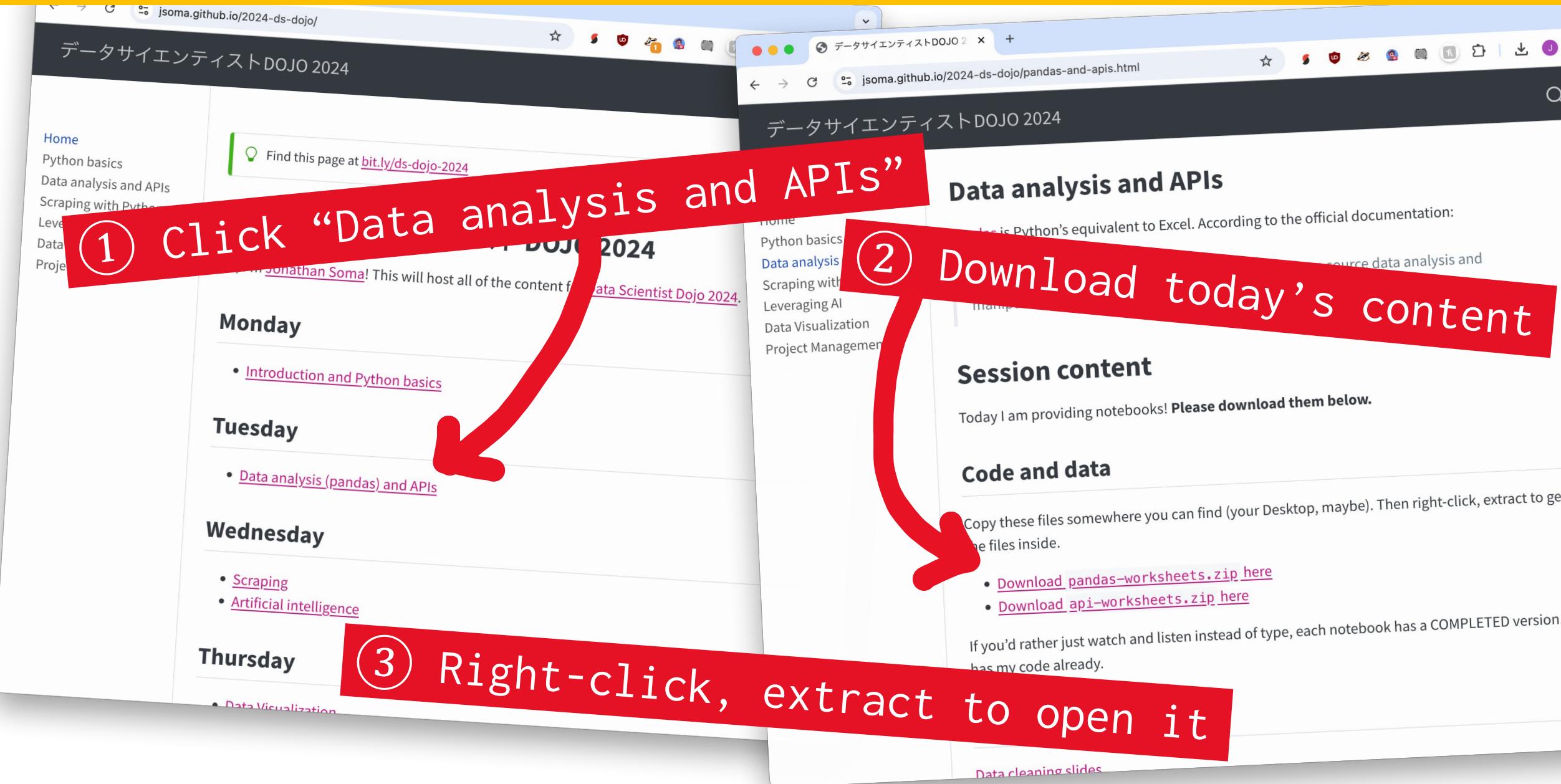


Visit <https://bit.ly/ds-dojo-2024> for materials

Structured data

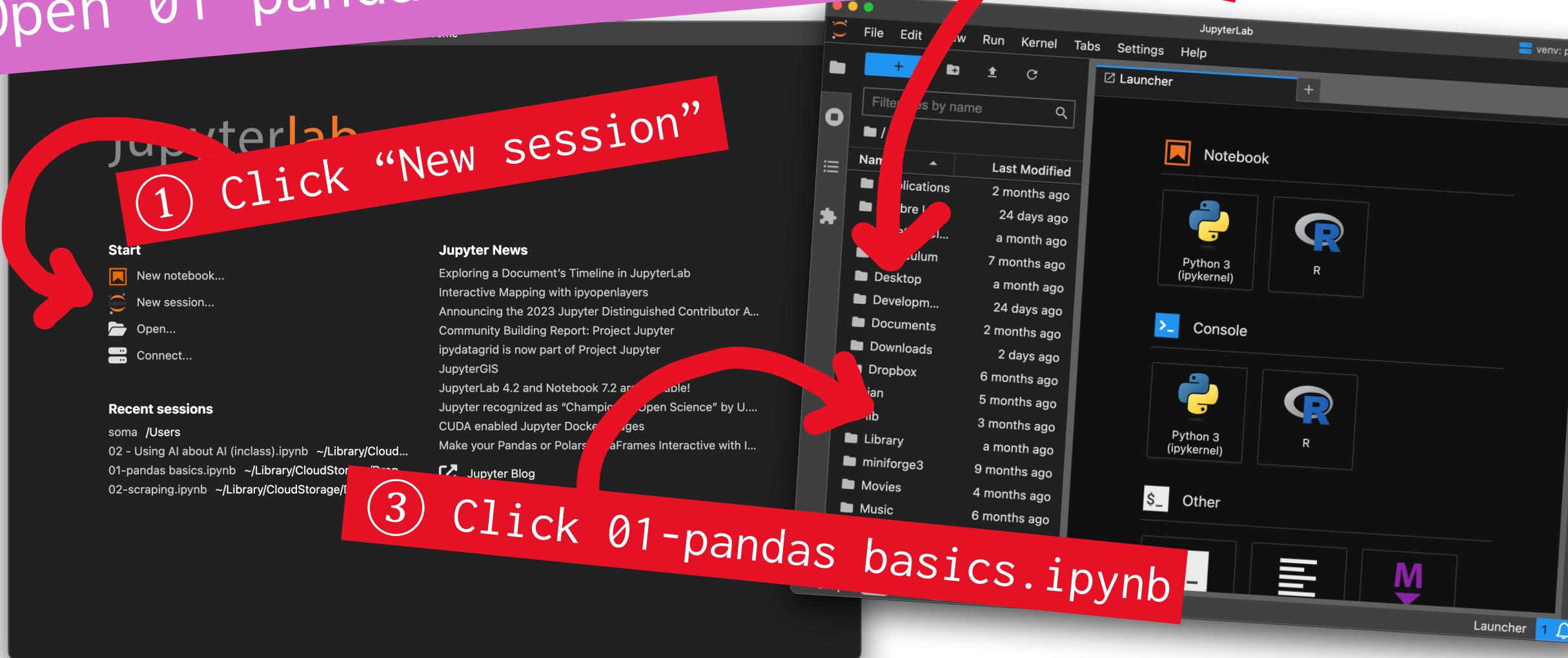
Pandas, CSVs, Excel and APIs

Visit <https://bit.ly/ds-dojo-2024> for materials



Visit <https://bit.ly/ds-dojo-2024> for materials

Open 01-pandas basics.ipynb



Now let's do a
little pandas

Basic data cleaning

replace and split are
99% of the game

66 years old

66 years old

66

convert to number

replace “years old”
with... nothing!

j soma, js4571@columbia.edu, @dangerscarf

j soma | js4571@columbia.edu | @dangerscarf

js4571@columbia.edu



j soma, js4571@columbia.edu, @dangerscarf

j soma | js4571@columbia.edu | @dangerscarf

take second

replace with nothing

js4571@columbia.edu

take second

js4571 | columbia.edu
columbia.edu

split on @

split

I have text like "j soma, js4571@columbia.edu, @dangerscarf" in a Google Sheet and I want to split it by ", ", take the second part, then replace the "@columbia.edu" with nothing. Just give me the formula.

