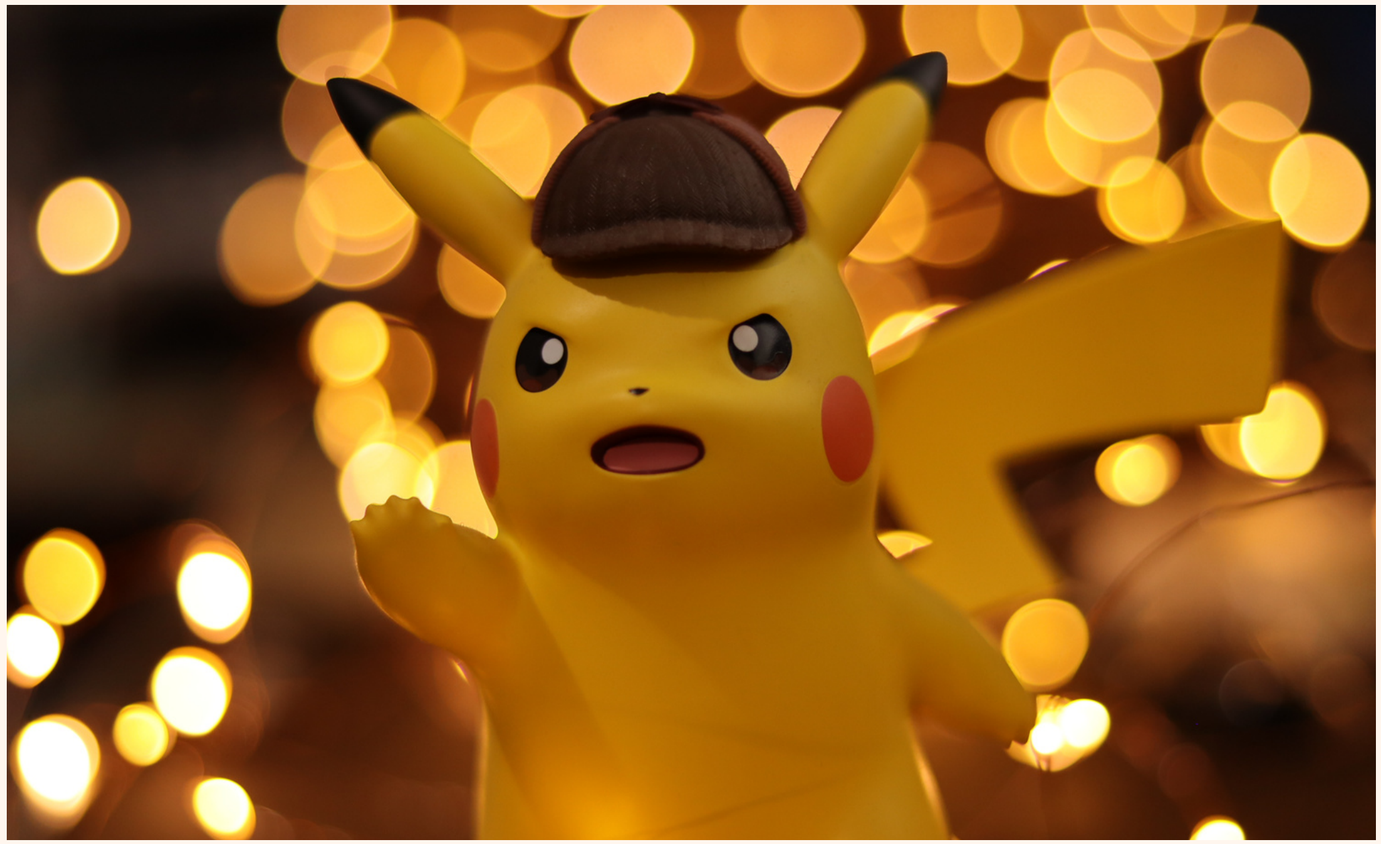
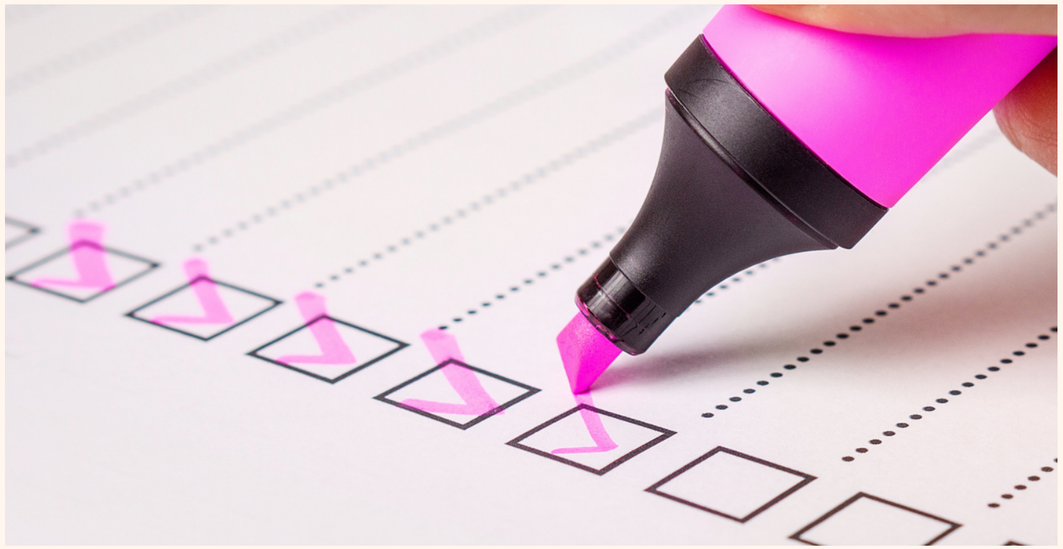


