



PROGRAMAÇÃO EM AMBIENTE WEB I

Lista de Exercícios I

Instruções gerais

Atividade **individual**

Prazo de entrega: 23:59:59 GMT -4 do dia 21/08/2016.

Forma de entrega: Enviar por e-mail, usando a tag **L1PW120161** no **assunto** um arquivo compactado (.zip) contendo todo o conteúdo desenvolvido e imagens utilizadas. Verifique o conteúdo do arquivo compactado antes de enviar! O nome do arquivo compactado deve seguir o formato **<NOME>_<RGA>**.

Valor: 0.0-1.0 pontos

BOA PROVA!

Instruções da atividade

Para realizar estas atividades, você deve criar uma página em formato HTML devidamente formatada de acordo com o que aprendemos em sala (lembre-se da Aula 1). Cada exercício deve estar contido em um bloco (use as novas tags para organizar artigo, seção, etc) diferente, com um título identificando o exercício (Exercício 1, Exercício 2...), com seu enunciado e o trecho HTML da solução (semanticamente marcado, também podendo usar a nova tag semântica **header** para separar cabeçalho do exercício do conteúdo). A página também deve ter um menu (usando a tag semântica **nav**), contendo uma âncora/link para cada exercício (para isso, cada exercício (seção) deve ter um id próprio; no href da âncora/link, use "#id", sendo id o identificador único do exercício). Nos códigos CSS e Javascript utilizados identifique, utilizando a sintaxe de comentário de cada linguagem, a qual exercício se refere aquela formatação ou trecho de código/função. É **fundamental** que os códigos Javascript apresentados funcionem. Portanto, testem antes de enviá-los! **Quaisquer dúvidas** podem ser enviadas para cslaviero@gmail.com.

Exercícios:

- 1) Adicione dois vídeos em sua página (dica: o site www.sample-videos.com tem vários vídeos de 1MB de exemplo). Nenhum dos dois vídeos pode tocar ao mesmo tempo, ou seja, você deve, na execução de um vídeo (onplay), pausar (pause()) o outro vídeo.
- 2) Defina uma seção com um SVG que contém um círculo, e faça dois botões. Um aumenta o tamanho do círculo, outro diminui (dica: `getAttribute()`).
- 3) Crie um formulário com no mínimo três campos distintos e um botão. Ao se clicar no botão, o formulário, além ser direcionado para a mesma página (atributo `action`), deve ter suas informações armazenadas no `localStorage` (ou `sessionStorage`). Ao carregar a página novamente, as



informações devem ser recuperadas do mesmo localStorage/sessionStorage.

- 4) Crie um canvas e adicione abaixo dele um botão que cria retângulos de tamanhos e posições diferentes no canvas. Dica: utilize `Math.floor((Math.random() * 100) + 1)` para gerar números aleatórios de 0 a 100. Modifique o número 100 caso precise gerar mais ou menos número aleatórios.
- 5) Crie um botão em sua página que recupera a localização do usuário. Além disso, crie dois botões, um para aumentar a latitude, e outro a longitude. Cada vez que essa informação for alterada, (re-)desenhe um mapa estático da localização, usando o endereço do da API do Google Maps apresentada nos slides. Dica: O google chrome tem restrições no uso de localização por documentos html locais. Use o Firefox em sua última versão, aonde este problema não acontece.