*just ask!
then TEST IT*



You can use the following Google Sheets formula to achieve this:

plaintext

Copy code

```
=SUBSTITUTE(INDEX(SPLIT(A1, ", "), 2), "@columbia.edu", "")
```

Explanation:

1. `SPLIT(A1, ", ")` splits the text in cell `A1` by ", ".
2. `INDEX(SPLIT(A1, ", "), 2)` selects the second part of the split text.
3. `SUBSTITUTE(INDEX(SPLIT(A1, ", "), 2), "@columbia.edu", "")` removes the "@columbia.edu" part from the selected text.



*just ask!
then TEST IT*

I have text like "j soma, js4571@columbia.edu, @dangerscarf" in a Google Sheet and I want to extract the "@columbia.edu" part. Write me a formula to do this. Only show me the formula.



excel

Copy code

```
=RIGHT(A1,LEN(A1)-FIND("@",A1)+1)
```



Feel free to reach out at 555-555-5555



555-555-5555

just ask!
then TEST IT

8

I have text like "j soma, js4571@columbia.edu, @dangerscarf" in a pandas dataframe column named "details" and I want to split it by ", ", take the second part, then replace the "@columbia.edu" with nothing. Try not to use .apply. Don't create the df, just give me the code.

< 6/8 >



Certainly! Here's the code to achieve that:

python

Copy code

```
# Split the 'details' column by ", " and take the second part
df['email'] = df['details'].str.split(', ').str[1]

# Replace "@columbia.edu" with nothing
df['email'] = df['email'].str.replace('@columbia.edu', '', regex=False)
```

This will split the column and then replace the domain part as required.



just ask!
then TEST IT

I have text like "j soma, js4571@columbia.edu, @dangerscarf" in a pandas dataframe column named "details" and I want to extract the "@columbia.edu" part. Try not to use .apply. Don't create the df, just give me the code.



You can use the `str.extract` method to extract the "@columbia.edu" part from the "details" column. Here's the code to do that:

python

Copy code

```
df['extracted'] = df['details'].str.extract(r'(\b@\w+\.\edu\b)')
```



Regular expressions

regex is the other
other 1%

66 years old

hug what we want
(capture group)

(66) years old

match anything

(.*) years old

I am 66 years old



I am (66) years old



I am (.*) years old

Contact me at 555-555-5555

Contact me at (.*)

Contact me at 555-555-5555 or
js4571@columbia.edu

Contact me at (\d\d\d-\d\d\d-\d\d\d\d)

but maybe not!

hug what we want

just numbers

Contact me at 555-555-5555

(\d\d\d-\d\d\d-\d\d\d\d)



maybe we don't
need an anchor?

Contact me at 5555555555

(\d\d\d-?\d\d\d-?\d\d\d\d)



Contact me at 555+555+5555

(\d\d\d.?\d\d\d.?\d\d\d\d)

*any character is fine!
(and also optional)*

I have a table that looks like this. Write the SIMPLEST POSSIBLE regex to just extract the values for "Deforested area" in km. Maybe anchor around the year at the start of each line?

just ask!
then TEST IT

Deforestation of the Amazon rainforest[9][10][11]		
	Year Deforested area	Accumulated loss
2005	10,4 km2 (7,341 sq mi)	332,470 km2 (128,370 sq mi)
	6 km2 (5,516 sq mi)	346,756 km2 (133,883 sq mi)
	km2 (4,498 sq mi)	358,407 km2 (138,382 sq mi)
	11 km2 (4,985 sq mi)	371,318 km2 (143,367 sq mi)
2009	7,464 km2 (2,882 sq mi)	378,782 km2 (146,249 sq mi)
2010	7,000 km2 (2,700 sq mi)	385,782 km2 (148,951 sq mi)
2011	6,418 km2 (2,478 sq mi)	392,200 km2 (151,400 sq mi)
2012	4,571 km2 (1,765 sq mi)	396,771 km2 (153,194 sq mi)
2013	5,891 km2 (2,275 sq mi)	402,662 km2 (155,469 sq mi)
2014	4,848 km2 (1,872 sq mi)	407,510 km2 (157,340 sq mi)