POKEMON TOP TRUMPS - A PYTHON PROJECT

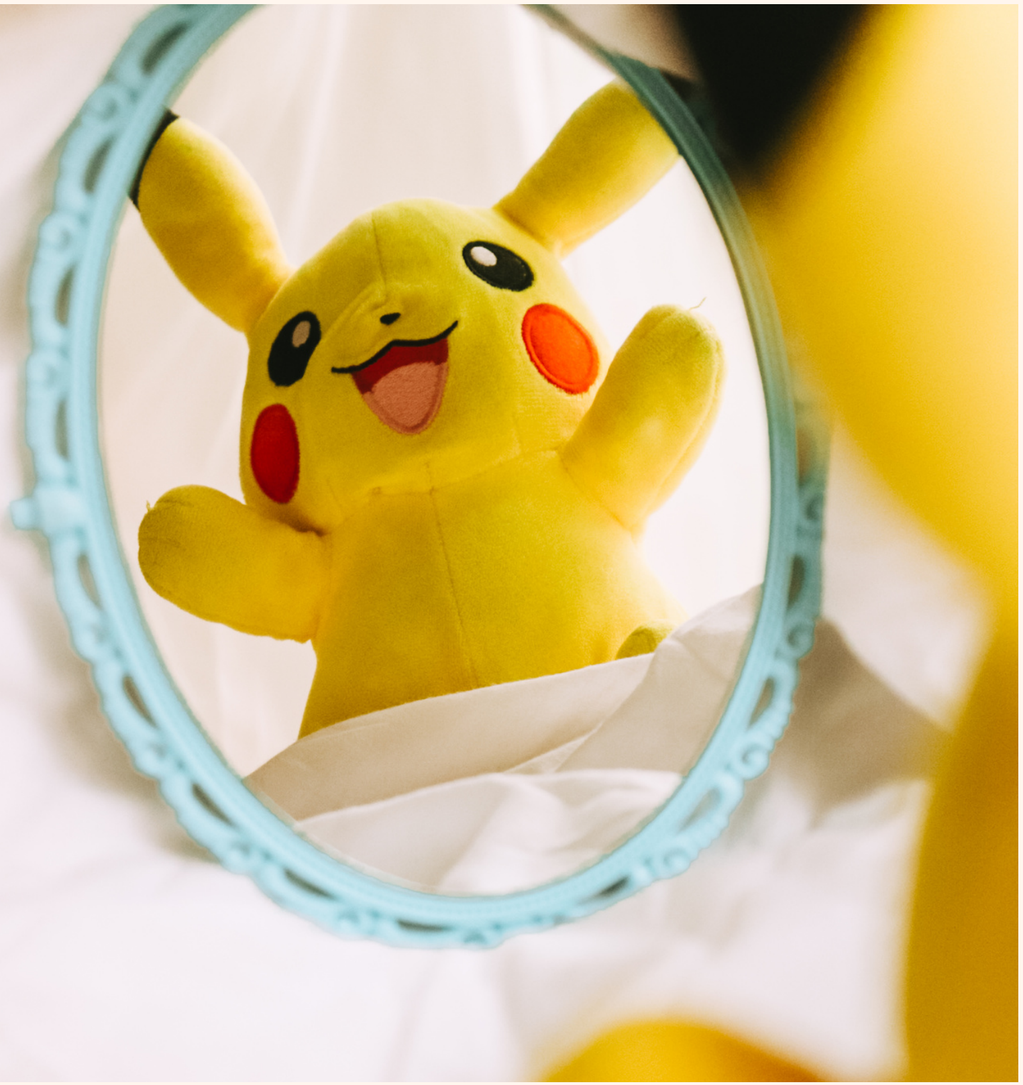
PRESENTED BY EMILY KNOX-CLIFTON, BECKS STUNELL AND ROMY TAYLOR

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



PROJECT OVERVIEW

Our Pokemon Top Trumps game utilises the PokeAPI to fetch information about Pokémon in the database.

