INTRODUÇÃO

*Algoritmia - JavaScript*

O JavaScript é uma linguagem vocacionada para a web. Usada em biliões de páginas web, acrescenta funcionalidades, valida *forms*, comunica com servidores e muito mais.

Considere os elementos numerados da imagem. Identifica cada elemento numa comunicação cliente/servidor.

5

4



3

2

1

# Como apresentar informação num Output?

O *javascript* pode enviar os dados para um *output* de diferentes formas:

* uma janela alert: **window.alert();**
* um documento HTML: **document.write();**
* um elemento HTML: **innerHTML();**
* a consola do browser: **console.log();**

# Manipulação de elementos HTML

As instruções *JavaScript* são executadas enquanto se processa o carregamento da página. O acesso aos elementos HTML pode realizar-se recorrendo aos métodos:

* ***document.getElementById(id)*** , onde o ***id*** representa o identificador do elemento HTML.
* ***document.getElementsByClassName("classname")*** , onde ***class*** representa a classe elemento HTML. O método ***getElementsByClassName()*** devolve todos os elementos do documento que tenham a classe “classname”, como um objeto NodeList. Um objeto NodeList representa um conjunto de nós (nodes). Os nós são acessíveis por um índice numérico, que inicia com o valor 0. Pode utilizar a propriedade length do objeto NodeList que devolve o número de elementos que possuem a classe definida.

# Teste e controlo de erros em algoritmia

O **teste e o controlo de erros** num algoritmo permitem verificar se este resolve o problema pretendido. Para isso, utiliza-se a técnica do *tracing* (traçagem), que permite acompanhar passo a passo a execução de um algoritmo, eliminando, desta forma, a possibilidade de ocorrência de erros.

O *tracing* consiste, assim, em testar um algoritmo com valores de entrada, observando o comportamento interno ao longo dos vários passos que compõem o algoritmo.

Vejamos um exemplo que ilustra o teste ao algoritmo:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Ler a;  Ler b;  c ← a + b;  Escrever (c) | | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **a** | **b** | **C** ← **a+b** | **Saída** | | 1º Passo | 4 |  |  |  | | 2º Passo | 4 | 5 |  |  | | 3º Passo | 4 | 5 |  |  | | 4º Passo | 4 | 5 |  |  | |

Crie a traçagem dos seguintes algoritmos, atribuindo à variável **a** o valor de entrada **2**, na alínea **a)** e às variáveis **a** e **b** os valores de entrada **true** e **false**, respetivamente, na alínea **b)**.

1. Faz o *tracing* dos seguintes algoritmos.
2. Algoritmo A

Ler (a);

B ←2 \* a;

C ← a + b;

A ← 2 \* c + 2 \* b;

B ← a – 10;

Escrever (a, b, c)

1. Algoritmo B

Ler (a);

Ler (b);

C ← a;

A ← b;

B ← c;

Escrever (a, b)

# Estruturas de controlo

As estruturas de controlo avaliam o desenrolar das instruções de um programa. Estas subdividem-se em sequencial, decisão ou seleção e repetição ou ciclos.

## Estrutura sequencial.

As instruções de um programa são executadas numa determinada sequência sem que esta possa sofrer alterações através dos dados de entrada.

Considera o algoritmo em pseudocódigo

Algoritmo Total\_produto;

Início

Escrever (“Digite a designação do produto”);

Ler (designa);

Escrever (“Digite a quantidade do produto”);

Ler (quantidade);

Escrever (“Digite o preço por unidade do produto”);

Ler (preco);

Total ← quantidade \* preco;

Escrever (“O valor total do produto ”, designa, “ é “, total);

Fim

1. Cria um fluxograma que permita implementar o algoritmo em pseudocódigo.
2. Apresenta uma traçagem (tracing) do algoritmo, considerando os seguintes dados de entrada:

**designa** = “Lápis”

**quantidade** = 10

**preco** = 0.50

1. Cria um programa em javascript que implemente o algoritmo.

# Informação útil

## Como determinar o tipo de uma variável?

O comando ***typeof*** permite avaliar o tipo de dados de uma variável.

**document.write("resultado : <b>", typeof resultado, "</b><br />");**

## Como converter uma string num número inteiro?

var num1 = **parseInt**("1234blue");

## Como converter uma string num número real?

var num4 = **parseFloat**(“22.5”);

## Como criar uma constante?

**const altura** = 1.74;

## Como fixar o número de casas decimais?

num = **Math.pow(1.73, 2.5)** – a variável num guarda o resultado do valor 1.732.5

num1 = **num.toFixed(2)** – a variável num1 guarda o valor num com 2 casas decimais.

