SENAI SÃO PAULO

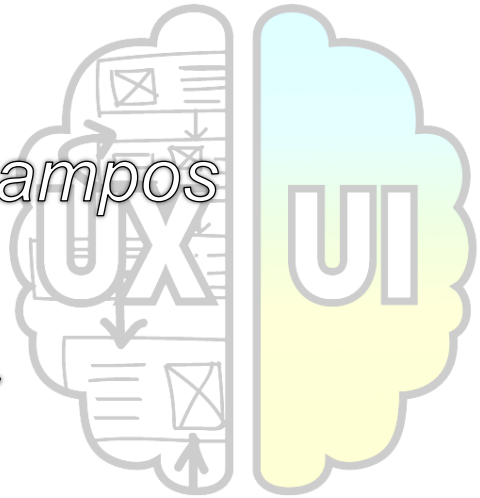
PROGRAMAÇÃO FRONT-END

Professor:

Marco Antonio da Silveira Campos

Cel: (19) 99768-1683

marco.campos@sp.senai.br



SENAI SÃO PAULO

Capítulo 1

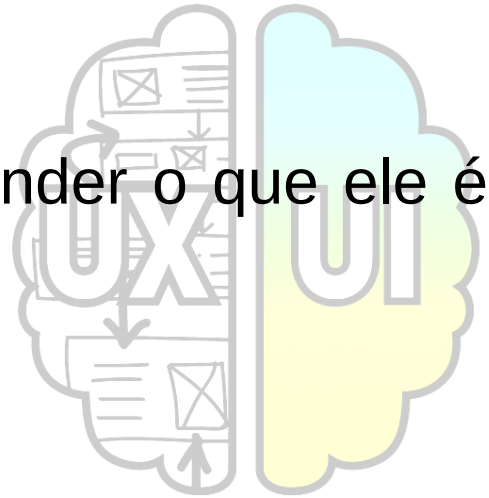
JavaScript
Conceitos Básicos

Prof. Marco Campos
SENAI – Alvares Romi

O QUE É JavaScript?

JavaScript é uma linguagem de programação que permite a você implementar itens complexos em páginas web, mostrando conteúdo que se atualiza em um intervalo de tempo, mapas interativos ou gráficos 2D/3D animados, etc. Se você viu tudo isso, pode apostar que o JavaScript provavelmente está envolvido.

Vamos criar um pequeno exemplo para entender o que ele é capaz de fazer.



Crie o exemplo, execute e em seguida clique sobre o botão para ver o que acontece:

```html
<!DOCTYPE html>
<html lang="pt-BR">
    <head>
        <meta charset="UTF-8">
        <title>Exemplo de JavaScript</title>
        <style>
            p {
              font-family: 'helvetica neue', helvetica, sans-serif;
              letter-spacing: 1px;
              text-transform: uppercase;
              text-align: center;
              border: 2px solid rgba(0,0,200,0.6);
              background: rgba(0,0,200,0.3);
              color: rgba(0,0,200,0.6);
              box-shadow: 1px 1px 2px rgba(0,0,200,0.4);
              border-radius: 10px;
              padding: 3px 10px;
              display: inline-block;
              cursor:pointer;
            }
        </style>
    </head>
    <body>
        <p>Jogador 1: Chris</p>
        <script>
            var para = document.querySelector('p');
            para.addEventListener('click', atualizarNome);
            function atualizarNome() {
              var nome = prompt('Insira um novo nome');
              para.textContent = 'Jogador 1: ' + nome;
            }
        </script>
    </body>
</html>
```

O que é ainda mais empolgante é a funcionalidade construída no topo do núcleo da linguagem JavaScript. As APIs (Application Programming Interfaces - Interface de Programação de Aplicativos) provêem a você superpoderes extras para usar no seu código JavaScript. Elas geralmente se dividem em duas categorias.

API de Navegadores e API de terceiros.

