

MÓDULO

JAVASCRIPT/AJAX

PRÁTICA

CRIAÇÃO DE UM RELÓGIO AUTOMÁTICO
RECORRENDO A JAVASCRIPT E A SUA
ESTILIZAÇÃO ATRAVÉS DE CSS

CRIAÇÃO DE UM RELÓGIO AUTOMÁTICO RECORRENDO A JAVASCRIPT E A SUA ESTILIZAÇÃO ATRAVÉS DE CSS

1.1. DESCRIÇÃO DA PRÁTICA

Tempo estimado: 1 h.

REQUISITOS
■ Ter concluído a unidade didática “Folhas de estilo CSS e JavaScript”.
OBJETIVOS
■ Compreender melhor a utilização de objetos e funções e como podem resultar numa simples aplicação dinâmica.
■ Compreender a aplicação de estilos dinâmicos através de CSS.
MATERIAL E FERRAMENTAS NECESSÁRIAS PARA A PRÁTICA
■ Editor de código.

1.2. DESENVOLVIMENTO DA PRÁTICA

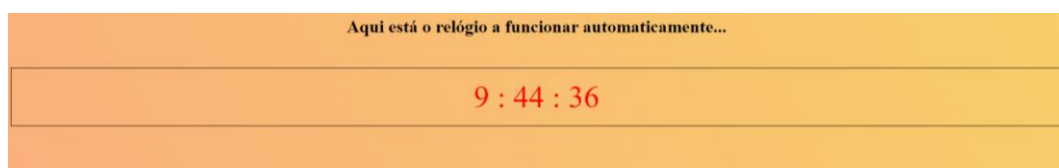
Nesta prática, irá criar um simples relógio, mas será um relógio totalmente automático. Relembrará o objeto `date` para o relógio assumir a data atual e utilizará o `setInterval` para que o relógio se atualize automaticamente.

Este tipo de lógica é também muito utilizado para, por exemplo, contagem decrescente para saldos num website e para atualizar a data (dia, mês e ano) no mesmo, entre outros.

A visualização será:



Uma particularidade desta prática é que, ao passar com o rato sobre o relógio, o mesmo muda de cor e aumenta o tamanho de letra, voltando ao estado original quando se retira o rato.



A prática inicia-se com a abertura do editor de código, gravação do ficheiro em formato HTML e inserção do código base, que já conhece, onde irá colocar o código JavaScript do relógio.

```
<!doctype html>

<html>

<head>

<meta charset="UTF-8">

<title>

Relógio com Javascript

</title>

    <script type="text/javascript">

    </script>

</head>

<body>

</body>

</html>
```

Podemos já incluir no <body> do documento os elementos necessários. Assim, inclui-se um título e um parágrafo com id relógioAuto. Este id vai ser necessário para utilizar na definição das funções em JavaScript. Também no parágrafo, vão já incluir-se os efeitos onmouseover e onmouseout, chamando, respetivamente, as funções mouseOver e mouseOut, que serão definidas posteriormente e necessárias para adicionar estilos dinâmicos ao elemento.

```
<body>

    <h1>Aqui está o relógio a funcionar automaticamente... </h1>

    <p id="relogioAuto" onmouseover="mouseOver()" onmouseout="mouseOut()"></p>

</body>
```

Dentro da tag <script> irá começar por criar a função responsável por devolver a data atual.

```
function relógio() {}
```

Esta função terá de incluir uma variável para criar o objeto e mais três variáveis que funcionam quase como “parâmetros em separado” da primeira variável, ou seja, embora sejam variáveis em separado, elas dependem da variável “atual” para devolverem a data atual. Assim, ficamos com:

```
function relógio() {  
    atual = new Date();  
    hora = atual.getHours();  
    minuto = atual.getMinutes();  
    segundo = atual.getSeconds();  
}
```

Basicamente a função “relógio” define a hora atual, mas ainda não devolve nenhum resultado. Para isso, irá acrescentar uma outra variável que irá conter as variáveis das horas, minutos e segundos, separados por dois pontos (:) para ficar mais idêntico a um relógio.

```
imprimirHora = hora + " : " + minuto + " : " + segundo;
```

Para que o JavaScript devolva as horas no documento, irá utilizar um parágrafo vazio (com o id “relógioAuto”) no body, a que irá ter acesso através do getElementById e imprimir o resultado usando o .innerHTML.

