Laboratório 01 – Lógica de Programação

1. Introdução

Esta aula de laboratório é baseada em [Silveira & Almeida, 2013][1]. A ideia é fazer com que os alunos revejam os conceitos básicos de algoritmos em um ambiente de programação real e que está disponível em qualquer computador com acesso à Internet. As instruções abaixo devem ser passadas para os alunos para que vejam os programas em execução.

2. Executando na máquina local

2.1. Uma página HTML:

- Abra um editor de texto qualquer;
- Digite o seguinte código:

```
Meu primeiro teste!
<h3>Seria isso um programa?</h3>
```

- Salve o arquivo com o nome `pagina.html`;
- Localize o arquivo recém-salvo com um gerenciador de arquivos e abra-o com um duplo-clique;
- O que aconteceu? Qual é a resposta para a pergunta incluída na página?

2.2. Um programa JavaScript:

- Altere seu programa, adicionando as seguintes linhas de código:

```
<script>
alert("Olá, mundo!");
</script>
```

- Volte ao navegador e atualize a página. O que aconteceu?

3. Utilizando um ambiente JavaScript na Web

- Ainda no navegador, visite o site www.tinkerbin.com;
- Na parte **JavaScript**, digite:

```
document.write("Olá, mundo!");
```

- Clique no botão **Run** ou pressione **Ctrl+Enter**. O que aconteceu? Qual a diferença entre o `alert()` e o `document.write()`?

4. Outros programas – Ilustrando conceitos de algoritmos

- Os alunos podem escolher dentre os ambientes acima como eles preferem prosseguir;
- Modifique o código JavaScript para mostrar diferentes conceitos de algoritmos;

---

[1] Silveira, P; Almeida, A. Lógica de Programação. Casa do Código, 2013.

- Antes de escrever o código no quadro para os alunos experimentarem, substitua termos em *ITÁLICO_E_CAIXA_ALTA* pelos valores que eles representam (dados sobre o professor, aluno ou contexto atual);
- Os prefixos **[NEW]** e **[ADD]** são usados para indicar, respectivamente, se o código anterior deve ser substituído pelo novo ou se o novo código deve ser adicionado ao anterior (atenção: a adição pode ser acima ou abaixo do antigo código ou então modificar partes do mesmo);
- O prefixo **[NOTA]** é usado em códigos que contém observações importantes, mas que devem ser feitos no canto do quadro, sem substituir o código atual.

4.1. Diferença entre *strings* e números:

- **[NEW]** Imprimindo dados literais e dados numéricos:

```
document.write("Minha idade é: ");
document.write(IDADE);
```

- **[NEW]** Diferença entre os tipos, soma de números, concatenação de *strings*:

```
document.write(25 + 25);
document.write(", ");
document.write("25" + "25");
document.write(", ");
document.write("25 + 25");
```

4.2. Operações matemáticas:

- **[NEW]** Uma operação simples de subtração:

```
document.write("Eu nasci em: " + (ANO_ATUAL - MINHA_IDADE) + "<br>");
document.write("Meu colega nasceu em: " + (ANO_ATUAL -
IDADE_DO_MEU_COLEGA) + "<br>");
```

- **[ADD]** Precedência de operadores – o que está errado na expressão? Corrija:

```
document.write("A média das nossas idades é: " + (MINHA_IDADE +
IDADE_DO_MEU_COLEGA / 2));
```

4.3. Organizando os dados em variáveis:

- **[ADD]** Declare variáveis no início e substitua suas ocorrências no código anterior:

```
var ano = ANO_ATUAL;
var minhaIdade = MINHA_IDADE;
var idadeColega = IDADE_DO_MEU_COLEGA;
```

- **[NOTA]** Diferença entre variáveis e *strings*:

```
document.write(ano);
document.write("ano");
```

- **[ADD]** Fazendo passo a passo as operações:

```
var media = minhaIdade = idadeColega;
media = media / 2;
document.write("A média das nossas idades é: " + media);
```

4.4. Utilizando funções:

- **[ADD]** Uma função para pular a linha. Repare a identação (tab) do conteúdo da função:

```
var pulaLinha = function() {
     document.write("<br>");
}

document.write("Eu nasci em: " + (ano - minhaIdade));
pulaLinha();
document.write("Meu colega nasceu em: " + (ano - idadeColega));
pulaLinha();
document.write("A média das nossas idades é: " + ((minhaIdade +
idadeColega) / 2));
```

- **[ADD]** Passando parâmetros para funções:

```
var escreve = function(frase) {
     document.write(frase + "<br>");
}

escreve("Eu nasci em: " + (ano - minhaIdade));
escreve("Meu colega nasceu em: " + (ano - idadeColega));
escreve("A média das nossas idades é: " + ((minhaIdade + idadeColega)
/ 2));
```

- **[ADD]** Função chamando função:

```
var escreve = function(frase) {
     document.write(frase);
     pulaLinha();
}
```

4.5. Comentários

- [ADD] Documentando as funções e o código:

```
// Esta função escreve uma quebra de linha na página.
var pulaLinha = function() {

// Esta função escreve uma frase, seguida de uma quebra de linha.
var mostra = function(frase) {

// Início do algoritmo. Calcula a média entre duas idades.
```

