

[WRITE A POST](#)

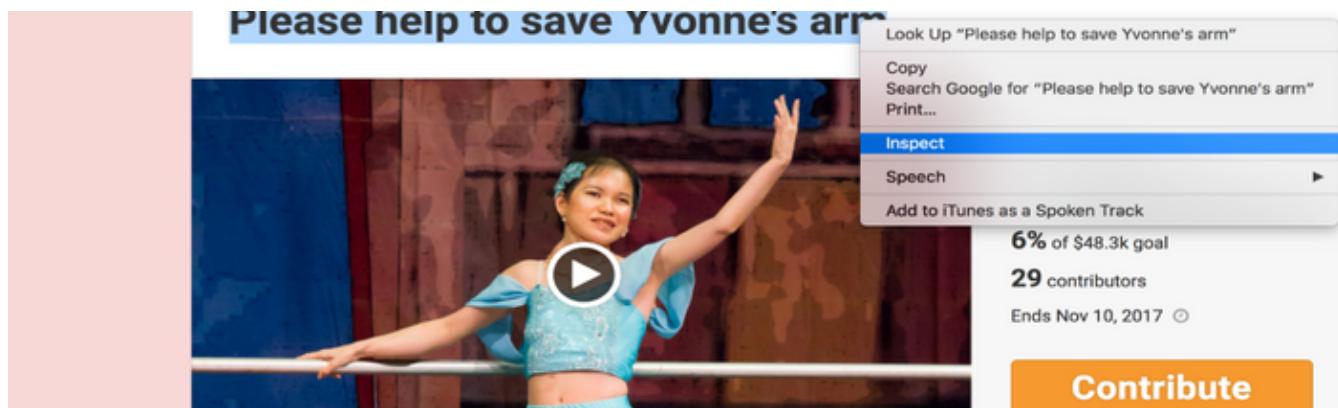
Michael

[FOLLOW](#)

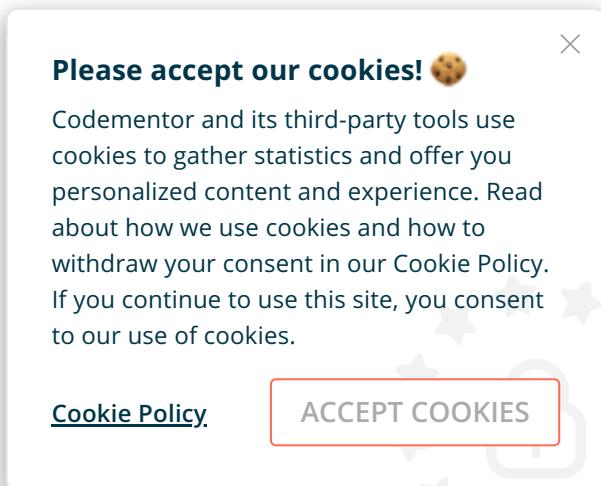
Data Scientist, Blogger, and Youtuber

## Web Scraping in Python using Scrapy

Published Oct 18, 2017



When I first started working in industry, one of the things I quickly realized is sometimes you have to gather, organize, and clean your own data. For this tutorial, we will gather data from a crowdfunding website called [FundRazr](#). Like many websites, the site has its own structure, form, and has tons of accessible useful data, but it is hard to get data from the site as it doesn't have a structured API. In result, we will web scrape the site to get that unstructured website data and put into an ordered form to build our own dataset.



In order to scrape the website, we will use [Scrapy](#). In short, Scrapy is a framework built to build web scrapers more easily and relieve the pain of maintaining them. Basically, it allows you to focus on the data extraction using CSS selectors and choosing XPath expressions and less on the intricate internals of how spiders are supposed to work. This blog post goes a little beyond the great [official tutorial from the scrapy documentation](#) in the hopes that if you need to scrape something a bit harder, you can do it on your own. With that, lets get started. If you get lost, I recommend opening the [video](#) above in a separate tab.

