep' ).blur(function() {  
    validaCEP();  
});
```

```
$( 'select.cidade' ).change(function() {  
    mostraBairros();  
});
```

# Capturar eventos - teclado

```
$( 'input.nome' ).keydown(function() {  
    validaCaracter();  
});
```

```
$( 'input.nome' ).keyup(function() {  
    limpaNome();  
});
```

```
$(input.fone).keypress(function() {  
    validaTelefone();  
});
```

# Manipular propriedades

```
$( 'elemento' ).append(outroElemento);
```

```
$( 'elemento' ).attr( 'atributo' );
```

```
$( 'elemento' ).css( 'propriedade' );
```

```
$( 'elemento' ).html();
```

```
$( 'elemento' ).val();
```

```
$( 'elemento' ).remove();
```

```
$( 'elemento' ).hide();
```

```
$( 'elemento' ).height();
```

```
$( 'elemento' ).addClass( 'classe' );
```

```
$( 'elemento' ).hasClass( 'classe' );
```

```
$( 'elemento' ).fadeOut();
```

```
$( 'elemento' ).width();
```

```
$( 'elemento' ).show();
```

```
$( 'elemento' ).fadeIn();
```

# Exercícios

1. Crie um script que leia um número inteiro (de um input) e calcule a soma de todos os seus antecessores (até 0), exibindo a resposta no final.
2. A partir de um form, criar um script que leia um valor em reais e a quantidade de parcelas desejadas. Ao clicar em calcular, mostrar o valor de cada parcela.
3. A partir do exercício anterior, adicionar juros quando a quantidade de parcelas for  $>4$ , sendo a taxa 1,2% por parcela adicional. Exibir o valor total e das parcelas com acréscimo.
4. A partir de um input text e um botão, criar um script disparado no evento click para calcular quantas notas/moedas de R\$ 100, 50, 20, 10, 5, 2 e 1 serão emitidas, em ordem decrescente. Exibir na tela a quantidade respectiva de cédulas.
5. A partir de um formulário com os campos nome, nome do produto, valor e quantidade desejada, criar um script que, a cada clique, lista o item informado, mostrando o valor total até o momento, limpando o formulário.
6. Num form, ler o nome de uma pessoa. Após a leitura do nome, ocultar o campo de nome e exibir campos de produto, valor e quantidade. Após ler 3 produtos, ocultar os campos, mostrando o resumo da pessoa e produtos. Mostrar um novo campo para informar a quantidade de parcelas, e um botão para calcular. Mostrar o valor da parcela após o cálculo.