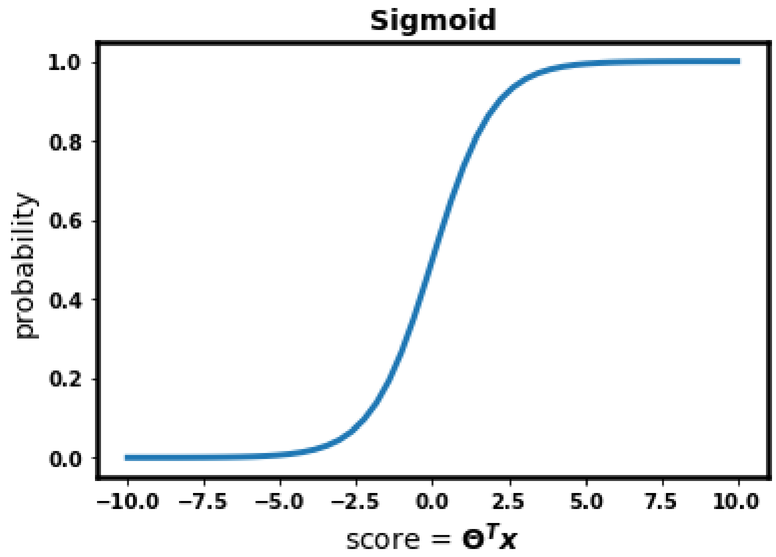
Logistic regression as Linear Regression of log odds

We can demonstrate a relationship between Logistic Regression and Linear Regression.

Recall the mapping of the score $\Theta^T \mathbf{x}$ into probabilities

In [4]:
```
s = np.linspace(-10,10, 50)
sigma_s = 1/(1 + np.exp(- s))

fig = plt.figure()
ax  = fig.add_subplot(1,1,1)
_= ax.plot(s, sigma_s)
_= ax.set_title("Sigmoid")
_= ax.set_xlabel("score = $\Theta^T x$")
_= ax.set_ylabel("probability")
```



Certainly doesn't look like a linear relationship between scores and probability.

Define the *odds* $\mathbf{o^{(i)}}$ of example i being in class 1 as

$$\mathbf{o^{(i)}} = \frac{\hat{p}^{(\mathbf{i})}}{1 - \hat{p}^{(\mathbf{i})}}$$

- the odds is just the ratio of the probability of being in class 1 versus not being in class 1
- **Note** this is called the *odds* **not** the odds ratio !
    - *odds ratio* is the ratio of two odds

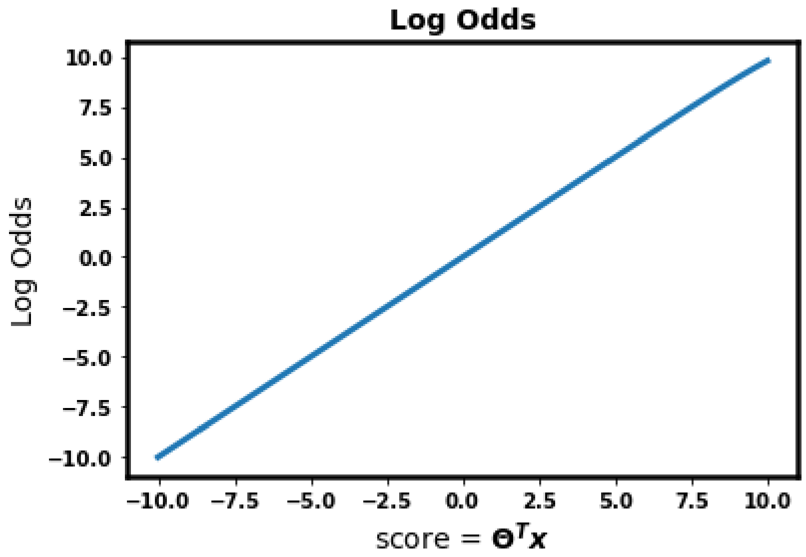
Let's graph the relationship between scores $\Theta^T \mathbf{x}$ and the *log of the odds.*

In [5]:
```
s = np.linspace(-10,10, 50)
sigma_s = 1/(1 + np.exp(- s))

p = sigma_s
epsilon = 10e-6

odds = p/(1 - p + epsilon)
log_odds = np.log(odds)

fig = plt.figure()
ax  = fig.add_subplot(1,1,1)
_= ax.plot(s, log_odds)
_= ax.set_title("Log Odds")
_= ax.set_xlabel("score = $\Theta^T x$")
_= ax.set_ylabel("Log Odds")
```



Linear !

