**Motivation**[**¶**](#gjdgxs)

Hello, and welcome! We're excited to be your gateway into machine learning. ML is a rapidly growing field that's buzzing with opportunity. Why? Because the tools and skills employed by ML specialists are extremely powerful and allow them to draw conclusions from large data sets quickly and with relative ease.

Take the Celeste project, for example. This is a project that took 178 **tera**bytes of data on the visible sky and used it to catalogue 188 millions stars and galaxies. "Cataloguing" these stars meant identifying characteristics like their locations, colors, sizes, and morphologies. This is an amazing feat, *especially* because this entire calculation took under 15 minutes.



How are Celeste's calculations so fast? To achieve performance on this scale, the Celeste team uses the Julia programming language to write their software and supercomputers from Lawrence Berkeley National Lab's NERSC as their hardware. In this course, we unfortunately won't be able to give you access to a top 10 supercomputer, but we will teach you how to use Julia!

We're confident that this course will put you on your way to understanding many of the important concepts and "buzz words" in ML. To get you started, we'll teach you how to teach a machine to tell the difference between images of apples and bananas, i.e to **classify** images as being one or the other type of fruit.

Like Project Celeste, we'll use the [Julia programming language](https://julialang.org/) to do this. In particular, we'll be working in [Jupyter notebooks](http://jupyter.org/) like this one! (Perhaps you already know that the "ju" in Jupyter comes from Julia.)

**What do the images we want to classify look like?**[**¶**](#30j0zll)

We can use the Images.jl package in Julia to load sample images from this dataset. Most of the data we will use live in the data folder in this repository.

In [1]:

**using** Images *# To execute hit <shift> + enter*

In [2]:

apple = load("data/10\_100.jpg")

Out[2]:



In [3]:

banana = load("data/104\_100.jpg")

Out[3]:



The dataset consists of many images of different fruits, viewed from different positions. These images are [available on GitHub here](https://github.com/Horea94/Fruit-Images-Dataset).

**What is the goal?**[**¶**](#1fob9te)

The ultimate goal is to feed one of these images to the computer and for it to identify whether the image represents an apple or a banana! To do so, we will **train** the computer to learn **for itself** how to distinguish the two images.

The following notebooks will walk you step by step through the underlying math and machine learning concepts you need to know in order to accomplish this classification.

They alternate between two different types of notebooks: those labelled **ML** (Machine Learning), which are designed to give a high-level overview of the concepts we need for machine learning, but which gloss over some of the technical details; and those labelled **Tools**, which dive into the details of coding in Julia that will be key to actually implement the machine learning algorithms ourselves.

The notebooks contain many **Exercises**. By doing these exercises in Julia, you will learn the basics of machine learning!

**Course outline**[**¶**](#3znysh7)

The course notebooks are listed below. We recommend that you follow them in order.

* [01. Representing data in a computer](http://docs.google.com/01.%20ML%20-%20Representing%20data%20in%20a%20computer.ipynb)
* [02. Using arrays to store data](http://docs.google.com/02.%20Tools%20-%20Using%20arrays%20to%20store%20data.ipynb)
* [03. Representing data with models](http://docs.google.com/03.%20ML%20-%20Representing%20data%20with%20models.ipynb)
* [04. Functions](http://docs.google.com/04.%20Tools%20-%20Functions.ipynb)
* [05. Building models](http://docs.google.com/05.%20ML%20-%20Building%20models.ipynb)
* [06. Adding a function parameter](http://docs.google.com/06.%20Tools%20-%20Adding%20a%20function%20parameter.ipynb)
* [07. Model complexity](http://docs.google.com/07.%20ML%20-%20Model%20complexity.ipynb)
* [08. Multiple function parameters](http://docs.google.com/08.%20Tools%20-%20Multiple%20function%20parameters.ipynb)
* [09. What is learning](http://docs.google.com/09.%20ML%20-%20What%20is%20learning.ipynb)
* [10. Minimizing functions - how a computer learns](http://docs.google.com/10.%20Tools%20-%20Minimizing%20functions%20-%20how%20a%20computer%20learns.ipynb)
* [11. Intro to Neurons](http://docs.google.com/11.%20ML%20-%20Intro%20to%20Neurons.ipynb)
* [12. Learning with a single neuron](http://docs.google.com/12.%20Tools%20-%20Learning%20with%20a%20single%20neuron.ipynb)
* [13. Intro to Flux.jl](http://docs.google.com/13.%20ML%20-%20Intro%20to%20Flux.jl.ipynb)
* [14. Learning with a single neuron using Flux.jl](http://docs.google.com/14.%20Tools%20-%20Learning%20with%20a%20single%20neuron%20using%20Flux.jl.ipynb)
* [15. Intro to neural networks](http://docs.google.com/15.%20ML%20-%20Intro%20to%20neural%20networks.ipynb)
* [16. Using Flux to build a single layer neural net](http://docs.google.com/16.%20Tools%20-%20Using%20Flux%20to%20build%20a%20single%20layer%20neural%20net.ipynb)
* [17. Introduction to deep learning](http://docs.google.com/17.%20ML%20-%20Introduction%20to%20deep%20learning.ipynb)
* [18. Multi-layer neural networks with Flux](http://docs.google.com/18.%20Tools%20-%20Multi-layer%20neural%20networks%20with%20Flux.ipynb)
* [19. Recognizing handwriting using a neural network](http://docs.google.com/19.%20Recognizing%20handwriting%20using%20a%20neural%20network.ipynb)

Let's get started with [Representing data in a computer](http://docs.google.com/01.%20ML%20-%20Representing%20data%20in%20a%20computer.ipynb)!