Building the TTS Engine

To make your software cross-platform between OS X and Linux, you have to determine which OS your software is running on. You can find that out by using sys.platform in Python. The value of sys.platform on Apple systems is Darwin, and on Linux-based systems it is either linux or linux2.

Let's write the Python code to accomplish the task:

```
import os
import sys

def tts(message):
    """
    This function takes a message as an argument and converts it to speech
    depending on the OS.
    """
    if sys.platform == 'darwin':
        tts_engine = 'say'
        return os.system(tts_engine + ' ' + message)
    elif sys.platform == 'linux2' or sys.platform == 'linux':
        tts_engine = 'espeak'
        return os.system(tts_engine + ' "' + message + '"')
```

Let's go through the code. The first two import statements import the os and sys modules. Then you define a function called tts that takes a message as an argument. The if statement determines whether the platform is OS X; then it assigns the say value to the tts_engine variable and returns os.system(tts_engine + ' ' + message). This executes the say command with the message on the terminal. Similarly, if the platform is Linux based, it assigns espeak to the tts_engine variable.

To test the program, you can add the following additional line at the bottom of the code:

```
tts("Hi handsome, this is Melissa")
```

Save the code, and run the Python file. It should execute successfully.

Repeat What I Say

For the sake of exercise and fun, construct a Python program that detects whatever you say and repeats it. This involves a combination of the STT and TTS engines. You have to make the following assignment:

```
message = r.recognize_google(audio)
```