* + use “hacky” gifs of some terrible 1990s hollywood hacker
  + Geek pushing up glasses
  + Crackingknuckles in front of keyboard
  + At end: <https://giphy.com/gifs/keanu-reeves-matrix-the-3o7btNhMBytxAM6YBa>

Dancing baby <https://giphy.com/gifs/dancing-baby-moves-12FwhN6Qh3cfxm>

The more you hack <https://giphy.com/gifs/trutv-hack-hacking-29HN29NtuCd7rw04t8>

Super saiyan brother <https://giphy.com/gifs/super-saiyan-baby-slap-dkqzY20GXyG40>

Super saiyan mom <https://giphy.com/gifs/funny-lady-6KlLzO38CkLjG>

Super saiyan baby <https://giphy.com/gifs/super-saiyan-funny-uAnmHj5wTXxcY>

Ford example

Scraping directly <https://nbviewer.jupyter.org/github/nealcaren/ScrapingData/blob/master/Notebooks/6_Web.ipynb>

* Pandas can scrape if data is in html (and not javascript or other!)
* Try on a wiki page with multiple tables
* LOTS OF GOOD TOYS

Building up a scraper <https://nbviewer.jupyter.org/github/nealcaren/ScrapingData/blob/master/Notebooks/Bonus_Downloading.ipynb>

Undocumented APIs <https://nbviewer.jupyter.org/github/nealcaren/ScrapingData/blob/master/Notebooks/Bonus_Undocument_APIs.ipynb>

* Super clear
* Exchange rates! <https://github.com/nealcaren/UiOBigData/blob/master/notebooks/04_APIs.ipynb>

def search\_itunes(search\_term):

'''Simplified iTunes search'''

base\_url = 'https://itunes.apple.com/search'

search\_parameters = {'term': search\_term}

r = requests.get(base\_url, params = search\_parameters)

results\_df = pd.DataFrame(r.json()['results'])

return results\_df

search\_itunes('billie eilish')

Or Spotify?

Replicating dataset (to a degree) of Lazy Prices: <https://www.quantopian.com/posts/scraping-10-ks-and-10-qs-for-alpha> (includes some 10-K parsing)

10-k text cleaning / general scraping pointers

https://towardsdatascience.com/useful-sentiment-analysis-mining-sec-filings-part-1-358942fc98ed

<https://sraf.nd.edu/data/stage-one-10-x-parse-data/>

<https://sraf.nd.edu/data/>