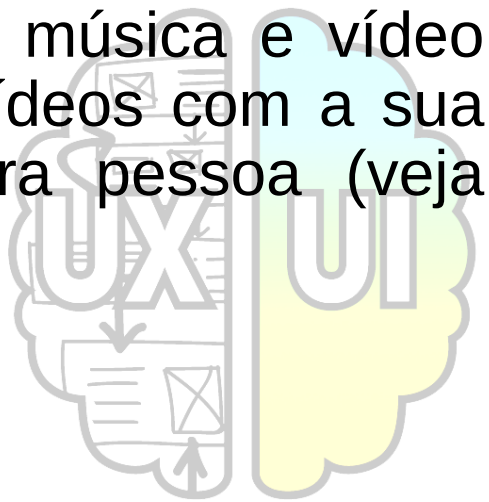
APIs de navegadores já vem implementadas no navegador, e são capazes de expor dados do ambiente do computador, ou fazer coisas complexas e úteis. Por exemplo:

- A API DOM (Document Object Model) permite a você manipular HTML e CSS, criando, removendo e mudando HTML, aplicando dinamicamente novos estilos para a sua página, etc. Toda vez que você vê uma janela popup aparecer em uma página, ou vê algum novo conteúdo sendo exibido (como nós vimos acima na nossa simples demonstração), isso é o DOM em ação.

- A API de Geolocalização recupera informações geográficas. É assim que o Google Maps consegue entontrar sua localização e colocar em um mapa.

- As APIs Canvas e WebGL permite a você criar gráficos 2D e 3D animados. Há pessoas fazendo algumas coisas fantásticas usando essas tecnologias web — veja Chrome Experiments e webglsamples.

- APIs de áudio e vídeo como HTMLMediaElement e WebRTC permitem a você fazer coisas realmente interessantes com multimídia, tanto tocar música e vídeo em uma página da web, como capturar vídeos com a sua câmera e exibir no computador de outra pessoa (veja Snapshot demo para ter uma ideia).



APIs de terceiros não estão implementados no navegador automaticamente, e você geralmente tem que pegar seu código e informações em algum lugar da Web. Por exemplo:

- A API do Twitter permite a você fazer coisas como exibir seus últimos tweets no seu website.
- A API do Google Maps permite a você inserir mapas customizados no seu site e outras diversas funcionalidades.

Como as coisas acontecem?

O JavaScript é executado pelo motor de renderização do navegador, depois que o HTML e CSS forem traduzidos e colocados juntos em uma página web. Isso assegura que a estrutura e estilo da página já estão no lugar na hora em que o JavaScript for executado.

Isso é uma coisa boa, considerando que o JavaScript é comumente usado para modificar dinamicamente HTML e CSS para atualizar a interface do usuário, via API DOM (como mencionado acima). Se o JavaScript fosse carregado e tentasse executar antes que o HTML ou CSS estivesse lá para ser modificado, erros poderiam ocorrer.

Como as coisas acontecem?

Quando o navegador encontra um bloco de código JavaScript, ele geralmente executa na ordem, de cima para baixo. Isso significa que você precisa ter cuidado com a ordem na qual você coloca as coisas.

JavaScript é uma linguagem interpretada — o código é executado de cima para baixo e o resultado da execução do código é imediatamente retornado.

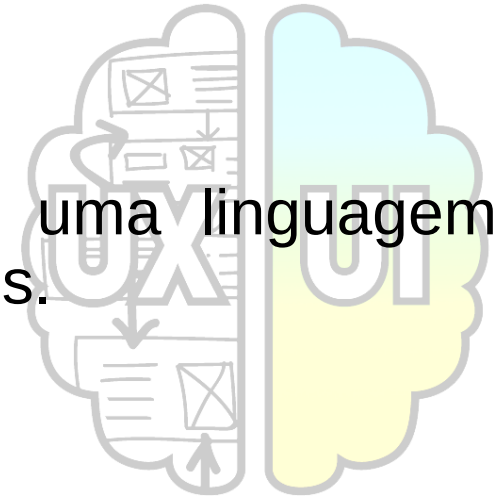
Você pode também ouvir os termos lado do servidor (server-side) e lado do cliente (client-side), especialmente no contexto de desenvolvimento web. Códigos do lado do cliente são executados no computador do usuário.