4.6. Mais sobre funções

- **[NEW]** Funções podem ser declaradas de uma outra forma mais direta (e mais comum):

```
function pulaLinha() {
     document.write("<br>");
}

function escreve(frase) {
     document.write(frase);
```

```
    pulaLinha();
}
```

- **[ADD]** Um novo exemplo:

```
var peso = MEU_PESO;
var altura = MINHA_ALTURA;
var imc = peso / (altura * altura);
escreve("Meu IMC é: " + imc);
```

- **[ADD]** Para fazer o IMC do colega sem repetir o código todo, use uma função:

```
function calculaIMC(altura, peso) {
    var imc = peso / (altura * altura);
    escreve("IMC calculado é: " + imc);
}
escreve("Meu IMC é:");
calculaIMC(MINHA_ALTURA, MEU_PESO);
escreve("O do colega é:");
calculaIMC(ALTURA_COLEGA, PESO_COLEGA);
```

- **[ADD]** Melhor que o escreve() dentro de calculaIMC() é usar retorno de função:

```
function calculaIMC(altura, peso) {
    var imc = peso / (altura * altura);
    return imc;
}
escreve("Meu IMC é: " + calculaIMC(MINHA_ALTURA, MEU_PESO));
escreve("O do colega é: " + calculaIMC(ALTURA_COLEGA,
PESO_COLEGA));
```

- **[ADD]** Arredondando o resultado:

```
    return Math.round(imc);
```

- **[ADD]** Determinando o número de casas decimais:

```
    return imc.toFixed(2);
```

4.7. Entrada de dados

- **[ADD]** Alterar o programa para calcular o IMC de cada pessoa é ruim. Use entrada de dados:

```
var altura = prompt("Bom dia! Qual é a sua altura?");
var peso = prompt("E o seu peso?");
escreve("Seu IMC é: " + calculaIMC(altura, peso));
```

- **[NEW]** Novo exemplo: cálculo de pontos de um time de futebol. Mantenha as funções `pulaLinha()` e `escreve()`. O que acontece com esse programa? Por que não soma?

```
var time = prompt("Bom dia! Para que time de futebol você torce?");
var vitorias = prompt("Quantos jogos o " + time + " já ganhou esse
ano?");
```

```
var empates = prompt("E quantos jogos empatou?");
var pontos = (vitorias * 3) + empates;
escreve("O " + time + " tem " + pontos + " pontos.");
```

- **[ADD]** A multiplicação converte vitorias para um tipo numérico (o que aconteceu também no `calculaIMC()` anteriormente), porém o símbolo + é considerado uma concatenação. Isso ocorre porque `prompt()` retorna uma *string*. Correção:

```
var empates = parseInt(prompt("E quantos jogos empatou?"));
```

4.8. Controle de fluxo

- **[ADD]** Compare o time com um rival:

```
var timeRival = prompt("Qual é o time rival do " + time + "?");
var vitoriasRival = prompt("Quantos jogos o " + timeRival + " já
ganhou esse ano?");
var empatesRival = parseInt(prompt("E quantos jogos empatou?"));
var pontosRival = (vitoriasRival * 3) + empatesRival;
escreve("O " + timeRival + " tem " + pontosRival + " pontos.");

if (pontos > pontosRival) {
     escreve("Procure um torcedor do " + timeRival + " para tirar
onda!");
}
else {
     if (pontos == pontosRival) {
          escreve("Pelo menos não estamos atrás do " + timeRival +
"!");
     }
     else {
          escreve("Se encontrar um torcedor do " + timeRival + ",
disfarça e sai de fininho pra não ser zoado...");
     }
}
```

- **[ADD]** Encadeamento else-if:

```
else if (pontos == pontosRival) {
```

- **[NEW]** Escolha com switch-case (sempre mantendo a função `escreve()`):

```
var mes = prompt("Digite o número de um mês do ano (1 = Janeiro, 2 =
Fevereiro, etc.).");
mes = parseInt(mes);

switch (mes) {
  case 1, 3, 5, 8, 10, 12:
    escreve("O mês " + mes + " tem 31 dias.");
    break;

  case 2:
    escreve("Fevereiro tem 28 ou 29 dias, dependendo do ano.");
```

```
    break;

  case 4, 6, 7, 9, 11:
    escreve("O mês " + mes + " tem 30 dias.");
    break;

  default:
    escreve(mes + " não é um número de mês válido.");
}
```

- **[NEW]** Números de Fibonacci (Aula 5, Exercício 5) com laço contado:

```
var n = prompt("Quantos números de Fibonacci você quer imprimir?");
var ant1 = 1;
var ant2 = 0;

if (n > 0) document.write("0, ");
if (n > 1) document.write("1, ");

var i;
for (i = 3; i <= n; i++) {
var atual = ant1 + ant2;
     document.write(atual + ", ");
     ant2 = ant1;
     ant1 = atual;
}
```

- **[NEW]** Fatorial com laço condicional (Aula 5, Exercício 7):

```
var fat = 1;
var n = prompt("De que número você quer calcular o fatorial?");
var original = n;

while (n > 0) {
     fat = fat * n;
     n--; // n = n - 1;
}

escreve(original + "! = " + fat);
```

5. Outros exercícios

- Os alunos ao final da aula podem tentar resolver outros exercícios em JavaScript.

6. Começando com C

- Fazer um "Hello, World!" em C e mostrar o processo de compilação. Detalhes da linguagem na aula seguinte.