## Getting Started (Prerequisites)

If you already have anaconda and google chrome (or Firefox), skip to Creating a New Scrapy Project.

1. Install Anaconda (Python) on your operating system. You can either download anaconda from the official site and install on your own or you can follow these anaconda installation tutorials below.

Operating System	Blog Post	Youtube Video
Mac	<a href="#">Install Anaconda on Mac</a>	<a href="#">Youtube Video</a>
Windows	<a href="#">Install Anaconda on Windows</a>	<a href="#">Youtube Video</a>
Ubuntu	<a href="#">Install Anaconda on Ubuntu</a>	<a href="#">Youtube Video</a>
All	<a href="#">Environment Management with Conda (Python 2 + 3, Configuring Jupyter Notebooks)</a>	<a href="#">Youtube Video</a>

2. Install Scrapy (anaconda comes with it, but just in case). You can also install on your terminal (mac/linux) or command line (windows). You can type the following:

```
conda install -c conda-forge scrapy
```

3. Make sure you have Google chrome or Firefox. In this tutorial, I am using Google Chrome. If you don't have google chrome, you can install it here using this [link](#).

## Creating a New Scrapy Project

1. Open a terminal (mac/linux) or command line (windows). Navigate to a desired folder (see the image below if you need help) and type

```
scrapy startproject fundrazr
```

```
Michaels-MacBook-Pro-3:~ mgalarny$ cd Desktop
Michaels-MacBook-Pro-3:Desktop mgalarny$ cd Python_Tutorials/
Michaels-MacBook-Pro-3:Python_Tutorials mgalarny$ cd Scrapy/
Michaels-MacBook-Pro-3:Scrapy mgalarny$ scrapy startproject fundrazr
New Scrapy project 'fundrazr', using template directory '/Users/mgalarny/anaconda2/lib/python2.7/site-packages/scrapy/templates/project', created in:
/Users/mgalarny/Desktop/Python_Tutorials/Scrapy/fundrazr

You can start your first spider with:
  cd fundrazr
  scrapy genspider example example.com
```

*scrapy startproject fundrazr*

This makes a fundrazr directory with the following contents:

```
fundrazr/
  scrapy.cfg          # deploy configuration file

  fundrazr/
    __init__.py        # project's Python module, you'll import your code from here

    items.py           # project items definition file (we will use this)

    pipelines.py       # project pipelines file

    settings.py        # project settings file

    spiders/
      __init__.py      # a directory where you'll later put your spiders (we will use this)
```

*fundrazr project directory*

## Finding Good Start URLs using Inspect on Google Chrome (or Firefox)

In the spider framework, `start_urls` is a list of URLs where the spider will begin to crawl from, when no particular URLs are specified. We will use each element in the `start_urls` list as a means to get individual campaign links.

1. The image below shows that based on the category you choose, you get a different start url. The highlighted part in black are the possible categories of fundrazrs to scrape.

The image displays two side-by-side screenshots of the Fundrazr website. The left screenshot shows the 'Accidents & Disasters' category, with a highlighted section of the sidebar. The right screenshot shows the 'Health' category, also with a highlighted section of the sidebar. Both pages list several fundraising projects, each with a title, a brief description, the amount raised, the number of days left, and a small thumbnail image.

*Finding a good first start\_url*

For this tutorial, the first in the list `start_urls` is:

<https://fundrazr.com/find?category=Health>

2. This part is about getting additional elements to put in the `start_urls` list. We are finding out how to go to the next page so we can get additional urls to put in `start_urls`.

