



ECM252 – Linguagens de Programação II

5ª lista de exercícios

Regras:

- Responder cada questão como um projeto Web contendo (se aplicável) páginas HTML. Arquivos Javascript e arquivos CSS, organizados convenientemente em pastas distintas;
- Cada arquivo deverá ser comentado apropriadamente;
- Enviar um único arquivo ZIP por equipe, devidamente identificado internamente e contendo todos os projetos solução desta lista para o link indicado no Moodle da disciplina.

Peso das questões:

- Até 2,0 pontos cada questão.
- (1) Elaborar uma função Javascript denominada `rotateText(id,dt)` que, ao ser executada, exibirá em um intervalo `dt` (milissegundos) o texto presente no elemento com propriedade `id` igual à passada no primeiro argumento da função rotacionado de um caractere à direita. Por exemplo, se em um `div` de sua página inicialmente houver o texto Olá Mundo!, espera-se que este texto seja substituído pelos seguintes, eternamente, até que a página seja fechada:

```
Olá Mundo!
!Olá Mundo
o!Olá Mund
do!Olá Mun
ndo!Olá Mu
undo!Olá M
Mundo!Olá
  Mundo!Olá
á Mundo!Ol
lá Mundo!O
Olá Mundo!
```

...

Nesta função, utilizar a função `document.getElementById(id)` para obter o elemento (`e`) cujo texto será manipulado. Para obter o texto do elemento, já que pode ser qualquer elemento HTML, utilizar a propriedade `e.childNodes[0]` (assumindo que este elemento não contém nenhum outro elemento interno, além de seu texto interno) e então obtendo o texto com a propriedade `data`. Para gerar o texto modificado em intervalos de `dt` milissegundos, ainda na sua função, utilizar a função Javascript do objeto `window`, `setInterval(fun, dt)`, que executará de `dt` em `dt` milissegundos a função sem parâmetros `fun` (é mais prático criar uma função anônima para isso). Nesta função, escreva a lógica que fará a rotação do texto do elemento `e`, em seguida, atualize-o no documento com a propriedade `data` citada anteriormente. Executar a função

rotateText quando a janela estiver pronta, no evento onload de body, por exemplo:

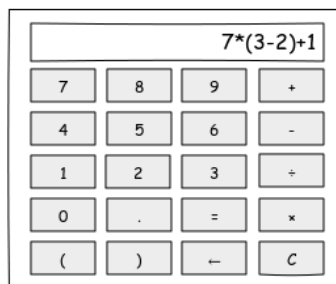
```
<body onload="rotateText(id,1000);">
```

ou (escolha um ou outro), no bloco de script Javascript, após a definição da função rotateText, com a função addEventListener de window, por exemplo:

```
window.addEventListener('DOMContentLoaded', function() {  
    rotateText(id, 1000);  
}, false);
```

Saiba mais sobre eventos Javascript: <https://developer.mozilla.org/en-US/docs/Web/Events>.

- (2) Elaborar uma calculadora com HTML5 e Javascript. A calculadora deverá contemplar o leiaute e as funções conforme descritas pelo esboço a seguir:



Caberá ao desenvolvedor definir estilos que tornem a calculadora “atraente” e usável (cores, fontes etc). Funcionamento esperado:

- Ao se clicar ou digitar o símbolo do botão (fora de qualquer elemento) o símbolo correspondente deverá ser apresentado no visor. O efeito desta ação é acumulativo, tal como apresentado no esboço anterior;
- Ao se clicar ou digitar o símbolo “=”, a expressão apresentada no visor deverá ser calculada e o conteúdo do visor ser substituído pelo resultado calculado;
- Se houver um erro na expressão, deixe o visor com a expressão como estiver ou, se desejar, defina algum meio de sinalização para o usuário. Para interceptar o erro, utilize o tratamento de exceção do Javascript (veja em “Dicas”);
- O usuário poderá corrigir sua expressão digitando a tecla “backspace” ou clicando no botão “←”;
- Ao se clicar no botão “C” ou teclar “CTRL+backspace” limpa-se o visor;
- Não é necessário manipular o cursor do visor, permitindo alterar partes selecionadas e nem realizar operações de cópia, colar e recortar.

