



## Coding Challenge

### Context

In America, taxes are often complex and require many different PDF forms and posted informational notices. The IRS keeps records of the current tax year's forms and historical forms going back many years online on their [Prior Year Products](#) page.

### Challenge

For this challenge you must write two different utilities for searching IRS tax forms:

- Taking a list of tax form names (ex: "Form W-2", "Form 1095-C"), search the website and return some informational results. Specifically, you must return the "Product Number", the "Title", and the maximum and minimum years the form is available for download. The forms returned should be an exact match for the input (ex: "Form W-2" should not return "Form W-2 P", etc.) The results should be returned as json, in the format of the following example:

```
[
  {
    "form_number": "Form W-2",
    "form_title": "Wage and Tax Statement (Info Copy Only)",
    "min_year": 1954,
    "max_year": 2021
  }
  ...
]
```

- Taking a tax form name (ex: "Form W-2") and a range of years (inclusive, 2018-2020 should fetch three years), download all PDFs available within that range. The forms returned should be an exact match for the input (ex: "Form W-2" should not return "Form W-2 P", etc.) The downloaded PDFs should be downloaded to a subdirectory under your script's main directory with the name of the form, and the file name should be the "Form Name - Year" (ex: Form W-2/Form W-2 - 2020.pdf)



## Requirements

The challenge must be written in Python 3 and can use any publicly available library. If you use any non-standard libraries, you MUST include a standard pip-compatible "requirements.txt" file outlining every requirement needed to run your script(s).

You may use any method for taking input and returning required output from your script (command line arguments, command line input, files, etc.)

You must include a README.txt file explaining which specific version of Python is used (ex: Python 3.8.0), how to properly run your script (including how it takes input parameters and how it outputs JSON for part 1.) Also feel free to use this README.txt to add any additional info you'd like to share about your script or any feedback on the challenge itself.

## Submitting your code

Please submit your code along with any applicable files via the submission link provided in the coding challenge invitation. We recommend submitting a zipped file so you won't have any issues. Don't hesitate to reach out to your recruiter or [recruiting@getpinwheel.com](mailto:recruiting@getpinwheel.com) with any questions! Always happy to help. :)