Como as coisas acontecem?

Códigos do lado do servidor, por outro lado, são executados no servidor e o resultado da execução é baixado e exibido no navegador. Exemplos de linguagens do lado do servidor populares incluem PHP, Python, Ruby, e ASP.NET.

E o JavaScript?

JavaScript também pode ser usada como uma linguagem server-side, por exemplo, no ambiente Node.js.



Como as coisas acontecem?

O JavaScript (JS) é inserido na sua página de uma maneira similar ao CSS. Enquanto o CSS usa o elemento <link> para aplicar folhas de estilo externas e o elemento <style> para aplicar folhas de estilo internas, o JavaScript só precisa da tag <script>.

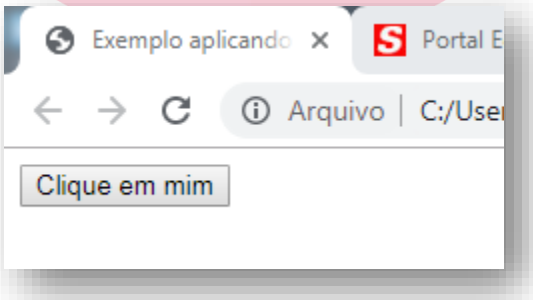
JavaScript é uma linguagem case sensitive (isso significa que a linguagem vê diferença entre letras maiúsculas e minúsculas) e muito confusa, então você precisa digitar a sintaxe exata, senão não vai funcionar e nada será exibido.

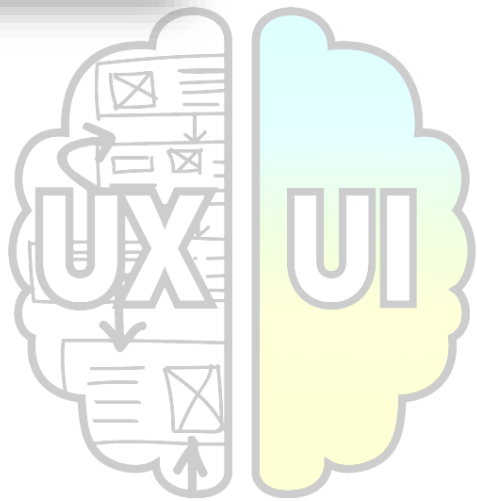
Vamos montar mais um exemplo:

Crie o html abaixo:

```html
<!DOCTYPE html>
<html lang="pt-BR">
    <head>
      <meta charset="utf-8">
      <title>Exemplo aplicando JavaScript</title>
    </head>
    <body>
      <button>Clique em mim</button>
    </body>
</html>
```

Resultado:





Acrescente o JavaScript e veja o resultado:

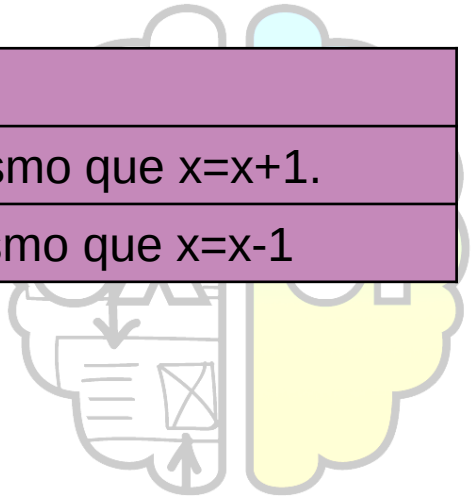
```html
<!DOCTYPE html>
<html lang="pt-BR">
    <head>
      <meta charset="utf-8">
      <title>Exemplo aplicando JavaScript</title>
    </head>
    <body>
        <button>Clique em mim</button>
        <script>
            function criarParagrafo() {
                var para = document.createElement('p');
                para.textContent = 'Você clicou no botão!';
                document.body.appendChild(para);
            }
            var botoes = document.querySelectorAll('button');
            for(var i = 0; i < botoes.length ; i++) {
                botoes[i].addEventListener('click', criarParagrafo);
            }
        </script>
    </body>
</html>
```

