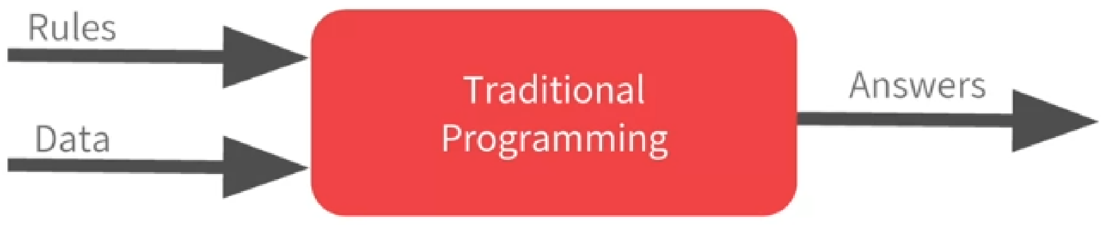
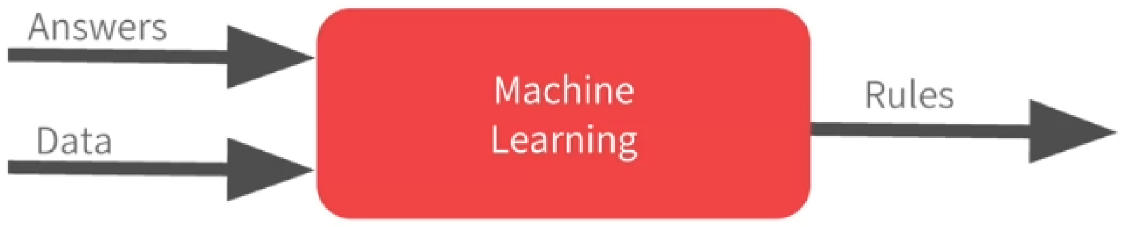
# Week 1

## A primer in machine learning

- Traditional Programming works with rules and answers



- Machine learning gets rules out of Answers



## "Hello World" of Neural Network

main File

```
# ---------------------------------------------------------------------------- #
# Importing required libraries including tensorflow, keras and numpy
import tensorflow as tf
from tensorflow import keras
import numpy as np
# ---------------------------------------------------------------------------- #

# ---------------------------------------------------------------------------- #
# Define and compile the network
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
# Create layers of neurons
model.compile(optimizer='sgd', loss='mean_squared_error')
```

```
# This measures how bad or how good the model performs
# --------------------------------------------------------------------------- #

# --------------------------------------------------------------------------- #
# Providing the data
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)
# Data for x and data for y
# --------------------------------------------------------------------------- #


# --------------------------------------------------------------------------- #
# Training the data

model.fit(xs, ys, epochs=500)
# Testing hte model with 500 epocs (500 times)

print(model.predict([10.0]))
# predict the model with x of 10
# --------------------------------------------------------------------------- #
```

## Form Rules to Data

In these videos you were given an introduction to the concepts and paradigms of Machine Learning and Deep Learning. You saw that the traditional paradigm of expressing rules in a coding language may not always work to solve a problem. As such, scenarios such as Computer Vision are very difficult to solve with rules-based programming. Instead, if we feed a computer with enough data that we describe (or label) as what we want it to recognize, given that computers are really good at processing data and finding patterns that match, then we could potentially ‘train’ a system to solve a problem. We saw a super simple example of that -- fitting numbers to a line. So now let’s go through a notebook and execute the code that trains a neural network to learn how a set of numbers we give it make up a line, so it can then extend the line if we need to.

## Weekly test

```
import tensorflow as tf
import numpy as np
from tensorflow import keras

# GRADED FUNCTION: house_model
def house_model(y_new):
    xs = np.array([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], dtype=float)
    ys = np.array([1, 1.5, 2, 2.5, 3, 3.5], dtype=float)
    model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
    model.compile(optimizer='sgd', loss='mean_squared_error')
    model.fit(xs, ys, epochs=500)
    return model.predict(y_new)[0]

prediction = house_model([7.0])
print(prediction)
```