GUIA DE LABORATÓRIO 1.1
INTERACTIVIDADE C/ JAVASCRIPT

OBJECTIVOS

- Conhecer o ambiente de desenvolvimento
- Tomar contacto com os elementos básicos da linguagem: valores, variáveis e tipos de dados

INSTRUÇÕES

PARTE I – UM EXEMPLO

1. Utilizando um editor de texto, por exemplo, o Notepad++, abra o ficheiro `exemplo.html` distribuído com este laboratório.

> *Este ficheiro consiste no esqueleto de um "jogo" de adivinha designado por "O Número Mágico". O número mágico é um número pré-definido (no nosso caso será o 19) que o utilizador deve tentar adivinhar. Se acertar, o jogo dá os parabéns ao jogador de forma personalizada, isto se ele tiver introduzido o respectivo nome. Se falhar, é notificado disso mesmo e pode voltar a tentar.*
> *A interface com o utilizador já foi desenvolvida nas linguagens HTML e CSS. A nossa missão consiste em programar as decisões e acções do jogo, o que faremos em JavaScript.*

2. Estude o conteúdo deste ficheiro. Depois, abra a folha de estilos externa que o acompanha (ficheiro `exemplo.css`) e estude, também, o conteúdo deste ficheiro.

3. Abra o ficheiro html através do navegador Web e tente "jogar". O que sucede?

4. Existe uma **`div`** no documento que não é exibida. Qual é esta **`div`** e qual a propriedade CSS responsável por ela não ser exibida?

5. Temporariamente experimente substituir esta propriedade por

```
visibility: hidden;
```

O que sucede? Reponha a propriedade original em vez desta.

JavaScript é uma linguagem interpretada que é utilizada, na grande maioria dos casos, como complemento das páginas Web. Tanto pode ser utilizada para processar informação introduzida pelo utilizador, como para codificar e enviar dados para o servidor que aloja as páginas, ou para alterar a interface com o utilizador destas páginas sem que seja necessário descarregar novo código HTML/CSS do servidor, entre outras utilizações. É também utilizada para desenvolver animações gráficas, jogos e, cada vez mais, começa a ser utilizada para desenvolver aplicações fora do browser.
Quando descarrega uma página, o navegador organiza-a numa estrutura hierárquica designada por DOM - Document Object Model. Esta consiste numa "árvore" de objectos estando na raiz o objecto que representa a própria página, designado por ***`document`*** *e tendo com "descendentes os restantes elementos html (`body`, `div`, `p`, etc.). O JavaScript pode interagir com esta a estrutura da página e, se necessário, alterá-la.*

Da Wikipedia:
"JavaScript foi originalmente desenvolvido por Brendan Eich da Netscape sob o nome de Mocha, posteriormente teve seu nome mudado para LiveScript e por fim JavaScript. LiveScript foi o nome oficial da linguagem quando foi lançada pela primeira vez na versão beta do navegador Netscape 2.0 em setembro de 1995, mas teve seu nome mudado em um anúncio conjunto com a Sun Microsystems em dezembro de 1995 quando foi implementado no navegador Netscape versão 2.0B3.
A mudança de nome de LiveScript para JavaScript coincidiu com a época em que a Netscape adicionou suporte à tecnologia Java em seu navegador (Applets). A escolha final do nome causou confusão dando a impressão de que a linguagem foi baseada em java, sendo que tal escolha foi caracterizada por muitos como uma estratégia de marketing da Netscape para aproveitar a popularidade do recém-lançado Java.".
Para mais informação sobre a história e aspectos gerais de JavaScript siga a seguinte ligação
-> http://pt.wikipedia.org/JavaScript

6. O botão `Adivinhar!` deve mostrar o resultado. Vamos começar por "dar acção" ao botão `adivinhar`. Acrescente o seguinte atributo ao **`input`** cujo o **`id`** é **"**`adivinhar`**"**

```
onclick='alert("Olá, Mundo! Saudações aqui do JavaScript.")'
```

Teste. O que sucede?