Fundamentos!

Agora que você já sabe como utilizar um JavaScript, vamos partir dos fundamentos:

OPERADORES PARA MUDANÇA DE CONTEÚDO:

=	Atribui valor a uma variável.
++	Incrementa valor de uma variável, x++ é o mesmo que x=x+1.
--	Decrementa valor de uma variável, x-- é o mesmo que x=x-1



Fundamentos!

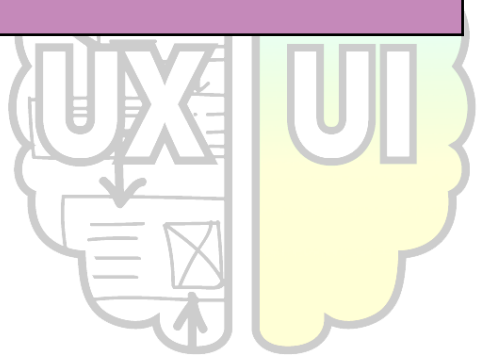
OPERADORES DE COMPARAÇÃO:

==	Igual
!=	Diferente
===	Estritamente igual (verifica conteúdo e tipo da variável).
!==	Estritamente diferente (verifica conteúdo e tipo da variável);
<	Menor que
<=	Menor ou Igual a
>	Maior que
>=	Maior ou igual a

Fundamentos!

OPERADORES ARITMÉTICOS:

%	Módulo – Resto
+	Soma
-	Subtração
*	Multiplicação
/	Divisão



Fundamentos!

OPERADORES LÓGICOS:

&&	E
||	OU
!	Não

OPERADORES ESPECIAIS:

?	Efetua operador condicionada, exemplo x = (a > 1) ? 3 : 4; ou seja se o valor da variável a for maior que 1, será atribuído a x o valor 3 caso contrario 4.
,	A vírgula efetua operador da esquerda para a direita sendo que o último elemento é retornado. Ex.: z=(x=1, y=2); faz com que x passe a valer 1, e y e z passem a valer 2.

Fundamentos!

FUNÇÕES DE DATA E HORA:

http://jlneto.com.br/jlneto/2010/09/javascript-funcoes-de-data-e-hora/

- **classe Date():** retorna a data e hora

```
<html>
    <head>
        <script language="JavaScript">
            function DataHora(){
                var data = new Date();
                document.getElementById("tempo").innerHTML = data;
                setTimeout("DataHora()",1000)
            }
        </script>
    </head>
    <body onload="DataHora()">
        <p id="tempo">Hora: </p>
    </body>
</html>
```

– **getDay():** Retorna um número inteiro do dia da semana. Domingo 0, segunda 1, terça 2, etc.

```html
<html>
    <head>
        <script language="JavaScript">
            function DataHora(){
                var data = new Date();
                dia_semana = data.getDay();
                /*  0 - Domingo
                    1 - Segunda
                    2 - Terça
                    3 - Quarta
                    4 - Quinta
                    5 - Sexta
                    6 - Sábado
                */
                document.getElementById("tempo").innerHTML = dia_semana;
                setTimeout("DataHora()",1000)
            }
        </script>
    </head>
    <body onload="DataHora()">
        <p id="tempo"></p>
    </body>
</html>
```

