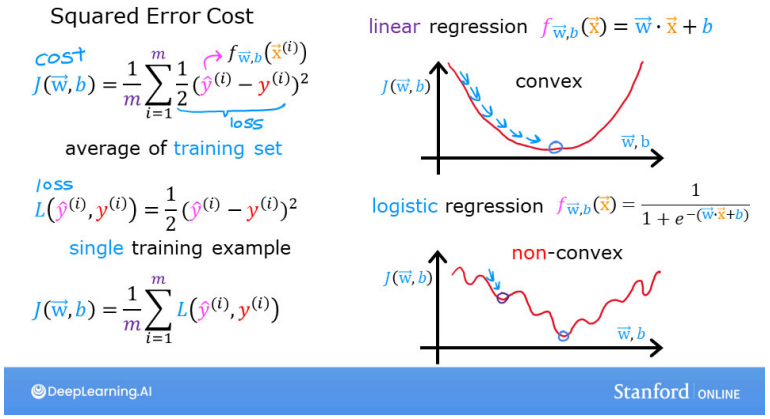
Optional Lab: Logistic Regression, Logistic Loss

In this ungraded lab, you will:

- explore the reason the squared error loss is not appropriate for logistic regression
- explore the logistic loss function

In [1]:
```
import numpy as np
%matplotlib widget
import matplotlib.pyplot as plt
from plt_logistic_loss import  plt_logistic_cost, plt_two_logistic_loss_curves, plt_simple_example
from plt_logistic_loss import soup_bowl, plt_logistic_squared_error
plt.style.use('./deeplearning.mplstyle')
```

Squared error for logistic regression?



Recall for **Linear** Regression we have used the **squared error cost function**: The equation for the squared error cost with one variable is:

$$J(w,b) = \frac{1}{2m}\sum_{i=0}^{m-1}(f_{w,b}(x^{(i)}) - y^{(i)})^2 \tag{1}$$

where

$$f_{w,b}(x^{(i)}) = wx^{(i)} + b \tag{2}$$

Recall, the squared error cost had the nice property that following the derivative of the cost leads to the minimum.