**What’s the Hottest City?**

* **Note**: This project is provided by **Codecademy** (<https://www.codecademy.com>)
* **Note**: I did this project in the **Ubuntu** terminal.

**Project description**

The purpose of the project is to determine the hottest city. I will:

* Fetch forecast temperatures for multiple cities.
* Save these temperatures to a file.
* Sort the cities based on their temperature.

**Start Project Timestamp**

A screenshot of a computer

Description automatically generated

**Project Setup**

1. I first downloaded a modified **weather** script offered by Codecademy. This script is a modification from **Bash-Snippets** at <https://github.com/alexanderepstein/Bash-Snippets>. The command (highlighted in yellow) that I used for downloading the Codecademy version is shown below:

* **Optional**: Installing the original **weather** from **Bash-Snippets is optional**. However, it is recommended if you are familiar with **git** and if you wish to further improve on the script. The API link for this project is <https://wttr.in/>.

A screen shot of a computer

Description automatically generated

1. I then gave the script executable permissions with the following command.



1. To se the usage of the script, I used the following command (highlighted in yellow).

A screen shot of a computer

Description automatically generated

**Note**: Using the **-s** argument when using the **./weather.sh** command will output a simpler version of the weather. An example of this is shown below:

A black background with white text

Description automatically generated

The fancy version without the **-s** argument is shown below:

A screenshot of a computer

Description automatically generated

**Create Script File**

1. I created a shell script file with the **touch** command called **sortCityTemps.sh** and gave it permissions with **chmod +x** command to execute:

A black background with white text

Description automatically generated

1. Choose a method for working in the file. This file can be worked in either the **Ubuntu Desktop Editor** (which comes pre-installed), a code editor of your choosing, or inside the terminal itself with **nano**. You can open the file in Ubuntu Desktop Editor using the command **gedit sortCityTemps.sh &**. If you decide to use nano, you can save changes to the file using **CTRL/CMD + O** and exit using **CTRL/CMD + X.**
   1. I chose the **nano** code editor with the Ubuntu terminal and added the line **#!/bin/bash** to the top of the script and saved.

A yellow line on a black background

Description automatically generated

A black and white image of a person with a white background

Description automatically generated with medium confidence

A screen shot of a computer

Description automatically generated

**Collect and Sort Temperatures**

1. I created an array of cities of my own choosing that I will be getting the temperatures from.

A black and white screen with yellow text

Description automatically generated

1. In order to save the temperatures, I created an empty file called **temperatures.txt**. The “**>”** command was used to create an empty file.
2. Next, I set up a loop to iterate through the cities, along with the temperatures, with the **-s** (simple) argument in the array I created. I included a **sleep 1** command within the loop that pauses the script for 1 second at every iteration to prevent too many requests to the API, if one is being used, in a short amount of time.

A screen shot of a computer program

Description automatically generated

The output of the script in Ubuntu is shown below:

A screen shot of a computer

Description automatically generated

1. I order to avoid having issues with sorting the temperatures, I removed the “**+**” and “**°F**” symbols. To do this I **piped** the output of the weather script using the **sed** command and replaced each symbol with an empty string. Which is shown below:

A screen shot of a computer screen

Description automatically generated

Output in Ubuntu without the symbols shown below:

A black screen with white text

Description automatically generated

1. I then redirected the iteration to the **temperatures.txt** file and removed to echo command to avoid overriding on each iteration.

A screen shot of a computer

Description automatically generated

1. I ran the script and checked out the contents of the **temperatures.txt** file with the **cat** command.

A screen shot of a computer

Description automatically generated

1. I sorted the cities based on the temperatures. Using the **sort** command to do this along with the **-k2** argument to sort on the second value in each line (first value is city and second value is the temperature) and the **-r** argument to sort in descending order, this is the output in the Ubuntu terminal:

A black screen with white text

Description automatically generated

1. Finally, I added this sorted line to the script after the loop and outputted it to file called **sorted\_tepmeratures.txt**, ran the script in Ubuntu, and checked the contents of the new file.

A screen shot of a computer

Description automatically generated

A screen shot of a computer screen

Description automatically generated

**End Project Timestamp**

A screen shot of a computer

Description automatically generated