*Getting additional elements to put in `start_urls` list by inspecting Next button*

The second start url is: <https://fundrazer.com/find?category=Health&page=2>

The code below will be used in the code for the spider later in the tutorial. All it does is make a list of `start_urls`. The variable `npages` is just how many additional pages (after the first page) we want to get campaign links from.

```
start_urls = [ "https://fundrazr.com/find?category=Health"]

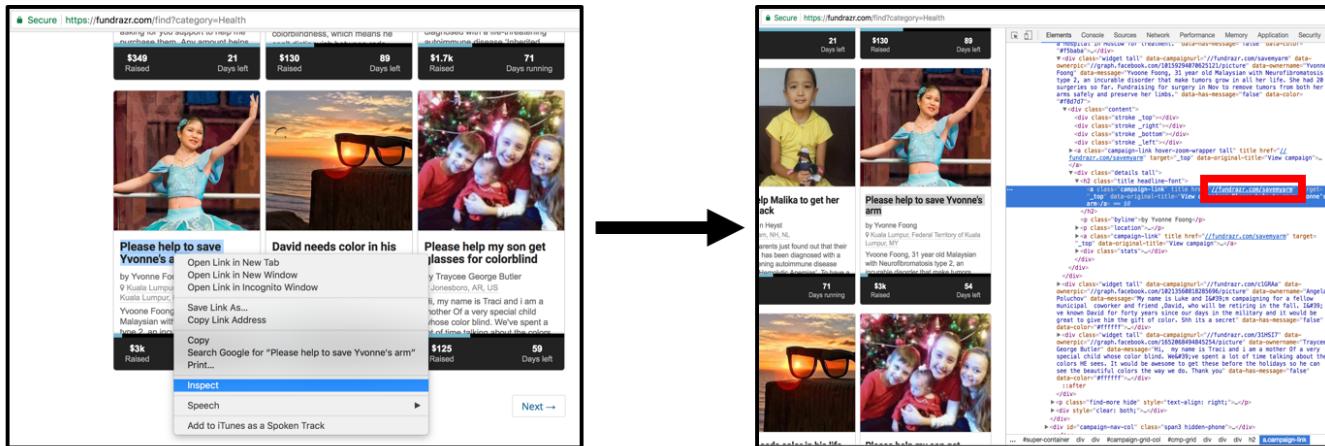
npages = 2

# This mimics getting the pages using the next button.
for i in range(2, npages + 2):
    start_urls.append("https://fundrazr.com/find?category=Health&page=" + str(i))
```

## Scrapy Shell for finding Individual Campaign Links

The best way to learn how to extract data with Scrapy is using the Scrapy shell. We will use XPaths which can be used to select elements from HTML documents.

The first thing we will try and get the xpaths for are the individual campaign links. First we do inspect to see roughly where the campaigns are in the HTML.



Right click on the title for the first campaign you see and then click on Inspect

The text (enclosed within the red rectangle) is part of a url for an individual campaign.

*Finding links to individual campaigns*

We will use XPath to extract the part enclosed in the red rectangle below.

```
<a class="campaign-link" title href="//fundrazr.com/savemyarm" target="_top" data-original-title="View campaign" style="color: inherit; text-decoration: none;">Yvonne's arm</a> == $0
```

*The enclosed part is a partial url we will isolate*

In terminal type (mac/linux):

```
scrapy shell 'https://fundrazr.com/find?category=Health'
```

In command line type (windows):

```
scrapy shell "https://fundrazr.com/find?category=Health"
```

Type the following into scrapy shell (to help understand the code, please see the video):

```
response.xpath("//h2[contains(@class, 'title headline-font')]/a[contains(
```

```
[In [9]: response.xpath("//h2[contains(@class, 'title headline-font')]/a[contains(@class, 'campaign-link')]/@href").extract()
Out[9]:
[u'//fundrazr.com/savemyarm',
 u'//fundrazr.com/exjqd',
 u'//fund.ridetogive.com/e1HGb1',
 u'//fundrazr.com/b1G182',
 u'//fundrazr.com/f1HGf5',
 u'//fundrazr.com/americangut',
 u'//fundrazr.com/e1HLzb',
 u'//fundrazr.com/britishgut',
 u'//fundrazr.com/611LCf',
 u'//fundrazr.com/f17Vw9',
 u'//fundrazr.com/matthewsfight',
 u'//fundrazr.com/f1GKB7']
```

*There is a good chance you will get different partial urls as websites update over time*

The code below is for getting all the campaign links for a given start url (more on this later in the First Spider section)

```
for href in response.xpath("//h2[contains(@class, 'title headline-font')]")
    # add the scheme, eg http://
    url = "https://" + href.extract()
```

Exit Scrapy Shell by typing `exit()`.

