

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SCRIPT</title>
  <script>
    alert('A');
  </script>
  <style>
    p{
      font-weight: bold;
    }
  </style>
</head>
<body>
```

`<!-- OBJETIVO:`

A tag "script" ou "escritura", é a tag que define um script JS para adicionar um determinado comportamento a nossa página, ela pode ser atribuída tanto a área "head" de uma página, quanto a "body" (Muito embora não seja recomendado utilizar a tag "script" dentro da área "head", pois isso pode dificultar a renderização da nossa página por causa dos SEO's, o ideal é utilizar a tag "script" ao final do body, depois de todo o conteúdo da nossa página ter sido renderizado.). Dentro da tag "script" podemos escrever códigos em JS na própria página, que serão executados no próprio navegador do cliente. Quanto podemos apontar para um arquivo JS que será executado a partir de um servidor. A tag "script" pode receber os seguintes atributos:

=> async:

"Async" ou "Assíncrono", é um atributo da tag "script" que faz com que o nosso script funcione de forma assíncrona em comparação com o navegador do lado do cliente. OBSERVAÇÕES: esse atributo só funciona em arquivos externos. Usamos "async" dessa forma: `<script async>`.

=> charset:

Definire o padrão de caracteres que será utilizado no nosso script, o padrão é utilizar o UTF-8. Usamos "charset" dessa forma: `<script charset="UTF-8">`.

=> defer:

"defer" ou "adiamento", é um atributo que define que um script seja carregado somente depois da página ser carregada, isso ajuda na velocidade de renderização de uma página. Usamos "defer" dessa forma: `<script defer>`.

=> src:

Ou "source", é um atributo que aponta para um URL onde o nosso arquivo JS se encontra, esse atributo fará com que o nosso script seja carregado mesmo não estando presente dentro da nossa página html. Usamos "src" dessa forma: <script src="url... ">.

=> type:

Definine o tipo de arquivo que estamos usando no nosso script, o estilo padrão é o "text/javascript", mesmo se não usarmos esse atributo, o html já entenderá que é ele que deverá ser utilizado. Usamos "type" dessa forma: <script type="javascript">.

OBSERVAÇÕES: Por padrão, o estilo CSS do "script" é:

```
style {  
    display: none;  
}
```

-->

<!-- EXEMPLO:

Nesse exemplo estamos usando scripts de 4 formas:

1º script "A": usado na head, sendo carregado antes da página;

2º script "B": usado no top da body, sendo carregado antes do conteúdo da página;

3º script "C": usado depois das 2 tags "p", só é carregado antes por que é um alert;

4º script "D": usado depois das 2 tags "p" e de forma externa, é um alert, deveria ter sido declarado antes, mas, por ser externo perde tempo em renderização;

5º script usando defer: é um script que tem como função pegar a tag "h1 e mostra no console, não era para ele conseguir isso, afinal, ele foi declarado antes da "h1". Porém, por usamos o atributo "defer" conseguimos adiar a execução desse script e executá-lo somente depois que "h1" é renderizado na página. Como prova de comparação, retire o atributo "defer" e veja...;

-->

```
<script>alert('B')</script>
```

<p>Só fui executado depois do script que está na área "head", o script "A". Depois só fui executada depois do script que está na área body antes de mim, o script "B" e só depois o script "C" que está depois de mim, veio aparecer, e isso só aconteceu por que ele é um "alert"!</p>

<p>Detalhe, o script "D" foi o último, e ele foi escrito em arquivo externo, por demorar mais para renderizar, consegui ser visualizado antes de o script "D" aparecer.</p>

```
<script>alert('C')</script>
```

```
<script src="016-script.js"></script>
```

```
<script src="016-defer.js" defer></script> <!-- O "defer" recebe com valor o próprio "defer" -->
```

```
<h1>O defer que me pegou veja o console...</h1>
```

```
</body>
```

```
</html>
```