O HTML permite associar um conjunto de acções a um acontecimento ou evento. Estes eventos, pré-definidos para cada elemento, correspondem a atributos do elemento. Por seu turno, as acções correspondem a instruções JavaScript. Neste caso, utilizamos o evento/atributo ***`onclick`*** *para exibir uma mensagem quando o utilizador pressionar (ie, "clickar") no botão `adivinhar` . A instrução* ***`alert("...")`*** *é responsável por fazer aparecer uma pequena caixa de diálogo com a mensagem.*

7. Substitua todo o atributo/evento **`onclick`** pelo seguinte e teste no navegador:

```
onclick='let soma = 0; for (let x = 1; x <= 10; x++) soma+=x; alert(soma);'
```

Qual o resultado?

8. Em alguma parte do documento (por exemplo, dentro `head`), acrescente o seguinte elemento.

```
<script type="text/javascript">

</script>
```

*Não somos obrigados a escrever todo o código entre as plicas/aspas dos atributos correspondentes a eventos (como o **`onclick`**). Á semelhança do que fazemos com CSS, podemos agrupar instruções JavaScript nalguma parte do documento, desde que fiquem "dentro" do elemento `script`, que serve justamente para isso.*
Em XHTML poderá ter que escrever as etiquetas do elemento script da seguinte forma:

```
<script type="text/javascript">//<![CDATA[
//]]></script>
```

*Isto evita que o XHTML tente interpretar o conteúdo do elemento **`script`**.*
Podemos também colocar as instruções num ficheiro com extensão `.js` e referi-lo através do atributo src. Por exemplo:

```
<script type="text/javascript"
src="ficheiro.js"></script>
```

O elemento script pode ser colocado em qualquer parte do documento.

9. Dentro das etiquetas de abertura e de fecho do elemento `script` coloque o seguinte código.

```
function somaAte (limite) {
   let soma = 0;
   for (let x = 1; x <= limite; x++) {
        soma+=x;
   }
   return soma;
}
```

*Não somos obrigados a escrever todo o código entre as plicas/aspas do atributo **`onclick`**. Podemos "empacotar" um conjunto de acções nalguma parte do documento, dar um nome a esse conjunto e depois executar essas acções através do nome. Esse "pacote" de acções é designado por função. Em geral, uma função é um bloco de instruções com um nome, zero ou mais parâmetros e que pode produzir nova informação a partir desses parâmetros. Neste caso o nome da função é `somaAte`, possui apenas um parâmetro designado por `limite` e produz o resultado da soma dos números de um até `limite`.*

10. Agora vamos modificar o atributo/evento **`onclick`** para:

```
onclick='alert(somaAte(10));'
```

Teste no navegador. A título de curiosidade, experimente colocar a expressão **`soma(10)`** entre aspas. O que sucede? Remova as plicas, altere o valor 10 para 50 e volte a testar.

11. Agora vamos activar o "painel" `resultado`. A ideia consiste em fazer aparecer esta **`div`** com o resultado da adivinha (indicando se acertou ou não). Para tal, vamos começar por acrescentar o seguinte código ao conteúdo do elemento **`script`** (antes ou depois da função `somaAte`):

```
function mostra (elemID, texto) {
  let item = document.getElementById(elemID);
  if (item) {
    if (texto !== undefined) {
       item.innerHTML = texto;
```

```
      }
      item.style.visibility="visible";
      item.style.display="block";
    }
  }
```

*Esta função garante que um elemento fica visível modificando as respectivas propriedades CSS **visibility** e **display**. Para tal recebe dois parâmetros: o valor do **id** do elemento a mostrar (`elemID`) e uma mensagem (`texto`). Depois utiliza o valor do parâmetro `elemID` para localizar o elemento na hierarquia do documento, utilizando para tal uma função do elemento **html** (aqui representando pelo objecto **document**). Uma vez localizado o elemento, podemos aceder às propriedades CSS do elemento através de uma outra propriedade que corresponde ao atributo **style**. Vamos utilizá-la para exibir a **div** `resultado`.*

12. Modifique o valor do atributo **onclick** do botão `adivinhar` para:

```
onclick='mostra("resultado", "Olá!");'
```

Teste no navegador.