[In [14]: `exit()`

Michaels-MacBook-Pro-3:spiders mgalarny\$ █

*exit scrapy shell*

## Inspecting Individual Campaigns

While we should previously worked on understanding the structure of where individual campaigns links are, this section goes over where things are on individual campaigns.

1. Next we go to an individual campaign page (see link below) to scrape (I should note that some of these campaigns are difficult to view)

<https://fundrazr.com/savemyarm>

2. Using the same inspect process as before, we inspect the title on the page



The screenshot shows a FundRazr campaign page. At the top, there's a navigation bar with the FundRazr logo, a 'Like 124K' button, a search bar, and links for 'Start', 'How it works', and 'Sign in'. Below the navigation is a pink header with the title 'Please help to save Yvonne's arm'. The main content area features a photo of a woman in a blue leotard dancing, with a play button overlaid. To the right of the photo is a sidebar with options like 'Look Up "Please help to save Yvonne's arm"', 'Copy', 'Search Google for "Please help to save Yvonne's arm"', 'Print...', 'Inspect' (which is highlighted in blue), 'Speech', 'Add to iTunes as a Spoken Track', and campaign statistics: '6% of \$48.3k goal', '29 contributors', and 'Ends Nov 10, 2017'. A large orange 'Contribute' button is at the bottom right.

*Inspect Campaign Title*

3. Now we are going to use scrapy shell again, but this time with an individual campaign. We do this because we want to find out how individual campaigns are formatted (including finding out how to extract the title from the webpage).

In terminal type (mac/linux):

```
scrapy shell 'https://fundrazr.com/savemyarm'
```

In command line type (windows):

```
scrapy shell "https://fundrazr.com/savemyarm"
```

The code to get the campaign title is

```
response.xpath("//div[contains(@id, 'campaign-title')]/descendant::text()")
```

```
[In 4]: response.xpath("//div[contains(@id, 'campaign-title')]/descendant::text()").extract()[0]
Out[4]: u"\n\t\t\t\t\t\t\tPlease help to save Yvonne's arm\n\t\t\t\t\t\t\t"
```

4. We can do the same for the other parts of the page.

amount Raised:

```
response.xpath("//span[contains(@class, 'stat')]/span[contains(@class, 'am')]")
```

goal:

```
response.xpath("//div[contains(@class, 'stats-primary with-goal')]/span[1]")
```

currency type:

```
response.xpath("//div[contains(@class, 'stats-primary with-goal')]/@title")
```

campaign end date:

```
response.xpath("//div[contains(@id, 'campaign-stats')]/span[contains(@cl')]")
```

number of contributors:

```
response.xpath("//div[contains(@class, 'stats-secondary with-goal')]/span[1]")
```

story:

```
response.xpath("//div[contains(@id, 'full-story')]/descendant::text()").e
```

url:

```
response.xpath("//meta[@property='og:url']/@content").extract()
```

5. Exit scrapy shell by typing:

```
exit()
```

## Items

The main goal in scraping is to extract structured data from unstructured sources, typically, web pages. Scrapy spiders can return the extracted data as Python dicts. While convenient and familiar, Python dicts lack structure: it is easy to make a typo in a field name or return inconsistent data, especially in a larger project with many spiders (almost word for word copied from the great scrapy official documentation!).

```
fundrazer/
    scrapy.cfg          # deploy configuration file
    fundrazer/
        __init__.py      # project's Python module, you'll import your code from here
        items.py          # project items definition file (we will use this)
        pipelines.py      # project pipelines file
        settings.py       # project settings file
    spiders/
        __init__.py      # a directory where you'll later put your spiders (we will use this)
```

*File we will be modifying*

The code for items.py is [here](#).