O aspeto final da função deverá ser semelhante a:

```
function relógio() {  
    atual = new Date();  
    hora = atual.getHours();
```

```
minuto = atual.getMinutes();  
  
segundo = atual.getSeconds();  
  
imprimirHora = hora + " : " + minuto + " : " + segundo;  
  
document.getElementById("relogioAuto").innerHTML =  
imprimirHora;  
  
}
```

De seguida, iremos converter esta função estática numa função que se atualize, segundo a segundo, devolvendo sempre a hora atualizada. Para isso, basta criar mais uma variável, com um `setInterval`, para a função “relogio”, que se atualize segundo a segundo, ou seja, no parâmetro irá indicar 1000 porque o JavaScript trata todas as datas em milissegundos. Esta variável será definida pela seguinte linha de código e deverá ser incluída antes da função “relogio”.

```
var loopRelogio = setInterval(relogio, 1000);
```

Falta ainda definir as funções que irão dar os estilos CSS aquando da passagem do rato sobre o elemento e na retirada do mesmo. Pretende-se que o elemento fique com cor vermelha e tamanho de letra 56px quando o rato passa por cima do elemento. A função `mouseover`:

```
function mouseOver() {  
  
    document.getElementById("relogioAuto").style.color = "red";  
  
    document.getElementById("relogioAuto").style.fontSize =  
    "56px";  
  
}
```

Para adicionar o efeito que faz com que os estilos desapareçam e voltem ao original quando se retira o rato do elemento, deve-se adicionar a seguinte função `mouseout`:

```
function mouseOut() {  
    document.getElementById("relogioAuto").style.color = "black";  
    document.getElementById("relogioAuto").style.fontSize = "24px";  
}
```

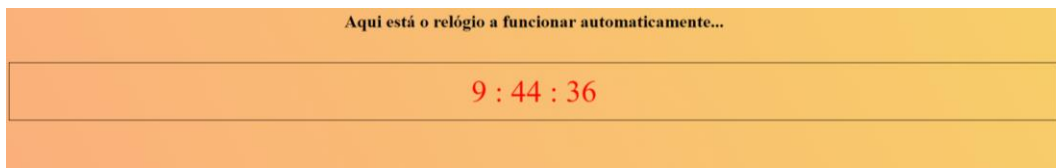
Para que a visualização da página fique mais atrativa devem incluir-se algumas propriedades CSS:

```
<style>  
    body {  
        background-color: #FBAB7E;  
        background-image: linear-gradient(62deg, #FBAB7E 0%, #F7CE68 100%);  
    }  
  
    h1 {  
        text-align: center;  
    }  
  
    p {  
        font-size: 24px;  
        text-align: center;  
        border: 1px solid black;  
        padding: 20px;  
    }  
</style>
```

Consegue-se, assim, a visualização final pretendida.



Visualização inicial da página web.



Visualização da página web com efeito mouseover.

O código final será:

```
<html>

<head>
  <title>Relógio com Javascript</title>
  <meta charset="utf-8">

  <script type="text/javascript">
    var loopRelogio = setInterval(relogio, 1000);
```



```

function relógio() {
    atual = new Date();
    hora = atual.getHours();
    minuto = atual.getMinutes();
    segundo = atual.getSeconds();

    imprimirHora = hora + " : " + minuto + " : " + segundo;
    document.getElementById("relogioAuto").innerHTML = impr
    imirHora;
}

function mouseOver() {
    document.getElementById("relogioAuto").style.color = "r
    ed";
    document.getElementById("relogioAuto").style.fontSize =
    "56px";
}

function mouseOut() {
    document.getElementById("relogioAuto").style.color = "b
    lack";
    document.getElementById("relogioAuto").style.fontSize =
    "24px";
}
</script>

<style>
    body {

```

```
        background-color: #FBAB7E;

        background-image: linear-
gradient(62deg, #FBAB7E 0%, #F7CE68 100%);

    }

    h1 {

        text-align: center;

    }

    p {

        font-size: 24px;

        text-align: center;

        border: 1px solid black;

        padding: 20px;

    }

</style>

</head>

<body>

    <h1>Aqui está o relógio a funcionar automaticamente... </h1>

    <p id="relogioAuto" onmouseover="mouseOver()" onmouseout="mouse
Out()"></p>
```

```
</body>
```

```
</html>
```

É importante referir que o relógio demorará um segundo a aparecer após a página ter sido carregada, pois foi o tempo definido para o JavaScript atualizar a hora, e visto que a hora irá aparecer num campo originalmente vazio.