– **getHours():** Retorna a hora do objeto Date, um número inteiro entre 0 e 23.

```html
<html>
    <head>
        <script language="JavaScript">
            function DataHora(){
                var data = new Date();
                hora = data.getHours();
                document.getElementById("tempo").innerHTML = hora;
                setTimeout("DataHora()",1000)
            }
        </script>
    </head>
    <body onload="DataHora()">
        <p id="tempo"></p>
    </body>
</html>
```

– **getMilliseconds():** Retorna os milissegundos do objeto Date, um inteiro entre 0 e 999.

```
<html>
    <head>
        <script language="JavaScript">
            function DataHora(){
                var data = new Date();
                ms = data.getMilliseconds();
                document.getElementById("tempo").innerHTML = ms;
                setTimeout("DataHora()",1000)
            }
        </script>
    </head>
    <body onload="DataHora()">
        <p id="tempo"></p>
    </body>
</html>
```

– **getMinutes():** Retorna os minutos do objeto Date, um inteiro entre 0 e 59.

```
<html>
    <head>
        <script language="JavaScript">
            function DataHora(){
                var data = new Date();
                min = data.getMinutes();
                document.getElementById("tempo").innerHTML = min;
                setTimeout("DataHora()",1000)
            }
        </script>
    </head>
    <body onload="DataHora()">
        <p id="tempo"></p>
    </body>
</html>
```

– **getMonth():** Retorna o mês do objeto Date, um inteiro entre 0 e 11 (0 janeiro, 1 fevereiro, etc)..

```
<html>
    <head>
        <script language="JavaScript">
            function DataHora(){
                var data = new Date();
                mes = data.getMonth();
                document.getElementById("tempo").innerHTML = mes;
                setTimeout("DataHora()",1000)
            }
        </script>
    </head>
    <body onload="DataHora()">
        <p id="tempo"></p>
    </body>
</html>
```

- **getDate():** Retorna o dia do mês do objeto Date.
- **getYear():** Retorna o ano do objeto Date.

```html
<html>
    <head>
        <script language="JavaScript">
            function DataHora(){
                var d = new Date();
                var dia = d.getDate();
                var mes = d.getMonth();
                var ano = d.getYear();
//                getYear retorna o ano menos 1900
                var anocorreto = ano+1900;
                document.getElementById("tempo").innerHTML = dia+"/"+mes+"/"+anocorreto;
            }
        </script>
    </head>
    <body onload="DataHora()">
        <p id="tempo"></p>
    </body>
</html>
```

Atribuição de data a variável:

var novaData = new Date(MES+”-”+DIA+”-”+ANO);

Ou

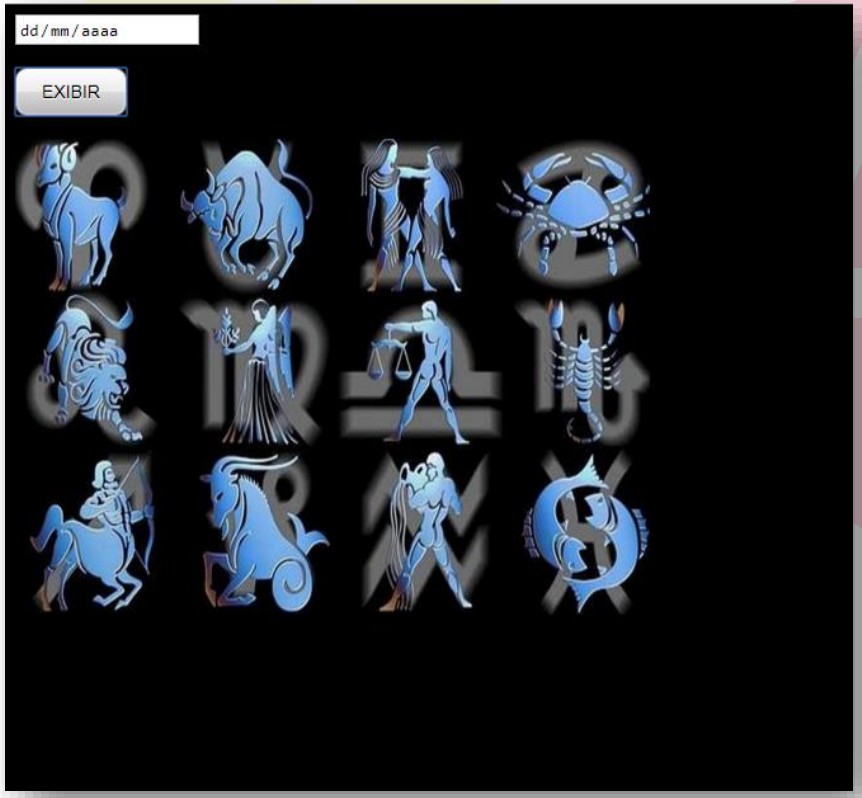
– **Date.parse(string_de_data):** Retorna o número de milissegundos de uma seqüência de caracteres (string) de data, desde 01 de janeiro de 1970 00:00:00 h (hora local).

```
1 data = new Date();
2 data.setTime(Date.parse("Apr 15, 2004"));
```