< 5/5 >

To extract the "Deforested area" values in km² using a regex, you can use the following simple pattern:

scss

[Copy code](#)

```
^\d{4}\s+(\d{1,5})\s+km2
```

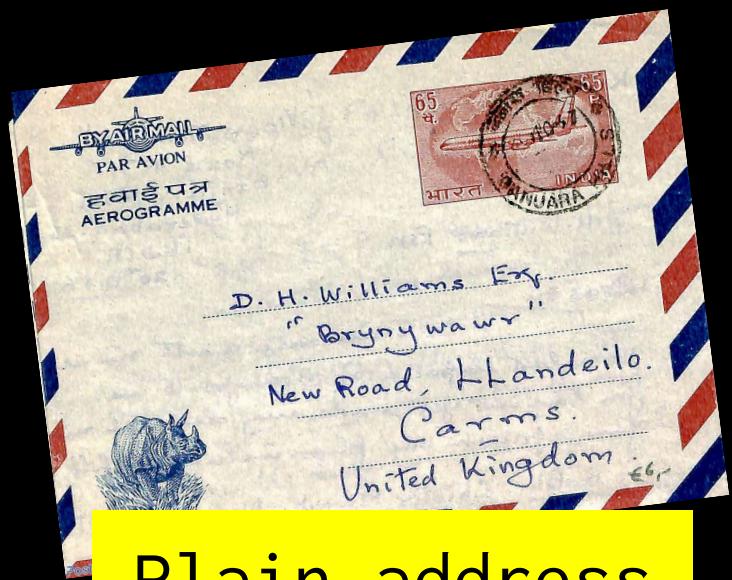
Here's a breakdown of the regex:

- `^` asserts the position at the start of the line.
- `\d{4}` matches exactly four digits (the year).
- `\s+` matches one or more whitespace characters.
- `(\d{1,5})` captures between 1 and 5 digits (the deforested area value) in a capturing group.
- `\s+` matches one or more whitespace characters.
- `km2` matches the literal string "km2".

This pattern will capture the deforested area in km² from each line.



Addresses and maps



Plain address

Geocoding



Latitude/Longitude

Reverse geocoding

Hassle-free geocoding

UPLOAD SPREADSHEET GET AN API KEY

Straightforward and easy-to-use geocoding, reverse geocoding, and data matching for US and Canadian addresses.

drag and drop: geocod.io

Used by thousands of happy customers

ORACLE Stanford PRO

sheets: geocode by awesome table

Geocode by Aweso...

Geocode is a tool that helps you get latitudes & longitudes from addresses in a Google Sheet to display them on a map you can share.

By: Talarian Listing updated: April 1, 2024

Uninstall

Works with:

Overview Permissions Reviews

Custom tooltip

Geocoder: Simple, Consistent

Release v1.38.1. [\(Installation\)](#)

Simple and consistent geocoding library written in Python. Many online providers such as Google & Bing have geocoding services, these providers do not include Python libraries and have different JSON responses between each other. It can be very difficult sometimes to parse a particular geocoding provider since each one of them have their own JSON schema.

Lat & Lng from

python: geocoder

```
com/maps/api/geocode/json'
address': 'Mountain View, CA'}
[{"location"]
```

Generating structured data
with the awful power of LLMs

FROM: Mulberry Peppertown
(mulbs@example.com)

When I pick up the cans of beans they are all so light! At first I thought they were empty, but it turns out they are just futuristic beans that are not heavy like the old style beans I was used to. It is incredible.