Dicas:

- Não é necessário construir um analisador de expressões. Javascript conta com a função `eval(str)` que analisa a expressão `str` e, se não houver erros, resulta no valor da expressão. No exemplo apresentado, o valor resultante poderia ser calculado como `eval('7*(3-2)+1')`;
- Se a expressão analisada contiver erro, `eval` resulta em uma exceção. Para tratar exceções em Javascript, utilize bloco `try-catch`. Experimentar:

```
try{eval("1/");} catch(err){console.log("Erro:" +  
err.message);};
```

- Para instalar um tratador de teclado que esteja associado globalmente ao documento pode-se utilizar o evento `keydown`, como no exemplo a seguir:

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>Exercício 2</title>  
</head>  
<body>  
  <div id="results"></div>  
  <script async defer type="text/javascript">  
    document.addEventListener("keydown", function(evt) {  
      evt = evt || window.event;  
      var d = document.getElementById("results");  
      var p = document.createElement("p");  
      var c = evt.charCode || evt.keyCode;  
      var ch = String.fromCharCode(c);  
      var t = document.createTextNode("CTRL: "  
        + evt.ctrlKey + " TECLA: "  
        + ch + "(" + c + ")");  
      p.appendChild(t);  
      d.appendChild(p);  
      if( c === 8) { //Evitar que backspace leve à página anterior  
        evt.preventDefault();  
      }  
    }, false);  
  </script>  
</body>  
</html>
```

- É de sua escolha os elementos que constituirão as “partes” da calculadora: utilizando CSS3 você poderá utilizar `span`, `button`, etc.

- (3) Elaborar uma página HTML que permita ao usuário entrar em um campo texto do tipo `input` o número de um cartão de crédito. Ao se digitar o número, caso seja identificado que o cartão em questão possui bandeira VISA, MASTERCARD ou AMERICAN EXPRESS, ao lado do campo deverá ser apresentado uma pequena imagem com a imagem de frente de um desses cartões. Além disso, enquanto se digita o número do cartão, se este for inválido (tamanho inválido e /ou dígito verificador inválido), o campo deverá ser destacado de modo que o usuário perceba que algo está errado com o número fornecido. Por outro lado, se o número for válido, o campo deverá ser destacado de modo que o usuário tenha certeza que o número está correto. Para conhecer o algoritmo de validação de números de cartões de crédito: <http://gizmodo.uol.com.br/como-sao-criados-os-numeros-de-cartao-de-credito/>.
- (4) Em HTML5 existe o elemento `canvas`, que proporciona uma área de desenho que possui funções que permite a criação de aplicações multimídia, gráficos, jogos etc. Uma aplicação de `canvas` para a criação de gráficos bidimensionais pode ser obtida aqui: <http://www.html5canvastutorials.com/labs/html5-canvas-graphing-an-equation/>.

Existem, também, bibliotecas Javascript que realizam parsing de expressões. Uma delas é o **jsep** (<http://jsep.from.so>), que retorna uma árvore de parsing contendo informações sobre a expressão em questão.

Pede-se: elaborar uma aplicação HTML+Javascript de modo que o usuário digitará em um

campo texto do tipo `input` uma expressão em função de apenas uma variável. Depois, se houver a produção de uma árvore de parsing, traduzir seus elementos em uma cadeia Javascript que possa ser interpretada e permitir a geração de um gráfico. Você pode escolher um conjunto de operações e funções matemáticas que seu sistema aceitará (documento claramente) e rejeitar as demais. Por exemplo, a entrada `1+exp(-0.5*x)*sin(x)` poderia ser traduzida em uma cadeia como `1+Math.exp(-0.5*x)*Math.sin(x)` e então ser passada à função `eval()` que, dentro de um laço de repetição conveniente, poderia ser usada para plotar o gráfico desta função (mas é possível utilizar outros meios, também).

- (5) O site <http://developers.agenciaideias.com.br/tempo> oferece uma API que permite retornar a previsão do tempo para o dia de hoje e também para uma semana à frente. Elaborar uma aplicação HTML+Javascript de modo a utilizar o objeto `XMLHttpRequest` para enviar o pedido da previsão do tempo para o local atual que você está utilizando seu navegador. Para saber o local atual que você está utilizando seu navegador, você pode utilizar a api de geolocalização acompanhado de algum serviço online que retorne a sua cidade e estado a partir das informações de geolocalização. O código a seguir exemplifica como fazer isso com a API de geolocalização de HTML5 e o site Open Street Map (funciona no Firefox; no Chrome, deve-se permitir o acesso aos dados de geolocalização):

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Geolocalização</title>
</head>
<body>
  <script async defer type="text/javascript">
    navigator.geolocation.getCurrentPosition(function(posicao) {
      console.log(posicao);
      var url = "http://nominatim.openstreetmap.org/reverse?lat="
        + posicao.coords.latitude + "&lon="
        + posicao.coords.longitude
        + "&format=json&json_callback=preencherDados";

      var script = document.createElement('script');
      script.src = url;
      document.body.appendChild(script);

    });
    function preencherDados(dados) {
      alert(dados.address.city);
      alert(dados.address.state);
    }
  </script>
</body>
</html>
```

Na aplicação a ser desenvolvida, apresentar automaticamente os dados de previsão do tempo (com imagens que facilitem a interpretação – caprichar na usabilidade), com o nome da cidade e estado do local de onde se abriu a página.