– **toLocaleString()**: Retorna uma seqüência de caracteres (string) de data, com formato definido pelas configurações do sistema operacional.

```
1  <html>
2      <head>
3          <script language="JavaScript">
4              function DataHora(){
5                  var d = new Date();
6                  var data = d.toLocaleString();
7                  document.getElementById("tempo").innerHTML = data;
8              }
9          </script>
10     </head>
11     <body onload="DataHora()">
12         <p id="tempo"></p>
13     </body>
14 </html>
15
```

– **EXEMPLO:** Criar um programa onde o usuário digita uma data de nascimento no formato dd/mm/aaaa e ele informa o signo do usuário mostrando uma imagem correspondente.



```html
<!DOCTYPE html>
<html lang="pt-br">
    <head>
        <meta charset="UTF-8">
        <title>Signos do Zodíaco</title>
        <link rel="stylesheet" type="text/css" href="css/estilo.css"/>
        <script type="text/javascript" src="jscript/jscript.js"></script>
    </head>
    <body>
        <form>
            <input type="date" id="data"/><br><br>
        </form>
        <button onclick="exibir()" id="botao">EXIBIR</button>
        <br><br>
        <img src="images/signos.jpg" id="signoimg">
        <p id="resposta"></p>
    </body>
</html>

```

```
var sDatas = [["01-01","01-19","CAPRICÓRNIO","capricornio.jpg"],
              ["01-21","02-18","AQUÁRIO",    "aquario.jpg"],
              ["02-19","03-20","PEIXES",     "peixes.jpg"],
              ["03-21","04-19","ARIES",      "aries.jpg"],
              ["04-21","05-20","TOURO",      "touro.jpg"],
              ["05-21","06-20","GÊMEOS",     "gemeos.jpg"],
              ["06-21","07-22","CÂNCER",     "cancer.jpg"],
              ["07-23","08-22","LEÃO",       "leao.jpg"],
              ["08-23","09-22","VIRGEM",     "virgem.jpg"],
              ["09-23","10-22","LIBRA",      "libra.jpg"],
              ["10-23","11-21","ESCORPIÃO",  "escorpiao.jpg"],
              ["11-22","12-21","SAGITÁRIO",  "sagitario.jpg"],
              ["12-22","12-31","CAPRICÓRNIO","capricornio.jpg"]];
var lLimpar=false;
//=============================================================================
function exibir(){
    if(lLimpar=!lLimpar){
        //    Pega a data digitada
        var dataDig = document.getElementById("data").value;
    //    Converte para data
        var data = new Date(dataDig);
    //    Separa o dia, o mes e o ano
        var dia = data.getDate()+1;
        var mes = data.getMonth()+1;
        var ano = data.getYear();
        var dataVetor;
    //    Cria a data para comparação
        var novaData = new Date(mes+"-"+dia+"-"+ano);
```

```
29      //    Percorre o vetor comparando as datas
30          for(var i=0;i<sDatas.length;i++){
31      //        Junta o ano com a data do vetor para comparação completa
32              dataVetor = sDatas[i][0]+"-"+ano;
33              dataSig1 = new Date(dataVetor);
34              dataVetor = sDatas[i][1]+"-"+ano;
35              dataSig2 = new Date(dataVetor);
36      //        Faz a comparação
37              if(novaData>=dataSig1 && novaData<=dataSig2){
38      //            Signo