## Como converter um número numa string?

var num = 10, numero;

numero = **num.toString()**;

## Como procurar uma string dentro de outra?

var msg = "Escola Secundária Emídio Navarro - Viseu";

var substring = "Emídio";

**var posicao = msg.search(substring);**

document.write(posicao);

Se a string a pesquisar não existir então o valor devolvido será -1. O primeiro elemento da string tem índice zero (0).

## Como atualizar uma string?

var msg = "Escola Secundária Emídio Navarro - Viseu";

var pesquisa = "Viseu";

var muda = "Jardim";

**var resultado = msg.replace(pesquisa, muda);**

document.write("<b>Mensagem final:</b> " + resultado);

## Quais são os operadores lógicos?

|  |  |
| --- | --- |
| **operador** | **descrição** |
| && | E |
| || | OU |
| ! | NEGAÇÃO |

# Como fazer *debugging?*

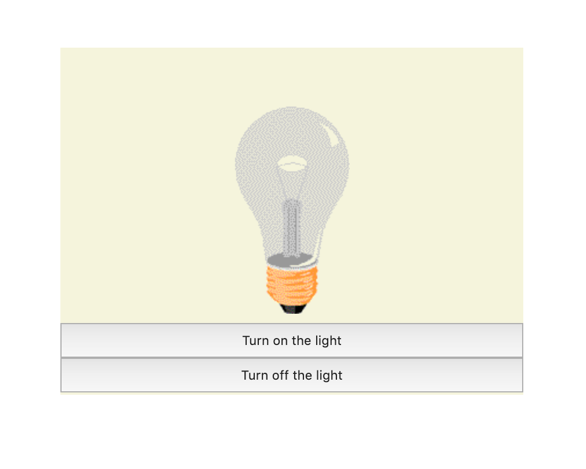
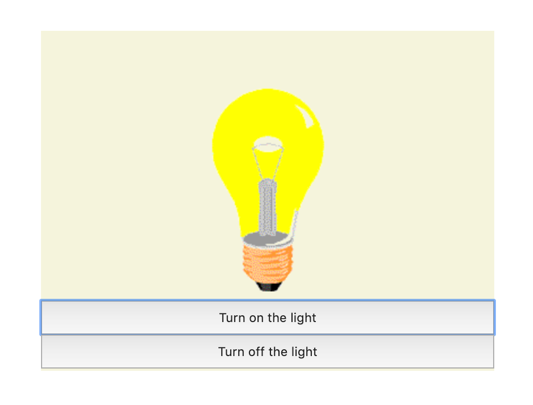
Enquanto se programa é importante analisar os erros cometidos durante o processo de programação. Apresenta-se uma tabela que ilustra o acesso aos erros nos diferentes browsers:

|  |  |
| --- | --- |
| **Browser** | **Como aceder às mensagens de erro** |
| Google Chrome | **Menu > Ferramentas > Consola JavaScript** ou Ctrl - Shift - J |
| Internet Explorer | **Ferramentas > Ferramentas de programação** (F12) |
| Firefox | **Ferramentas > Desenvolvedor web >** escolher ferramenta ou *Ctrl - Shift - J* |
| Opera | **Página > Ferramentas de desenvolvimento > Opera DragonFly** |

# Desafio

Aproveita o material existente no site: <http://www.w3schools.com/howto/default.asp>, adapta os exemplos que aí encontras para melhorar os exercícios solicitados.

1. Constrói um programa que permita acender e desligar uma lâmpada. Considera a interface:



As tags HTML possuem atributos que permitem o disparo de acontecimento, tecnicamente chamados de eventos. É o caso do botão <button ***onclick****="nome"*>Turn on the light </button>.

Podes explorar estes eventos em: <https://www.w3schools.com/js/js_htmldom_events.asp>

Mais tarde vamos explorar melhor este tema.

Por outro lado, a seleção de um elemento pelo identificador, **document.getElementById('identificador')** possui vários atributos, no nosso caso interessa o atributo src que permite definir uma imagem ou outro documento (**document.getElementById('identificador').src=’imagem.jpg’**).

Como deve funcionar a aplicação: ao clicar num botão “Turn on the light” a lâmpada acende, ao clicar no botão “Turn off the light” ela desliga.

Pratica a aplicação destes atributos na seguinte página: <https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dom_html6>