02

(Obrigatório) Download do projeto e infraestrutra

O arquivo do projeto

Nosso projeto se chamará *aluraframe* e deve ser baixado aqui (https://github.com/alura-cursos/javascript-avancado-i/archive/aula1.zip). É um arquivo zip que deve ser descompactado em sua Área de Trabalho ou qualquer outro diretório que você esteja confortável em trabalhar.

Existem duas subpastas dentro do projeto: *server* e *client*. A primeira pasta possui um servidor Node.js que será usado mais tarde, quando formos avançando com o treinamento, e pode ser ignorada por enquanto.

A segunda pasta é aquela que será carregada pelo nosso servidor, com a página `index.html` e todos os scripts que criaremos durante o treinamento. É nesta pasta que você deve focar primariamente.

Por fim, dentro de `aluraframe/client` temos algumas subpastas já criadas, não se preocupe com elas. Cada uma será esclarecida ao longo do curso.

Infraestrutura 1 - Chrome!

Neste treinamento é necessário usar o **Google Chrome versão 50** ou superior. **É importante essa escolha do navegador**, porque estaremos usando muitos recursos do ES6, inclusive alguns que estão sendo consolidados. Se você está inseguro, não fique. Todos os browsers hoje possuem atualização automática e desde o IE 10 a atualização do browser na plataforma Windows ficou independente do sistema operacional, isso significa que em pouquíssimas semanas todos os usuários de internet acabam recebendo o browser mais novo.

ATENÇÃO USUÁRIOS MAC: nem sempre o Chrome é instalado na língua **português-brasil**. Se o seu navegador exibe todas as opções de menu em inglês, ele exibirá a data da tag `<input type="date">` no formato `mês/dia/ano`. Não há problema nenhum ir até o final do curso neste formato, mas se você quiser que o `input` exiba a data no formato `dia/mês/ano`, precisará usar a versão em **português-brasil* do Chrome. Você não precisa baixar novamente seu navegador, basta abrir seu terminal e executar o comando:

```
exec defaults write com.google.Chrome AppleLanguages '(pt-BR)'
```

Veja que essa solução só é necessária se você deseja a data no formato bonitinho. Durante o treinamento, eu preferi pegar carona neste tipo de `input` ao invés de escrever um código de formatação. Existem zilhões de soluções aí fora para formatação, mas a ideia é usarmos tudo o que o browser oferece.

DICA: Como saber se determinada funcionalidade do ES2015+ (ES6) é suportada por cada navegador?

Durante o treinamento, evitarei dizer que determinada funcionalidade só funciona nesse ou naquele browser, porque pode ser que na semana seguinte o browser X passe a suportá-la. Contudo, para que o aluno tenha a informação mais recente dos recursos que utilizarei neste treinamento, ele pode consultar sempre que desejar: https://kangax.github.io/compat-table/es6/ (https://kangax.github.io/compat-table/es6/).

Não ache que os recursos que lhe mostrarei são experimentais, muito pelo contrário, eles já fazem parte da especificação ES6. Inclusive o conhecimento aqui adquirido pode ser aplicado na plataforma Node.js v6.0 ou superior,

exceto a parte de manipulação de DOM. Além disso, se você desenvolve usando algum tipo de *transpiler*, como Babel ou até mesmo o famoso TypeScript, também será beneficiado. Com essas tecnologias, escrevemos um código em ES6 e no final entregamos para o navegador o mesmo código convertido para ES5, para garantir máxima compatibilidade. Sendo assim, tudo o que você aprender aqui pode ser usado.

Infraestrutura 2 - Node.js

Em um certo ponto do nosso treinamento, precisaremos de um servidor web que disponibilize URL's para serem consumidas pela nossa aplicação. Já disponibilizamos um para você dentro do projeto. Para que ele funcione, você precisa ter no mínimo o Node.js **v4.0** instalado.

Você pode baixar o Node.js da sua plataforma preferida (Windows, MAC ou Linux) em https://nodejs.org (https://nodejs.org). Depois de instalá-lo, para saber se ele está funcionando, basta abrir seu terminal preferido da sua plataforma e executar o comando (logo a seguir, veja as dicas caso o comando não funcione):

```
node --version
```

Este comando deve exibir a versão do Node instalada no terminal. Se por acaso o comando `node` não for um comando válidos tente o seguinte:

1 - Windows: não mude o diretório padrão da instalação do Node.js. Há relatos que em algumas versões do Windows a pasta do Node não é colocada no *PATH* do Windows, sendo necessário adicioná-la manualmente. Não sabe como? Temos um treinamento de prompt no Windows (https://www.alura.com.br/curso-online-prompt) que pode ajudá-lo nesta tarefa.

2 - Linux: algumas distribuições Linux já possui um binário chamado **node**, que não tem nenhuma relação com o Node.js. Nestas distribuições, o binário passa a se chamar **nodejs**. Sendo assim, em todo lugar que eu referenciar o comando **node** ele deve ser trocado para **nodejs**.