39                  document.getElementById("resposta").innerHTML=sDatas[i][2];
40      //            Imagem do signo
41                  document.getElementById("signoimg").src="images/"+sDatas[i][3];
42                  document.getElementById("botao").innerHTML = "LIMPAR";
43                  document.getElementById("data").disabled = true;
44              }
45          }
46      }else{
47          document.getElementById("resposta").innerHTML="";
48          document.getElementById("signoimg").src="images/signos.jpg";
49          document.getElementById("botao").innerHTML = "EXIBIR";
50          document.getElementById("data").value = "";
51          document.getElementById("data").focus;
52          document.getElementById("data").disabled = false;
53      }
54  }
```

```css
body{
    background: rgb(0, 0, 0);
}
button{
    font-family: Arial, Helvetica, sans-serif;
    font-size: 14px;
    color: #050505;
    padding: 10px 20px;
    background: linear-gradient(to bottom,#ffffff 0%,#ebebeb 50%,#dbdbdb 50%,#b5b5b5);
    border-radius: 10px;
    border: 1px solid #949494;
    box-shadow:
        0px 1px 3px rgba(000,000,000,0.5),
        inset 0px 0px 2px rgba(255,255,255,1);
    text-shadow:
        0px -1px 0px rgba(000,000,000,0.2),
        0px 1px 0px rgba(255,255,255,1);
}
button:hover{
    background: linear-gradient(to top,#ffffff 0%,#ebebeb 50%,#dbdbdb 50%,#b5b5b5);
}
img{
    width: 500px;
    height: 375px;
}
#resposta{
    font-family: Arial, Helvetica, sans-serif;
    font-size: 25px;
    color: #FFFFFF;
}
```

– **getTime()**: Converte uma data em milissegundos

```
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3      <head>
4          <meta charset="UTF-8">
5          <title>Conversão para milissegundos</title>
6      </head>
7      <body>
8          <p>O relógio interno no JavaScript, inicia a meia noite do dia
           01 de Janeiro de 1970</p>
9          <p>Clique no botão para exibir a data atual, convertida em
           milissegundos desde a data de início do JavaScript</p>
10         <button onclick="converte()">Converte</button>
11         <p id="resposta"></p>
12         <script>
13             function converte(){
14                 var data = new Date();
15                 var mili = data.getTime();
16                 document.getElementById("resposta").innerHTML = mili;
17             }
18         </script>
19     </body>
20 </html>
```

EXERCÍCIO 1

SENAI SÃO PAULO

Criar uma página web, onde o usuário digita a data de nascimento no formato dd/mm/aaaa (<input type="date“>) e a página deverá exibir a idade em anos, meses e dias. Exibirá também o total de dias de existência.

Para realizar este cálculo, converta a data de nascimento em milissegundos. Depois disso, sabemos que:

1s = 1000 milissegundos;

1min = 60s

1h = 60min

1dia = 24h, portanto 1 dia = 1000*60*60*24 milissegundos

Para sabermos a quantidade de dias, pegamos a conversão dos milissegundos e dividimos pela conta acima.

Sabendo a quantidade de dias, basta, dividir por 365 para sabermos os anos, pegar o resto da divisão e dividir por 31 para sabermos os meses e o que restou são os dias.