



# CodeFirstGirls Beginners Python: Course Projects

Two types of project are available, API Based projects and data analysis projects.

## API Based projects

- Top Trumps Game
- Cryptocurrency Converter
- Recipe Generator

## Data analysis projects

- IMDb movie data
- London Resolved Crime 2019 data
- San Francisco Crime data
- University Rankings data
- World Food Production 2018 data
- World Happiness data

# API Projects

## TopTrumps game

In this project you'll create a small game where players compare stats, similar to the Top Trumps card game. The basic flow of the games is:

1. You are given a random card with different stats
2. You select one of the card's stats
3. Another random card is selected for your opponent (the computer)
4. The stats of the two cards are compared
5. The player with the stat higher than their opponent wins.

The standard project will use the Pokemon API, but you can use a different API if you want after completing the required tasks.

You will not need any additional knowledge beyond what is covered in this course to complete this project.

### Required Tasks

These are the required tasks for this project. You should aim to complete these tasks before adding your own ideas to the project.

1. Generate a random number between 1 and 151 to use as the Pokemon ID number
2. Using the Pokemon API get a Pokemon based on its ID number
3. Create a dictionary that contains the returned Pokemon's name, id, height and weight (<https://pokeapi.co/>)
4. Get a random Pokemon for the player and another for their opponent
5. Ask the user which stat they want to use (id, height or weight)
6. Compare the player's and opponent's Pokemon on the chosen stat to decide who wins

### Ideas for Extending the Project

Here are a few ideas for extending the project beyond the required tasks. These ideas are just suggestions, feel free to come up with your own ideas and extend the program however you want.

- Use different stats for the Pokemon from the API
- Get multiple random Pokemon and let the player decide which one that they want to use
- Play multiple rounds and record the outcome of each round. The player with most number of rounds won, wins the game
- Allow the opponent (computer) to choose a stat that they would like to compare
- Record high scores for players and store them in a file
- Use a different API (see suggestions below)

### Useful Resources

Star Wars API

- Homepage <https://swapi.co/>
- Documentation <https://swapi.co/documentation>

#### Anime/Manga API

- Homepage <https://jikan.moe/>
- Documentation <https://jikan.docs.apiary.io>
- Example API url <https://api.jikan.moe/v3/anime/5>

#### Example Project Code

In this section you will find some example code to complete the required tasks. You can use this code for guidance if you are finding it difficult to complete the required tasks for this project.

```
import random
import requests

def random_pokemon ():
    pokemon_number = random.randint(1, 151)
    url = 'https://pokeapi.co/api/v2/pokemon/{}/'.format(pokemon_number)
    response = requests.get(url)
    pokemon = response.json()
    return {
        'name': pokemon['name'],
        'id': pokemon['id'],
        'height': pokemon['height'],
        'weight': pokemon['weight'],
    }

def run ():
    my_pokemon = random_pokemon()
    print('You were given {}'.format(my_pokemon['name']))
    stat_choice = input('Which stat do you want to use? (id, height, weight) ')
    opponent_pokemon = random_pokemon()
    print('The opponent chose {}'.format(opponent_pokemon['name']))
    my_stat = my_pokemon[stat_choice]
    opponent_stat = opponent_pokemon[stat_choice]
    if my_stat > opponent_stat:
        print('You Win!')
    elif my_stat < opponent_stat:
        print('You Lose!')
    else :
        print('Draw!')

run()
```

## Cryptocurrency converter

In this project you will investigate different cryptocurrencies (such as Bitcoin, Litecoin, and many more). There is an option to analyse data on the coins (such as current prices, total numbers of coins circulation) and/or create a cryptocurrency price converter.

### Project setup:

1. Sign up for an API key here: <https://coinmarketcap.com/api/> Please select the basic account which is free.
2. Add the API key and ID to your PyCharm Environment variables. See Week 6 slides for instructions.