Save it under the fundrazr/fundrazr directory (overwrite the original items.py file).

The item class (basically how we store our data before outputting it) used in this tutorial looks like this.

```
import scrapy

class FundrazrItem(scrapy.Item):
    campaignTitle = scrapy.Field()
    amountRaised = scrapy.Field()
    goal = scrapy.Field()
    currencyType = scrapy.Field()
    endDate = scrapy.Field()
    numberContributors = scrapy.Field()
    story = scrapy.Field()
    url = scrapy.Field()
```

*items.py code*

## The Spider

Spiders are classes that you define and that Scrapy uses to scrape information from a website (or a group of websites). The code for our spider is below.

```

import scrapy
from fundrazr.items import FundrazrItem
from datetime import datetime
import re

class Fundrazr(scrapy.Spider):
    name = "my_scraper"

    # First Start Url
    start_urls = ['https://fundrazr.com/find?category=Health']

    npages = 2

    # This mimics getting the pages using the next button.
    for i in range(2, npages + 2):
        start_urls.append('https://fundrazr.com/find?category=Health&page=' + str(i) + '')

    def parse(self, response):
        for href in response.xpath("//h2[contains(@class, 'title headline-font')]/a[contains(@class, 'campaign-link')]//@href"):
            # add the scheme, eg http://
            url = "https://" + href.extract()
            yield scrapy.Request(url, callback=self.parse_dir_contents)

    def parse_dir_contents(self, response):
        item = FundrazrItem()

        # Getting Campaign Title
        item['campaignTitle'] = response.xpath("//div[contains(@id, 'campaign-title')]/descendant::text()").extract()[0].strip()

        # Getting Amount Raised
        item['amountRaised'] = response.xpath("//span[contains(@class, 'stat')]/span[contains(@class, 'amount-raised')]/descendant::text()").extract()

        # Goal
        item['goal'] = ".join(response.xpath("//div[contains(@class, 'stats-primary with-goal')]/span[contains(@class, 'stats-label hidden-phone')]/text()").extract()).strip()

        # Currency Type (US Dollar Etc)
        item['currencyType'] = response.xpath("//div[contains(@class, 'stats-primary with-goal')]/@title").extract()

        # Campaign End (Month year etc)
        item['endDate'] = "".join(response.xpath("//div[contains(@id, 'campaign-stats')]/span[contains(@class, 'stats-label hidden-phone')]/span[@class='nowrap']/text()").extract()).strip()

        # Number of contributors
        item['numberContributors'] = response.xpath("//div[contains(@class, 'stats-secondary with-goal')]/span[contains(@class, 'donation-count stat')]/text()").extract()

        # Getting Story
        story_list = response.xpath("//div[contains(@id, 'full-story')]/descendant::text()").extract()
        story_list = [x.strip() for x in story_list if len(x.strip()) > 0]
        item['story'] = ".join(story_list)"

        # Url (The link to the page)
        item['url'] = response.xpath("//meta[@property='og:url']/@content").extract()

        yield item

```

### Fundrazr Scrapy Code

Download the code [here](#).

Save it in a file named **fundrazr\_scrape.py** under the fundrazr/spiders directory.

