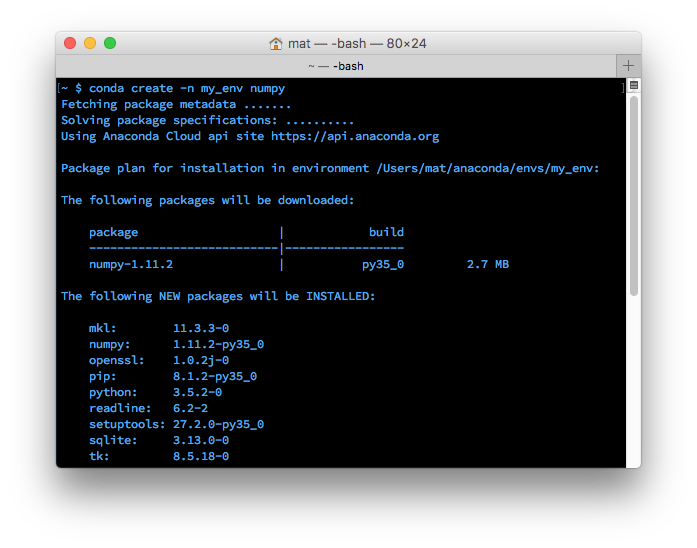
Gerenciando ambientes

Como foi mencionado anteriormente, o conda pode ser utilizado para criar ambientes que isolem seus projetos. Para criar um ambiente, use conda create -n nome\_amb lista de pacotes no seu terminal. Aqui, -n nome\_amb dá o nome do seu ambiente (-n de nome) e lista de pacotes é a lista de pacotes que você quer instalada no ambiente. Por exemplo, para criar um ambiente chamado my\_env e instalar o numpy nele, digite conda create -n my\_env numpy.



Ao criar um ambiente, você pode especificar qual versão do Python você quer instalada nele. Isso é útil para quando você trabalha com código tanto em Python 2.x quanto em Python 3.x. Para criar um ambiente com uma versão específica de Python, faça algo como conda create -n py3 python=3 ou conda create -n py2 python=2. Eu tenho ambos os ambientes no meu computador pessoal. Eu uso os dois como ambientes gerais que não estão atrelados a nenhum projeto específico, mas sim para trabalhos gerais com cada versão do Python facilmente acessível. Esses comandos irão instalar as versões mais recentes de Python 3 e Python 2, respectivamente. Para instalar uma versão específica, use, por exemplo conda create -n py python=3.3 para a versão 3.3 do Python.

Entrando em um ambiente

Ume vez criado o ambiente, use o comando source activate my\_env para entrar nele no OSX/Linux. No Windows, use activate my\_env.

Quando estiver no ambiente, você verá o nome dele no prompt do terminal. Algo como (meu\_amb) ~ $. O ambiente tem apenas alguns pacotes instalados automaticamente, além daqueles inseridos no comando de criação. É possível checar isso usando o comando conda list. Instalar pacotes no ambiente é feito da mesma maneira: conda install nome\_do\_pacote. Apenas com uma diferença: dessa vez, os pacotes específicos que forem instalados estarão disponíveis apenas enquanto o ambiente estiver ativo. Para sair do ambiente, digite source deactivate (no OSX/Linux). No Windows, use deactivate.