Mulberry Peppertown

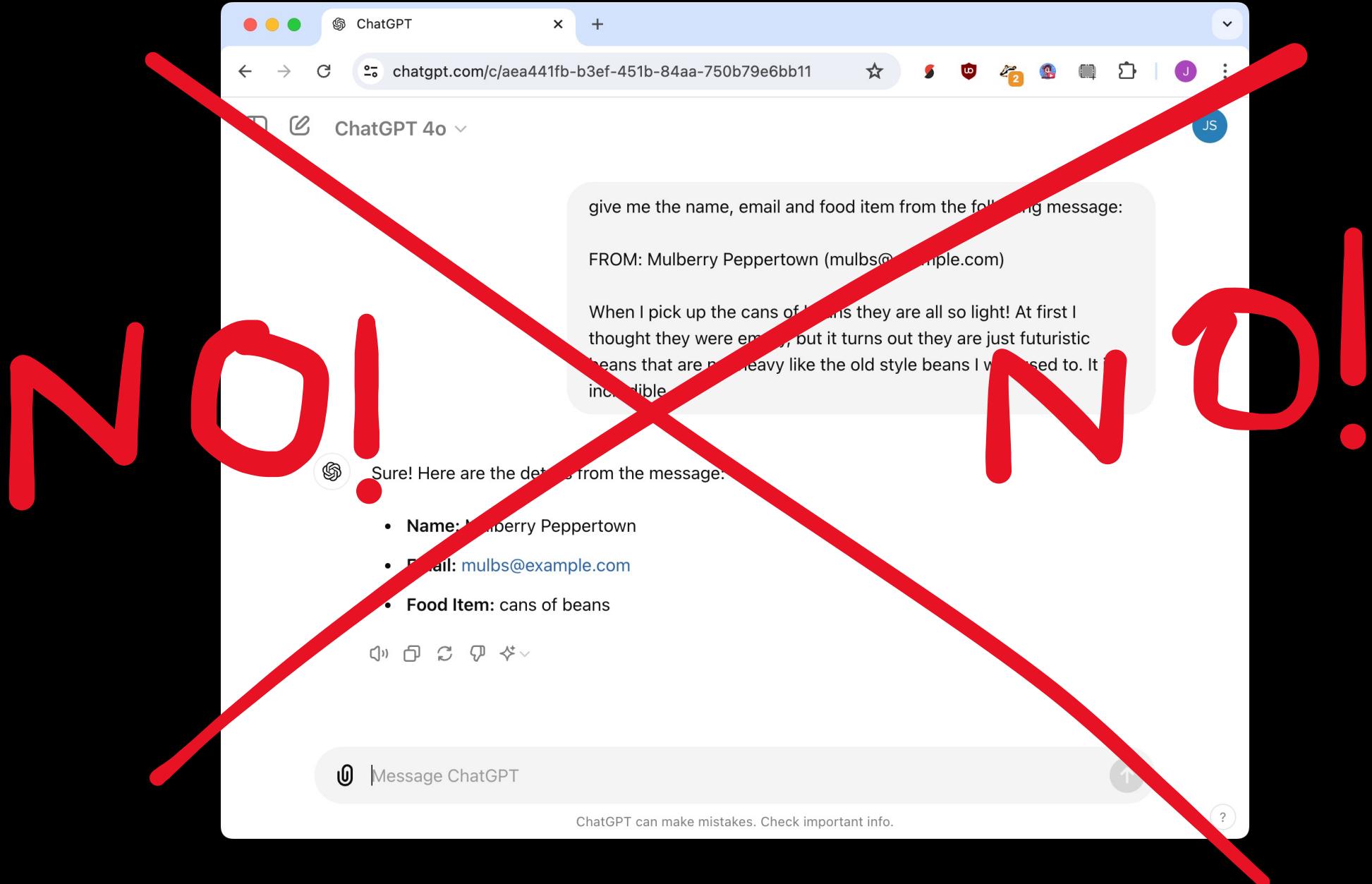
name

mulbs@example.com

email

beans

food item



email	name	email	product
FROM: Mulberry Peppertown (mulbs@example.com)			

When I pick up

=CLAUDEEXTRACT("grocery product, mention all if there are multiple", A2)

incredible.

I am irate about
shopping at yo
broccoli mailin

Jackary Balon
jackary.balone



Works with:



★★★★☆ 21 ⓘ ↓ 45K+

More details about user reviews

Claude for Sheets

Uninstall

Claude for Sheets
from Anthropic
By: Anthropic ⓘ
Listing updated:

=CLAUDEEXTRACT("name", A2)

=CLAUDEEXTRACT("email", A2)

Sheets: Claude for Sheets

Python: Instructor

```
class Comment(BaseModel):
    name: str = Field(description="Person who submitted the comment")
    email: Optional[str] = Field(description="Email address of commenter")
    food_item: str = Field(description="Food item the comment is about")
    emotion: Literal["positive", "negative", "uncertain"]
```

```
comment = """
FROM: Mulberry Peppertown, mulberry (at) example.co
When I pick up the cans of beans they are all so li
first I thought they were empty, but it turns out to
futuristic beans that are not heavy like the old s
I was used to. It is incredible.
"""

result = client.chat.completions.create(
    response_model=Comment,
    messages=[{"role": "user", "content": comment}],
```

```
{
    'name': 'Mulberry Peppertown',
    'email': 'mulberry@example.com',
    'food_item': 'cans of beans',
    'emotion': 'positive'
```

now let's do a
little more pandas

it will be... fun?