The current project should now have the following contents:

```

fundrazr/
scrapy.cfg          # deploy configuration file

fundrazr/
    __init__.py      # project's Python module, you'll import your code from here

    items.py         # project items definition file (we will use this)

    pipelines.py     # project pipelines file

    settings.py      # project settings file

spiders/
    __init__.py      # a directory where you'll later put your spiders (we will use this)

    fundrazr_scrape.py # We make this file and add in code for our first spider

```

*File we will be creating/adding*

## Running the Spider

1. Go to the fundrazr/fundrazr directory and type:

```
scrapy crawl my_scraper -o MonthDay_Year.csv
```

```
scrapy crawl my_scraper -o MonthDay_Year.csv
```

2. The data should be outputted in the fundrazr/fundrazr directory.

 <code>__init__.py</code>	Mar 10, 2017, 7:31 AM	Zero bytes
 <code>__init__.pyc</code>	Yesterday, 5:35 PM	140 bytes
 <code>items.py</code>	Yesterday, 6:16 PM	426 bytes
 <code>items.pyc</code>	Yesterday, 6:19 PM	616 bytes
 <code>middlewares.py</code>	Yesterday, 5:26 PM	2 KB
 <code>MonthDay_Year.csv</code>	Yesterday, 6:19 PM	57 KB
 <code>pipelines.py</code>	Yesterday, 5:26 PM	288 bytes
 <code>settings.py</code>	Yesterday, 5:26 PM	3 KB
 <code>settings.pyc</code>	Yesterday, 5:35 PM	288 bytes
 <code>spiders</code>	Yesterday, 6:19 PM	--

*Data Output Location*

## Our Data

1. The data outputted in this tutorial should look roughly like the image below. The individual campaigns scraped will vary as the website is constantly updated. Also it is possible there will be spaces between each individual campaign as excel is interpreting the csv file.

*The data should roughly be in this format.*

2. If you want to download a larger file (it was made by changing npages = 2 to npages = 450 and adding download\_delay = 2), you can download a bigger file with roughly 6000 campaigns scraped by downloading the file from my [github](#). The file is called MiniMorningScrape.csv (it is a large file).