13. Já que acrescentámos uma função para mostrar um elemento, vamos fazer também a função inversa. Adicione a função seguinte ao elemento **script**:

```
function esconde (elemID) {
   let item = document.getElementById(elemID);
   if (item) {
     item.style.visibility="hidden";
     item.style.display="none";
   }
}
```

14. Para testar esta função vamos utilizar o evento/atributo **onclick** do próprio painel `resultado`. Se o utilizador *clickar* dentro deste painel ele fecha-se. A **div** `resultado` deverá ficar assim:

```
<div id="resultado" onclick='esconde("resultado")'>
</div>
```

15. Já agora, vamos acrescentar mais duas funções que nos poderão vir a ser úteis mais à frente. A primeira desactiva um elemento **input**. Ou seja, o elemento não aceita qualquer tipo de acção e é exibido com uma cor ligeiramente diferente. A segunda faz o inverso:

```
function activa (elemID) {
  let item = document.getElementById(elemID);
  if (item) {
     item.disabled = false;
  }
}

function desactiva (elemID) {
```

*À semelhança das anteriores, estas funções localizam o elemento a desactivar/activar no documento e depois manipulam o seu atributo **disabled**.*

```
  let item = document.getElementById(elemID);
  if (item) {
     item.disabled = true;
  }
}
```

Acrescente as funções ao conteúdo do elemento **`script`**. Vamos utilizá-las mais à frente para activar/desactivar o botão `adivinhar`.

16. Agora queremos que a mensagem do painel `resultado` indique se o utilizador acertou ou não. Acrescente a seguinte função ao conteúdo do elemento **`script`**:

```
function verificaResultado () {
   let nome = document.getElementById("nome");
   let numero = document.getElementById("numero");
   if (numero) {
     let texto;
     if (numero.value == numMagico) {
       texto = "Parabéns, " + (nome.value ? nome.value : " caro anónimo");
       texto += "! Acertou";
     }
     else {
       texto = "Falhou! Tente novamente...";
     }
     mostra("resultado", texto);
     desactiva("adivinhar");
   }
}
```

*A função `verificaResultado` localiza no documento os elementos (os **`inputs`**) com **`id`** `numero` e `nome`. Se o conteúdo do **`input`** `numero` for igual ao número mágico `19`, então a função constrói uma mensagem personalizada. Se o utilizador não tiver inserido nada no **`input`** `nome`, é tratado por `"caro anónimo"`, caso contrário o conteúdo do **`input`** é utilizado nesta mensagem personalizada.*
Se o valor de `numero` for diferente de 19, então a função constrói uma mensagem a indicar que o utilizador falhou. Uma vez construída a mensagem, esta é passada para a função `mostra` que exibe o painel `resultado` com a mensagem lá colocada. De seguida, o botão `adivinhar` é desactivado através da função `desactiva`.

17. Agora voltamos a modificar o evento **`onclick`** do **`input`** `adivinhar` para:

```
onclick='verificaResultado();'
```

Teste no navegador. Repare que após a primeira tentativa o botão `adivinhar` fica desactivado.

18. Vamos acrescentar uma função para permitir ao utilizador voltar a tentar.

```
function voltaATentar () {
  let numero = document.getElementById("numero");
```

```
    if (numero) {
       numero.value = "";
    }
    activa("adivinhar");
    esconde("resultado");
 }
```

*Para voltar a tentar o utilizador "clicka" dentro do painel `resultado` e esta função "limpa" o **`input`** `numero`, activa o botão `adivinhar` e volta a esconder o painel `resultado`.*

19. Associe esta função ao evento **`onclick`** da **`div`** `resultado`.

```
onclick='voltaATentar();'
```

20. Teste com vários números e nomes.

21. Para terminarmos o exemplo, crie um ficheiro com o nome `exemplo.js` . Este ficheiro deverá ficar na mesma pasta que os ficheiros `exemplo.html` e `exemplo.css` . Mova para esse ficheiro todo o código JavaScript. O elemento script deverá ficar vazio.

22. Finalmente, acrescente o atributo **`src`** à etiqueta de abertura do elemento **`script`** e dê-lhe o valor `exemplo.js`. Ou seja, o código do elemento **`script`** será o seguinte:

```
<script type="text/javascript" src="exemplo.js"></script>
```