The main game loop prompts the player to choose between different game modes (Normal Mode, Difficult Mode, or Exit).

In each round, the player is presented with a randomly chosen Pokémon and prompted to choose a stat (ID, height, or weight) to compare with a randomly generated opponent Pokémon.

The winner of each round is determined by comparing the chosen stats, and the game continues for three rounds.

MoSCoW

Must

- Generate a random number between 1 and 151 to use as the Pokemon ID number
- Use the Pokemon API to receive a character based on its ID number
- Get a random character for the player and another for the opponent.
- Ask the player which stat they want to use.
- Compare the player and opponent's stats.
- Declare a winner.

Should

- Check that the randomly chosen characters are not duplicated.
- Be able to handle cases where the API request fails, the users puts an invalid input, or unexpected issue occurs
- Work out what to do if data is missing from the stats.

Could

- Assign numerical data (for comparison) to non-numerical data.
- Include images.
- Play multiple rounds on top trumps and tally up a score. Best of 3 wins.
- Create a leaderboard.

Could

- Allow the opponent to choose a stat that they would like to compare, every other round
- Implement different levels of difficulty. In easy mode, the player has more information about the opponent's Pokemon character before choosing a stat. If it's hard mode, you can't see what has been drawn.
- Create a multiplayer mode where two players can compete against each other.

Won't

- Use all types of data in the API (due to some statistics being impossible to rank effectively.
- Allow you to type in other data not listed in the card stats.

HARRY POTTER?

During development we explored the option of using the Harry Potter API instead.

The decision not to use this API was based on the lack of directly comparable statistics beyond date of birth and wand length. While we could have assigned integers to non-numerical data to aid comparison, this would have been arbitrary.

Gryffindor = 1
Ravenclaw = 2
Slytherin = 3
Hufflepuff = 4 ... could cause arguments!

HARRY POTTER?

```python
import random
import requests

def random_harry_potter_character():
    character_number = random.randint(1, 25)
    url = f'https://hp-api.onrender.com/api/characters/{character_number}'

    try:
        response = requests.get(url)
        response.raise_for_status()  # Check for any HTTP errors

        harry_potter_output = response.json()
        return harry_potter_output

    except requests.RequestException as e:
        print(f"Error: {e}")
        return None

harry_potter_data = random_harry_potter_character()

if harry_potter_data:
    print(harry_potter_data)
else:
    print("Failed to fetch Harry Potter data.")
```

✓ 0.2s

```
Error: 404 Client Error: Not Found for url: https://hp-api.onrender.com/api/characters/8
Failed to fetch Harry Potter data.
```

- While exploring the Harry Potter option we found that the code which worked for the Pokemon API didn't work for the Harry Potter one.
- To discover why, we added some error reporting and realised it couldn't fetch data based on character ID number.

```python
import requests
import random

url = 'https://hp-api.onrender.com/api/characters/'
response = requests.get(url)

character_data = response.json()

random_character_data = character_data[random.randint(1, 25)]

print(random_character_data)
```

```
{'id': '861c4cde-2f0f-4796-8d8f-9492e74b2573', 'name': 'Luna Lovegood',
```

- Another option would have been to return all the data from the API, store that within the program and call the individual statistics from there.

POKEMON GAMEPLAY

The code

```
def run():
    # Print a welcome message to the player
    print("Welcome to Pokemon Top Trumps!")

    # Create an instance of the Player class to track persistent stats
    player = Player()

    # Continuous loop for the game
    while True:
        # Display available game modes
        print("\nGame Modes:")
        print("1. Normal Mode")
        print("2. Difficult Mode")
        print("3. Exit")

        # Allow the player to choose a game mode
        mode_choice = input("Choose a game mode (1/2/3): ")

        # Check if the player wants to exit the game
        if mode_choice == '3':
            print("Thanks for playing! Goodbye.")
            break

        # Introducing the round counter
        rounds = 0
        # Loop for each round (best of 3 rounds)
        while rounds < 3:
            # Display the current round number
            print("\nRound {}".format(rounds + 1))

            # Generate a random Pokemon for the player
            player_pokemon = random_pokemon()
            print("Your Pokemon:")
            display_pokemon(player_pokemon)

            # Ask the user which stat they want to use (id, height, or weight)
            valid_stats = ['id', 'height', 'weight']
            stat_to_compare = input("Which stat do you want to compare? (id/height/weight): ").lower()
```

The game offers modes for the player to choose (Normal, Difficult, or Exit) and enters a continuous loop initiating best-of-three rounds.