story	amountRaised	goal	url	campaignTitle	numberContributors	endDate	currencyType
On Sunday, May	9,146	\$9k goal	<a href="https://fund.ridetogive.com/a1EsW6">https://fund.ridetogive.com/a1EsW6</a>	Ride to Give - Supporting Cezar's Cancer Journey	444		U.S. Dollar
The past month has	54,205	\$200k goal	<a href="https://fundrazer.com/f17Vw9">https://fundrazer.com/f17Vw9</a>	Jasper's fund	530		U.S. Dollar
Hi There! We're	105	\$50k goal	<a href="https://fundrazer.com/41GoD5">https://fundrazer.com/41GoD5</a>	Justin's Journey	3	13-Nov-17	U.S. Dollar
Since the summer	5,720		<a href="https://fundrazer.com/fbhX6">https://fundrazer.com/fbhX6</a>	Ancestral Microbiome: Chasing Health			
Hejsan! Jag heter	430	of kr3.5k goal	<a href="https://fundrazer.com/e1Gns5">https://fundrazer.com/e1Gns5</a>	Hjälp mig att se färger!	4		Swedish Krona
Our story begins	120	of \$15k goal	<a href="https://fundrazer.com/61bkNe">https://fundrazer.com/61bkNe</a>	Baby Blessings- We wont give up	2		U.S. Dollar
I'm raising money	6,826	of Å€4k goal	<a href="https://fundrazer.com/a181G4">https://fundrazer.com/a181G4</a>	Help Russell to speak again	197		Pound Sterling
WHAT IS	375	\$500 goal	<a href="https://fundrazer.com/colorforsteveknouse">https://fundrazer.com/colorforsteveknouse</a>	Help Steve Knouse See the World in Color	10		U.S. Dollar
This is Jake. Å He's	525	\$450 goal	<a href="https://fundrazer.com/0161Eb">https://fundrazer.com/0161Eb</a>	Enchroma glasses, for a full color life experien	2		U.S. Dollar
	1,900	of \$1.9k goal	<a href="https://fundrazer.com/61FQg6">https://fundrazer.com/61FQg6</a>	Please help Kristan pay for dental surgery!	7		U.S. Dollar
This year we are	100	of \$10k goal	<a href="https://fundrazer.com/MedicalCannabis">https://fundrazer.com/MedicalCannabis</a>	Cannabis Therapeutics Certification	5		U.S. Dollar
Written diligently	2,655	of \$11.3k goal	<a href="https://fundrazer.com/01ejG6">https://fundrazer.com/01ejG6</a>	Help for Howard	21		U.S. Dollar
Jhunnel Sarajan, a	236	\$300 goal	<a href="https://fundrazer.com/31e253">https://fundrazer.com/31e253</a>	Please help a Selfless Father pay his medical bills	15		Canadian Dollar
At only 32 year of	355		<a href="https://fundrazer.com/fq03d">https://fundrazer.com/fq03d</a>	Nikki Michaella's Fight for Life Fund			
***UPDATE*** MY	1,350	of \$10k goal	<a href="https://fundrazer.com/tg1lqueensland">https://fundrazer.com/tg1lqueensland</a>	Botched. Please donate..Every bit helps.	23		Australian Dollar
Le 23 janvier 2012,	2,340	of \$10k goal	<a href="https://fundrazer.com/11ANEf">https://fundrazer.com/11ANEf</a>	Soutenez Anne	30		Canadian Dollar
Hello. We are	1,125	of \$60k goal	<a href="https://fundrazer.com/cIMqd">https://fundrazer.com/cIMqd</a>	Help Zion continue her fight against Leukemia	21		U.S. Dollar
I am Hebert	1,000	of \$12k goal	<a href="https://fundrazer.com/f1CLY2">https://fundrazer.com/f1CLY2</a>	Please Help us get new legs for Pride	11		U.S. Dollar
Anyone who's	6,599	of \$20k goal	<a href="https://fundrazer.com/71Aek5">https://fundrazer.com/71Aek5</a>	Please help us pay for Nicola's surgery costs	91		U.S. Dollar
	602	of Å€800 goal	<a href="https://fundrazer.com/31CnF4">https://fundrazer.com/31CnF4</a>	Cancer	4		Euro
On February 5,	11,000	of Å€11.5M goal	<a href="https://fundrazer.com/11ct78">https://fundrazer.com/11ct78</a>	A Helping Hand for Auntie Arnieâ€™s Medica	5		Philippine Peso
Please donate to giv	100	of Å€10k goal	<a href="https://fundrazer.com/71Cbdc">https://fundrazer.com/71Cbdc</a>	Help me survive with bipolar and have a brea	2		Pound Sterling
Reverend Larry has	515	of \$5k goal	<a href="https://fundrazer.com/31clle">https://fundrazer.com/31clle</a>	Help Rev. Larry Walk After Lightning Strike	13		U.S. Dollar
I have a rare	4,978	of \$10k goal	<a href="https://fundrazer.com/7qgTa">https://fundrazer.com/7qgTa</a>	Save Me from Stiff Person Syndrome	43		U.S. Dollar
My name is Donna	655	of Å€1k goal	<a href="https://fundrazer.com/youthmindfulnesstraini">https://fundrazer.com/youthmindfulnesstraini</a>	Youth Mindfulness Training	19		Pound Sterling
My name is Harley	18,423	of \$50k goal	<a href="https://fundrazer.com/e19t54">https://fundrazer.com/e19t54</a>	Katsumi Edamatsu	123		U.S. Dollar
Hi all 10 years ago,	3,861	of \$5k goal	<a href="https://www.realp2p.com/619st6">https://www.realp2p.com/619st6</a>	Miles for Myles	39		U.S. Dollar
ĐœĐ½Đ³Nf Đ»Đ,	290	of \$500 goal	<a href="https://fundrazer.com/b189Y0">https://fundrazer.com/b189Y0</a>	I need help	9		U.S. Dollar
Hi my name is	135		<a href="https://fundrazer.com/a1axye">https://fundrazer.com/a1axye</a>	Help Patty With Double Lung Transplant Expenses			
A few years ago I	451	of Å€1k goal	<a href="https://fundrazer.com/d16mSe">https://fundrazer.com/d16mSe</a>	Helping Freddie & Louie battle Battens Disease	27		Pound Sterling
Jordan Lopez	1,810	of \$10k goal	<a href="https://fundrazer.com/d1Dl51">https://fundrazer.com/d1Dl51</a>	Jordan's medical expenses (Heart)	9		U.S. Dollar
Hello my name is	320		<a href="https://fundrazer.com/TeresaTeal">https://fundrazer.com/TeresaTeal</a>	Please help Teresa Teal beat Ovarian Cancer			
We are a group of	1,402	of \$1.5k goal	<a href="https://fundrazer.com/AbortionSupportTucson">https://fundrazer.com/AbortionSupportTucson</a>	Help fund abortion support training in Tucson	37		U.S. Dollar
In January of 2016	835	of \$10k goal	<a href="https://fundrazer.com/81CeFc">https://fundrazer.com/81CeFc</a>	Bevon colon cancer fundraising	12	31-Jan-18	U.S. Dollar
It is our honor to be	2,550	of \$50k goal	<a href="https://fundrazer.com/71DCU8">https://fundrazer.com/71DCU8</a>	Love Never Falls	4		Mexican Peso
My name is Nancy	225	of \$4k goal	<a href="https://fundrazer.com/a1ctue">https://fundrazer.com/a1ctue</a>	ADULT SIBLING ABUSE HOTLINE	7		U.S. Dollar
Welcome to my	1,141	of Å€2k goal	<a href="https://fundrazer.com/91AjBd">https://fundrazer.com/91AjBd</a>	Rachel's Cancer Battle - A chance of a longer life	61	1-Nov-17	Pound Sterling
Iâ€™ve been	1,835	of \$10k goal	<a href="https://fundrazer.com/dquintreurntothewalk">https://fundrazer.com/dquintreurntothewalk</a>	Help Duane "DJ DRD" Quintanilla Walk Again	26		U.S. Dollar
My name is George	435	of \$500 goal	<a href="https://fund.ridetogive.com/c1COmc">https://fund.ridetogive.com/c1COmc</a>	Team Ride to Give 2017	8		U.S. Dollar

