



Missão 42

Cell 3-3

*Sumário: Nesta célula nos vamos ver como javascript funciona.*

*Versão:*

# Conteúdo

<b>I</b>	<b>Sobre a Missão 42</b>	<b>2</b>
<b>II</b>	<b>Introdução</b>	<b>3</b>
<b>III</b>	<b>Intruções Gerais</b>	<b>4</b>
<b>IV</b>	<b>Exercício 03: To do or not to do</b>	<b>5</b>
<b>V</b>	<b>Submissão e avaliação em pares</b>	<b>6</b>

# Capítulo I

## Sobre a Missão 42

Bem-vindo !

Você está começando a primeira célula desta Missão de programação de computadores. Nós queremos mostrar a você que o código é o que faz o software que você usa todos os dias, e ao mesmo tempo experimentar o modelo de aprendizado 42 de peering.

Programação envolve lógica (não matemática). Ele fornece a você blocos de construção elementares, que você pode assemelhar como desejar. Nunca houve uma solução para um problema. Há a sua solução e as de cada um de seus vizinhos. Lento ou rápido, feio ou belo, se isso faz o trabalho, isso é tudo que precisa! Essa montagem de blocos de construção fará uma série de ordens (cálculo, exibição ... ) que o computador irá realizar, na ordem que você escolher.

Em vez de dar a você um curso com apenas uma solução para cada problema e que provavelmente estarão desatualizados em alguns anos, nós optamos por colocá-lo em uma situação de aprendizado de pares. Você encontrará os elementos que podem servir para o seu desafio, organizar os que são realmente interessantes ao testar e manipulando-os e criar seu próprio programa. Para fazer isso, discuta com outros participantes, troque seus pontos de vista, encontre novas ideias juntos e finalmente teste por si mesmo para se convencer sobre o que funciona.

Avaliação em pares é um momento chave para descoberta de maneiras diferentes de fazer coisas, bem como casos especiais que você não pensou e que podem quebrar seu programa (pense em seu grau nervosismo com software que trava). Como diferentes clientes que não prestam atenção aos mesmos problemas, cada avaliador será diferente dos outros. E quem sabe, você pode ter feito novas amizades para colaborações futuras.

No final desta piscina, você não fará o mesmo que os outros participantes, não validou projetos iguais e escolheu uma desafio em vez de outro, e isso é normal! É tanto uma experiência coletiva quanto pessoal. Todos terão benefícios do que experimentam durante esse tempo.

Boa sorte a todos, esperamos que você goste da descoberta.

# Capítulo II

## Introdução

O que essa célula mostrará para você:

- Descobrir como uma página web funciona.
- Descobrir o Javascript.

# Capítulo III


## Instruções Gerais

A menos que seja especificado explicitamente, as seguintes regras se aplicarão todos os dias dessa Piscina.

- Este arquivo é a única fonte confiável. Não confie em quaisquer boatos.
- Este arquivo pode ser atualizado até uma hora antes da data final de entrega.
- As atribuições em um projeto devem ser feitas na ordem dada. Os exercícios posteriores não serão avaliados se os anteriores não forem executados perfeitamente.
- Cuidado com os direitos de acesso dos seus arquivos e pastas.
- Seus trabalhos serão avaliados pelos pares da sua Missão.
- Todos os trabalhos de shell devem ser rodados usando `/bin/bash`.
- Você não deve deixar no envio do seu espaço de trabalho nenhum arquivo fora os que foram explicitamente requisitados pelo trabalho.
- Você tem uma dúvida? Pergunte ao seu vizinho à esquerda. Caso contrário, tente sua sorte com o vizinho da direita.
- Qualquer resposta técnica que você precisa está disponível no `man` ou na internet.
- Você deve ler os exemplos cuidadosamente. Eles podem revelar requisitos que não estão óbvios na descrição do projeto.

# Capítulo IV

## Exercício 03: To do or not to do

	Exercício : 03
To do or not to do	
Pasta de entrega : <i>ex03/</i>	
Ficheiros para entregar : <i>index.html</i> , <i>todo.js</i>	
Funções autorizadas : HTML, CSS, JS	

Para este exercício, teremos que criar um mini gerenciamento de tarefas locais. O design não importa desde que a estrutura apresentada a seguir seja respeitada. Seja criativo, mas concentre-se na prioridade das funcionalidades.

A lista de tarefas será representada por uma `div` que terá como atributo `id` o valor `'ft_list'`. Este bloco contém a lista de "TAREFAS". Cada TAREFA é representado por uma `div` contida no bloco `'ft_list'`. Quando uma TAREFA é criada, ela é colocado no topo da lista. Cabe a você criar o elemento e colocá-lo no lugar certo (Manipulação de DOM).

Deve haver um botão de criação chamado `'Adicionar tarefa'`. Quando clicado, ele abra uma janela de texto (verifique a função de `prompt`) que permitirá ao usuário preencher uma nova TAREFA. Uma vez validado, se não estiver vazio, deve aparecer no topo da lista.

Para remover uma TAREFA da lista, basta clicar nela. A janela de configuração deve abrir e perguntar se você deseja ou não remover a TAREFA. Se você confirmar, a TAREFA deverá desaparecer permanentemente do DOM, ela não pode simplesmente ser ocultada.

Pequena implementação adicional, sua lista de tarefas terá que ser salva como um cookie. Se a lista contiver alguma tarefa ao fechar o navegador, esta mesma lista deve ser carregada e exibida em `'ft_list'`. Se o(s) cookie(s) não existir, a lista ficará em branco.

# Capítulo V

## Submissão e avaliação em pares

- Crie uma pasta `missao_42` na raiz do seu diretório home e navegue até lá.
- Crie uma pasta `cell03` e navegue para lá.
- A partir de agora, todos os exercícios devem estar na pasta correta. Exercício 00 naa pasta `ex00`, Exercício 01 na pasta `ex01`, etc... Você entendeu a lógica.



Durante a defesa tudo o que não está presente na pasta para a avaliação será ignorado.