### Meaning of API data categories:

- "cmc\_rank": Rank of coin values
- "max\_supply": Maximum possible number of coins (circulating and non-circulating)
- "name": name of the coin
- "symbol": symbol of the coin eg. BTC
- "market\_cap": Total value of circulating coins in USD
- "percent\_change\_1h": Percentage price change in last hour
- "percent\_change\_24h": Percentage price change in last 24 hours
- "percent\_change\_7d": Percentage price change in last 7 days
- "price": Current price
- "volume\_24h": Total value of coins exchanged in 24 hours

### Some initial ideas for data analysis:

- Return the names of the top 10 cryptocurrencies for price, volume and market cap.
- Create a crypto price to USD converter. Ask the user for the name of the cryptocurrency and how many coins they have. Write a function to find this cryptocurrencies price in USD, and return the total value of the user's coins in USD.

### Example code:

```

from pprint import pprint
from requests import Session
import json
import os

api_key = os.environ.get("crypto_api_key")

url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest'
parameters = {
    'start': '1',
    'limit': '2000', # number of returned coins from the request
    'convert': 'USD' # currency the value is returned in
}
headers = {
    'Accepts': 'application/json',
    'X-CMC_PRO_API_KEY': api_key,
}

session = Session()
session.headers.update(headers)

response = session.get(url, params=parameters)
crypto_dict = json.loads(response.text)['data']
pprint(crypto_dict)

# print the names of all the coins and their value
for coin in crypto_dict:
    coin_price_usd = coin['quote']['USD']['price'] # get the value in USD
    print('{} is worth ${:.2f}'.format(coin['name'], coin_price_usd)) # print the value in USD

```

# Recipe Generator

In this project you'll create a program to search for recipes based on an ingredient. The standard project uses the Edamam API, but can be changed to use a different API after completing the required tasks.

You will not need any additional knowledge beyond what is covered in this course to complete this project.

## Project set up

1. Read the Edamam API documentation <https://developer.edamam.com/edamam-docs-recipe-api>
2. Sign up for an API key and API ID. Select 'Recipe Search' API- Developer.
3. Add the API key and ID to your PyCharm Environment variables. See Week 6 slides for instructions.

## Required Tasks

These are the required tasks for this project. You should aim to complete these tasks before adding your own ideas to the project.

1. Ask the user to enter an ingredient that they want to search for
2. Create a function that makes a request to the Edamam API with the required ingredient as part of the search query
3. Get the returned recipes from the API response
4. Display the recipes for each search result

## Ideas for Extending the Project

Here are a few ideas for extending the project beyond the required tasks. These ideas are just suggestions, feel free to come up with your own ideas and extend the program however you want.

- Save the results to a file
- Filter the recipes based on rating
- Ask the user additional questions to decide which recipe they should choose, for example, dietary requirements

## Example Code

In this section you will find some example code to complete the required tasks. You can use this code for guidance if you are finding it difficult to complete the required tasks for this project.

```
import requests
import os

# Register to get an API key https://developer.edamam.com/edamam-recipe-api
api_id = os.environ.get("edamam_api_id")
api_key = os.environ.get("edamam_api_key")

def recipe_search (ingredient):
    result = requests.get('https://api.edamam.com/search?
        q={}&app_id={}&app_key={}'.format(ingredient, api_id, api_key))
    data = result.json()
    return data['hits']

def run ():
    ingredient = input('Enter an ingredient: ')
    recipes = recipe_search(ingredient)
    for recipe in recipes:
        print(recipe['recipe']['label'])
        print(recipe['recipe']['url'])
        print()

run()
```



# Data analysis projects

## IMDb movie data

This project will investigate movie ranking data from English language movies.

More info: <https://www.kaggle.com/saipranava/top-ranked-enlglsh-movies-of-this-decade>

### Project setup:

Read the .csv file in python. Use example code below to help you. The .csv data file can be found in the CFG datasets folder on google drive.

### Analysis ideas:

- Print/ plot the top 10 movies by ranking.
- Plot the biggest grossing/ revenue/ budget movies
- Plot the revenue data by Domestic/ Foreign/ Worldwide
- Plot the top 10 movies for each genre

### Example code:

```
import csv

with open('IMDB_data.csv', 'r') as csv_file:
    movie_data = csv.DictReader(csv_file)

    headers = movie_data.fieldnames
    print(headers)

    for movie in movie_data:
        title = movie['Title']
        rating = movie['Rating']
        print('{}has a rating of {} on IMDb'.format(title, rating))
```

## London Crime data (December 2019)

This project will investigate resolved crimes in London during December 2019.