So you can implement Logistic Regression as Linear Regression of the log odds versus features $\mathbf{x}$

This is similar in spirit to our transforming the "curvy" data set of the previous lesson

- there, we transformed features to obtain a linear relationship
- here we transformed the target

So the Logistic Regression equation is the linear equation

$$\log(\mathbf{o}) = \Theta^T \mathbf{x} + \epsilon$$

In words:

- Logistic Regression is Linear Regression to predict log odds, given features $\mathbf{x}$

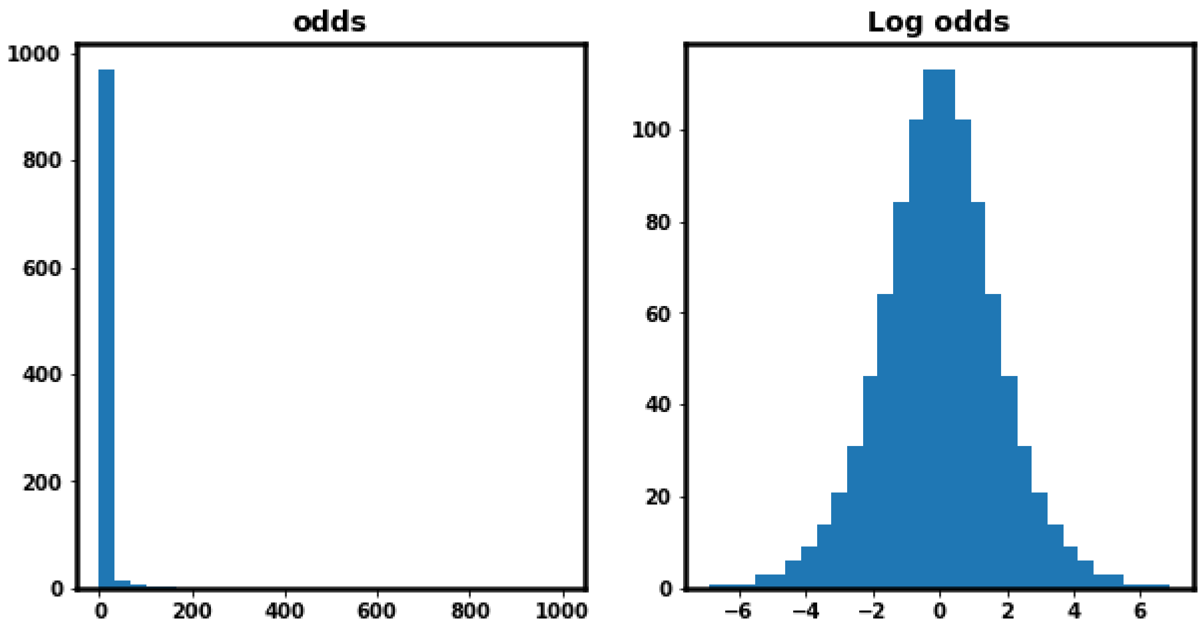
Knowing that the regression produces log odds will become very useful in interpreting coefficients Θ.

(Coming attraction: a unit change in Θ_j results in a *multiplicative* increase in odds)

Log odds are normally distributed

Let's examine the distribution of log odds.

In [6]:
```
tf = tmh.TransformHelper()
tf.plot_odds()
```



- Log of the odds is normally distributed
- Linear Regression errors will be normally distributed, satisfying model's mathematical assumptions

Logistic Regression as Linear Regression on the log odds: complication

Turns out you can't solve for the Θ in Logistic Regression by minimizing the RMSE cost function.

- Observe that
    - the log odds $\log(\frac{\hat{p}}{1-\hat{p}})$ is at $\hat{p} = 1$ $= \infty$
    - the log odds $\log(\frac{\hat{p}}{1-\hat{p}}) =$ is at $\hat{p} = 0$ $-\infty$

This will give infinite errors.

There is an alternate solution to Linear Regression using *Maximum Likelihood* which doesn't have this issue.

n.b., Minimizing RMSE produces a Maximum Likelihood estimate of Θ.

In [7]:

```
print("Done")
```

Done