Visit <https://bit.ly/ds-dojo-2024> for materials

Structured data

Pandas, CSVs, Excel and APIs

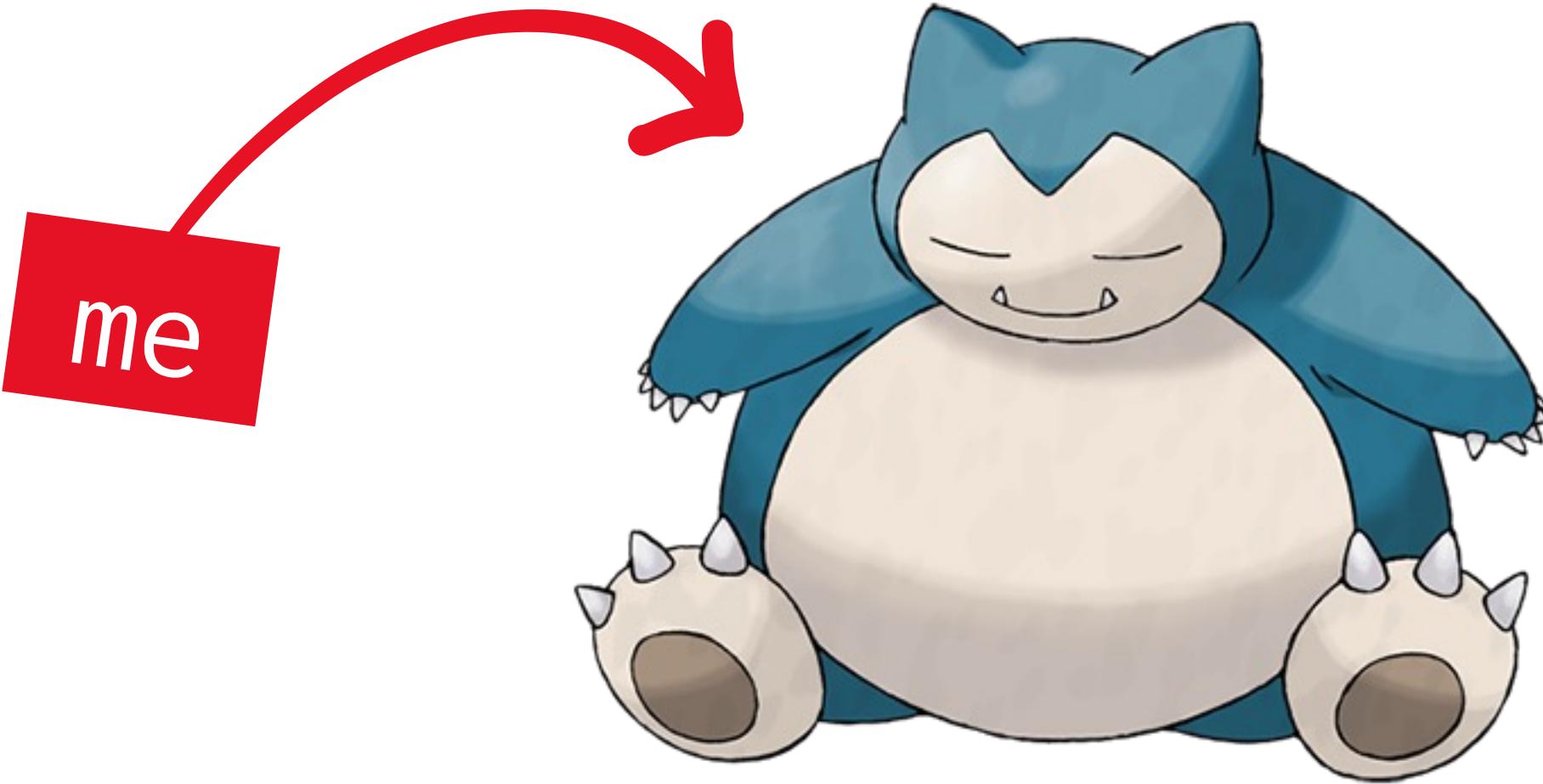
Visit <https://bit.ly/ds-dojo-2024> for materials

APIs

structured data from the
internet

Does anyone have a
favorite pokemon?