## Project setup:

Read the .csv file in python. Use example code below to help you. The .csv data file can be found in the CFG datasets folder on google drive.

### Analysis ideas:

- What are the most common and uncommon crime types?
- Plot the number of each crime type.
- What are the top crimes in each region? Plot the number of crimes by region.
- How is each crime type resolved? Plot the resolution from each crime type.

Example code:

[illegible]

## San Francisco Crimes

This project will investigate resolved crimes over one year in San Francisco.

## Project setup:

Read the .csv file in python. Use example code below to help you. The .csv data file can be found in the CFG datasets folder on google drive.

## Analysis ideas:

- What are the most common and uncommon crime types? Plot the number of each crime type.
- What are the top crimes in each District? Plot the number of crimes by region.
- How is each crime type resolved? Plot the resolution from each crime type.
- Which days of the week are crimes most and least likely to occur?
- Which times of the day are crimes most and least likely to occur?

Example code:

[illegible]

## QS world university rankings data

This project will investigate university world ranking data from 2019/2020.

QS world rankings: <https://www.topuniversities.com/university-rankings/world-university-rankings/2020>

### Project setup:

Read the .csv file in python. Use example code below to help you. The .csv data file can be found in the CFG datasets folder on google drive. This file contains ranking data based on several metrics/ indicators for universities around the world.

Information on what each of the ranking indicators mean can be found here:

Classifications: <http://www.iu.qs.com/university-rankings/qs-classifications/>

Rankings: <http://www.iu.qs.com/university-rankings/>

Go to the 'rankings' in the menu, then select the indicator you are interested in knowing more about.

### Analysis ideas:

- Print the top 10 universities in the world (from ranking) for 2020 and 2019
- Plot the number of universities in each country in the top 10/ 100
- Plot the top 10 universities of each size (eg. S, M, L, XL)
- Plot the top scores for each indicator with the name of the university (eg. bar plot).

### Example code

```
import csv

with open('university_rankings.csv', 'r') as csv_file:
    uni_data = csv.DictReader(csv_file)

    headers = uni_data.fieldnames
    print(headers)

    for university in uni_data:
        institution = university['institution']
        rank = university['rank_2020']
        country = university['country']
        print('Institution: {}, Rank: {}, Country: {}'.format(institution, rank, country))
```

## World Food Production

In this project you will analyse food production from around the world for 2018. The .csv file contains the amount of produced food of different types for each country (in tonnes).

This data is from the UN World Food Production dataset:

<http://www.fao.org/faostat/en/#home>

### Project setup:

Read the .csv file in python. Use example code below to help you. The .csv data file can be found in the CFG datasets folder on google drive.

### Analysis ideas:

- Print/ plot the top 10 food producers in the world.
- Plot the top food producers for each food type
- Create a program that allows users to search for a country and food type, then return the amount produced by that country
- Plot the top food producers by region (eg. Asia, Europe)

### Example code:

```
import csv

with open('world_food_production_2018.csv', 'r') as csv_file:
    food_data = csv.DictReader(csv_file)

    headers = food_data.fieldnames
    print(headers)

    for country in food_data:
        country_name = country['Country']
        region = country['Area']
        citrus = country['Citrus_fruit']
        print('In 2018 {} in {} produced {} tonnes of citrus.'.format(country_name,
                                region, citrus))
```

## World Happiness data 2015 & 2016

In this project you will analyse happiness scores for countries in 2015 and 2016. There are two .csv files, each contains the happiness score for either 2015/ 2016.

## Project setup:

Read the .csv file in python. Use example code below to help you. The .csv data file can be found in the CFG datasets folder on google drive.

### Analysis ideas:

- Print/ plot the top 10 happy countries for each year
- Print/ plot the bottom 10 happy countries for each year
- Show the differences (increases/ decreases) between happiness scores for each country between 2015 and 2016
- Create a program that allows users to search for a country and return the happiness score

### Example code for 2016 file

[illegible]