Each round involves generating a random Pokemon for the player, prompting them to choose a stat (ID, height, or weight) for comparison against an opponent Pokemon.

The winner is decided by stat comparison. After three rounds, the game displays the player's results, providing an option to exit or continue playing.

The result

```
Welcome to Pokemon Top Trumps!

Game Modes:
1. Normal Mode
2. Difficult Mode
3. Exit
Choose a game mode (1/2/3): 1

Round 1
Your Pokemon:
Name: sandslash
ID: 28
Height: 10 decimetres
Weight: 295 hectograms


Which stat do you want to compare? (id/height/weight): id
Your Pokemon's id: 28
```

CHECKING FOR DUPLICATES

```
# Function to get information about a random Pokemon from the PokeAPI
def random_pokemon():
    player_pokemon_number = random.randint( a: 1,  b: 151)
    opponent_pokemon_number = player_pokemon_number

    # Keep generating a new opponent's Pokemon until it's different from the player's Pokemon
    while opponent_pokemon_number == player_pokemon_number:
        opponent_pokemon_number = random.randint( a: 1,  b: 151)

    player_url = 'https://pokeapi.co/api/v2/pokemon/{}/'.format(player_pokemon_number)
    player_response = requests.get(player_url)
    player_pokemon = player_response.json()

    opponent_url = 'https://pokeapi.co/api/v2/pokemon/{}/'.format(opponent_pokemon_number)
    opponent_response = requests.get(opponent_url)
    opponent_pokemon = opponent_response.json()
```

We realised that we hadn’t accounted for the possiblity of both players drawing the same card so we added in a section to keep generating the opponent’s Pokemon until it is different from the player’s Pokemon.

POKEMON GAMEPLAY

The result

```
Round 1
Your Pokemon:
Name: sandslash
ID: 28
Height: 10 decimetres
Weight: 295 hectograms


Which stat do you want to compare? (id/height/weight): id
Your Pokemon's id: 28
Press Enter to reveal the opponent's Pokemon.
Name: shellder
ID: 90
Height: 3 decimetres
Weight: 40 hectograms


Opponent wins this round!
```

How a round of game play looks.The chosen stats are then compared between the player and an opponent Pokemon, with the winner determined at the end of each round.

The code

```python
# Generate a random Pokemon for the player
player_pokemon = random_pokemon()
print("Your Pokemon:")
display_pokemon(player_pokemon)

# Ask the user which stat they want to use (id, height, or weight)
valid_stats = ['id', 'height', 'weight']
stat_to_compare = input("Which stat do you want to compare? (id/height/weight): ").lower()

# Generate a random Pokemon for the opponent
opponent_pokemon = random_pokemon()

if stat_to_compare in valid_stats:
    player_stat = player_pokemon[stat_to_compare]
    opponent_stat = opponent_pokemon[stat_to_compare]

    print("Your Pokemon's {}: {}".format( *args: stat_to_compare, player_stat))

    # Display opponent's Pokemon after player's input
    input("Press Enter to reveal the opponent's Pokemon.")
    display_pokemon(opponent_pokemon)

    # Compare the stats and determine the winner
    if player_stat > opponent_stat:
        print("You win this round!")
        player.wins += 1
    elif player_stat < opponent_stat:
        print("Opponent wins this round!")
        player.losses += 1
    else:
        print("It's a tie!")
```

INVALID POKEMON GAMEPLAY

The code

```
# Ask the user which stat they want to use (id, height, or weight)
valid_stats = ['id', 'height', 'weight']
stat_to_compare = input("Which stat do you want to compare? (id/height/weight): ").lower()

# Generate a random Pokemon for the opponent
opponent_pokemon = random_pokemon()

if stat_to_compare in valid_stats:
    player_stat = player_pokemon[stat_to_compare]
    opponent_stat = opponent_pokemon[stat_to_compare]

    print("Your Pokemon's {}: {}".format( *args: stat_to_compare, player_stat))

    # Display opponent's Pokemon after player's input
    input("Press Enter to reveal the opponent's Pokemon.")
    display_pokemon(opponent_pokemon)

    # Compare the stats and determine the winner
    if player_stat > opponent_stat:
        print("You win this round!")
        player.wins += 1
    elif player_stat < opponent_stat:
        print("Opponent wins this round!")
        player.losses += 1
    else:
        print("It's a tie!")

    rounds += 1

else:
    print("Invalid stat. Please choose 'id', 'height', or 'weight.'")
```

If the player chooses an invalid stat, they will be presented with the following:

Invalid stat. Please choose 'id', 'height', or 'weight.