Roughly 6000 Campaigns Scraped

## Closing Thoughts

Creating a dataset can be a lot of work and is often an overlooked part of learning data science. One thing we didn't go over is that while we scraped a lot of data, we still haven't cleaned the data enough to do analysis. That is for another blog post though. Please let me know if you have any questions either here or on the [youtube video page](#)!

This article originally appeared on my [medium blog](#)

Python

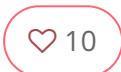
Data science in python

Python web scraping

Data Science

Web scraping

Enjoy this post? Give **Michael** a like if it's helpful.



**Michael**

Data Scientist, Blogger, and Youtuber

[FOLLOW](#) **1 Reply**

Leave a reply

**Russell Winterbotham** 9 months ago

▼

Can you help me set up a scraping program to extract names and scores from Leaderboard tables on a Fantasy Sports site?

Reply

# Hire the world's top freelance software developers

[HIRE A DEVELOPER](#)[APPLY AS A DEVELOPER](#)



Edaqa Mortoray

## We don't need a ternary operator



A staple of compact code, the ternary operator `?:` feels like it's earned a place in any programming language. I use it often — you should use it often. However, I don't like the operator itself. It feels complex and incomplete. Though [I've stopped development on Leaf](#), I did find a better option.

Let's take a deeper look at what this ternary operator is, then show how a language could avoid it.

[READ MORE](#)

Join and start **discussions** with fellow developers

START A DISCUSSION