This is an important question



me

Google Search x +

google.com/search?q=snorlax&oq=snorlax&gs_lcrp=EgZjaHJvbWUyDwgAEEUYORIRAhBABiKBTIMCAEQABhDGIAEGloFMg0IAhAAGJECGIAEGloFMgwIAxAA... ⋮

snorlax X Microphone Image Search

Google J

All Images Shopping Videos News Maps Web More Tools

Bulbapedia [https://bulbapedia.bulbagarden.net/wiki/Snorlax_\(Pok%C3%A9mon\)](https://bulbapedia.bulbagarden.net/wiki/Snorlax_(Pok%C3%A9mon)) ⋮

Snorlax (Pokémon) - Bulbapedia

Snorlax is a huge, bipedal, dark blue-green mammalian Pokémon with a cream-colored face, belly, and feet.

Ash's Snorlax III Menu Pokémon Sleep

People also ask :

What is Snorlax in Japanese? ⌄

Why is Snorlax sleepy? ⌄

What type of animal is a Snorlax? ⌄

What is Snorlax power? ⌄

Feedback

Pokemon.com <https://www.pokemon.com/pokedex/snorlax> ⋮

Snorlax | Pokédex

This gluttonous Pokémon eats constantly, apart from when it's asleep. It devours nearly 900 pounds of food per day.

Snorlax :



More images

Snorlax, known in Japan as Kabigon, is a Pokémon species in Nintendo and Game Freak's Pokémon franchise.

[Wikipedia](#)

Type: Normal

Weakness: Fighting

Ability (hidden): Gluttony

Abilities: Thick Fat, Immunity

Evolves from: Munchlax

Category: Sleeping

Weight: 460 kg

People also search for

Snorlax (Pokémon) - Bulbape... +

bulbapedia.bulbagarden.net/wiki/Snorlax_(Pokémon)

BULBAGARDEN News Bulbapedia Forums Discord More Editors Account Search Bulbapedia

article discussion view source history tools in other languages

In commemoration of Nintendo's 135th anniversary, September 20 to 30 is NIWA Cross-Wiki Week. [Click here](#) for event details and the possibility to win a \$20/€20 Nintendo eShop gift card.
Please remember to follow the [manual of style](#) and [code of conduct](#) at all times.
Check the [Bulbagarden home page](#) for up-to-date Pokémon news and discuss it on the [forums](#) or in the [Bulbagarden Discord server](#).

Snorlax (Pokémon)

For Pokémon GO information on this species, see [the game's section](#).

←  #0142: Aerodactyl Pokémon #0144: Articuno →

This article is about the species. For a specific instance of this species, see [Snorlax \(disambiguation\)](#).

Snorlax (Japanese: カビゴン *Kabigon*) is a Normal-type Pokémon introduced in [Generation I](#). It evolves from [Munchlax](#) when leveled up with high friendship. Snorlax can [Gigantamax](#) into [Gigantamax Snorlax](#) if it has the [Gigantamax Factor](#). In Generations I and II games, and [Pokémon X and Y](#), wild Snorlax can be found sleeping in inconvenient locations. In order to catch one, a Trainer must first awaken it with music from either a [Poké Flute](#) or the [Pokégear](#) set to the Poké Flute channel. Snorlax is the game mascot of [Pokémon Sleep](#).

Contents [hide]

- 1 Biology
 - 1.1 Forms
 - 1.2 Evolution
- 2 Game data
 - 2.1 NPC appearances
 - 2.2 Pokédex entries
 - 2.3 Game locations
 - 2.3.1 In side games
 - 2.3.2 In events
 - 2.3.2.1 Wild Area News
 - 2.4 Held items
 - 2.5 Stats
 - 2.5.1 Base stats
 - 2.5.2 Pokéathlon stats
 - 2.6 Type effectiveness
 - 2.7 Learnset

Snorlax #0143
Sleeping Pokémon Kabigon



Snorlax

Gigantamax Snorlax

Images on the Bulbagarden Archives

Snorlax (Pokémon) - Bulbape... X +

bulbapedia.bulbagarden.net/wiki/Snorlax_(Pokémon)

2.5.1 Base Stats
2.5.2 Pokéathlon stats
2.6 Type effectiveness
2.7 Learnset
2.7.1 By leveling up
2.7.2 By TM
2.7.3 By breeding
2.7.4 By a prior Evolution
2.8 Side game data
2.9 Form data
2.9.1 Gigantamax
2.10 Evolution data
2.11 Sprites