The result

```
Which stat do you want to compare? (id/height/weight): power
Invalid stat. Please choose 'id', 'height', or 'weight.'
```

TROUBLESHOOTING DIFFICULT MODE POKEMON GAMEPLAY

To improve difficult mode, we decided to hide the player's Pokemon character, meaning they have to choose between id, height or weight without seeing the stats.

The code

```
# Generate a random Pokemon for the player
player_pokemon = random_pokemon()
if mode_choice== '1':
    print("Your Pokemon:")
    display_pokemon(player_pokemon)

# Ask the user which stat they want to use (id, height, or weight)
valid_stats = ['id', 'height', 'weight']
stat_to_compare = input("Which stat do you want to compare? (id/height/weight): ").lower()
```

The result

```
Welcome to Pokemon Top Trumps!

Game Modes:
1. Normal Mode
2. Difficult Mode
3. Exit
Choose a game mode (1/2/3): 2

Round 1
Which stat do you want to compare? (id/height/weight): id
Your Pokemon's id: 109
Press Enter to reveal the opponent's Pokemon.
Name: ponyta
ID: 77
Height: 10 decimetres
Weight: 300 hectograms


You win this round!
```

EXAMPLE POKEMON TOP TRUMPS GAMEPLAY

```
Welcome to Pokemon Top Trumps!

Game Modes:
1. Normal Mode
2. Difficult Mode
3. Exit
Choose a game mode (1/2/3): 1

Round 1
Your Pokemon:
Name: rhyhorn
ID: 111
Height: 10 decimetres
Weight: 1150 hectograms


Which stat do you want to compare? (id/height/weight): id
Your Pokemon's id: 111
Press Enter to reveal the opponent's Pokemon.
Name: omastar
ID: 139
Height: 10 decimetres
Weight: 350 hectograms


Opponent wins this round!
```

```
Round 2
Your Pokemon:
Name: scyther
ID: 123
Height: 15 decimetres
Weight: 560 hectograms


Which stat do you want to compare? (id/height/weight): weight
Your Pokemon's weight: 560
Press Enter to reveal the opponent's Pokemon.
Name: poliwrath
ID: 62
Height: 13 decimetres
Weight: 540 hectograms


You win this round!
```

```
Round 3
Your Pokemon:
Name: victreebel
ID: 71
Height: 17 decimetres
Weight: 155 hectograms


Which stat do you want to compare? (id/height/weight): height
Your Pokemon's height: 17
Press Enter to reveal the opponent's Pokemon.
Name: spearow
ID: 21
Height: 3 decimetres
Weight: 20 hectograms


You win this round!

Game Over!
Wins: 2
Losses: 1

Game Modes:
1. Normal Mode
2. Difficult Mode
3. Exit
Choose a game mode (1/2/3): 3
Thanks for playing! Goodbye.

Process finished with exit code 0
```

END OF THE GAME - POKEMON

The final results, including the player's wins and losses, are displayed at the end of the game. The game can be exited at any time by selecting the "Exit" option. The script runs the game when executed.

```
Game Over!
Wins: 1
Losses: 2

Game Modes:
1. Normal Mode
2. Difficult Mode
3. Exit
Choose a game mode (1/2/3):
```

Game 1 - Normal

```
Game Over!
Wins: 4
Losses: 2

Game Modes:
1. Normal Mode
2. Difficult Mode
3. Exit
Choose a game mode (1/2/3):
```

Game 2 - Easy

```
Game Over!
Wins: 4
Losses: 5

Game Modes:
1. Normal Mode
2. Difficult Mode
3. Exit
Choose a game mode (1/2/3):
```

Game 3 - Difficult

After the game of 3 rounds has ended, the player get the choice to play again or to exit the game.

If the player decided to play again, they can pick a new game mode either ‘Normal’ or ‘Difficult’.

As long as the script hasn’t been executed, the ‘Wins’ and ‘Losses’ will continue to carry over each game.

```
Game Modes:
1. Normal Mode
2. Difficult Mode
3. Exit
Choose a game mode (1/2/3): 3
Thanks for playing! Goodbye.

Process finished with exit code 0
```

Exiting the game