3 In animation
3.1 Main series
3.1.1 Major appearances
3.1.1.1 Ash's Snorlax
3.1.1.2 Other
3.1.2 Minor appearances
3.1.3 Pokédex entries
3.2 Pokémon Origins
3.3 Pokémon Masters Animated Trailer
3.4 GOTCHA!
3.5 Pokémon Evolutions
3.6 The Adventures of Snorlax & Cubone

4 In the manga
4.1 Ash & Pikachu
4.2 The Electric Tale of Pikachu
4.3 Pokémon Adventures
4.4 Pokémon Battle Frontier
4.5 Pokémon: Yeah! I Got Pokémon!
4.6 Pokémon Mystery Dungeon: Ginji's Rescue Team
4.7 Pokémon Journeys
4.8 Pokémon Pocket Monsters
4.9 Pokémon Ruby-Sapphire
4.10 Snorlax's Dream Gourmet

Gigantamax Snorlax

Images on the Bulbagarden Archives

Type	Normal						
Abilities	Gluttony Hidden Ability Immunity or Thick Fat						
Gender ratio	87.5% male, 12.5% female						
Catch rate	25 (7.7%)						
Breeding	Egg Group: Monster Hatch time: 40 cycles						
Height	6'11" (Snorlax) 114'10"+ (Gigantamax Snorlax)						
Weight	2.1 m (Snorlax) 35.0+ m (Gigantamax Snorlax) 1014.1 lbs. (Snorlax) ??? lbs. (Gigantamax Snorlax) 460.0 kg (Snorlax) ??? kg (Gigantamax Snorlax)						
Base experience yield	154 (Gen. I-IV) 189 (V+)						
Leveling rate	Slow						
EV yield	Total: 2 <table border="1"><tbody><tr><td>2 (HP)</td><td>0 (Atk)</td><td>0 (Def)</td><td>0 (Sp.Atk)</td><td>0 (Sp.Def)</td><td>0 (Speed)</td></tr></tbody></table>	2 (HP)	0 (Atk)	0 (Def)	0 (Sp.Atk)	0 (Sp.Def)	0 (Speed)
2 (HP)	0 (Atk)	0 (Def)	0 (Sp.Atk)	0 (Sp.Def)	0 (Speed)		
Shape							
Footprint							
Pokédex color	Black						
Base friendship	70						

Structured vs unstructured data

- Easy for **computers** to understand and process
- CSV files + XML, JSON, Excel, etc etc etc
- Easy for **humans** to understand and process
- Websites, books, articles, etc etc etc

APIs

Application Programming Interface

A screenshot of a web browser displaying the PokeAPI homepage. The page features a large pink header bar with the word "APIS" in white. Below this is a red navigation bar with links for "Home", "About", "API v2", and "GraphQL v1beta". The main content area has a dark background with the "PokeéAPI" logo in yellow and blue. It also contains the text "The RESTful Pokémon API" and "Serving over 2.5 billion API calls each month!". A red arrow points from a red banner on the left that says "Structured data" towards the "PokeéAPI" logo. To the right of the logo is a yellow banner with the URL "https://pokeapi.co". At the bottom, there's a search bar with the URL "https://pokeapi.co/api/v2/pokemon/ditto", a magnifying glass icon, and a blue "Submit" button.

APIS

PokeéAPI

The RESTful Pokémon API

Serving over 2.5 billion API calls each month!

Structured data

All the Pokémon data you'll ever need in one place,
easily accessible through a modern free open-source RESTful API.

Check out the docs!

Try it now!

https://pokeapi.co/api/v2/pokemon/ditto

Submit

earthquake.usgs.gov/fdsnws/ x +

earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2014-01-01&endtime=2014-01-02

Pretty-print

```
{  
  "type": "FeatureCollection",  
  "metadata": {  
    "generated": 1727127591000,  
    "url": "https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2014-01-01&endtime=2014-01-02",  
    "title": "USGS Earthquakes",  
    "status": 200,  
    "api": "1.14.1",  
    "count": 325  
  },  
  "features": [  
    {  
      "type": "Feature",  
      "properties": {  
        "mag": 1.29,  
        "place": "10km SSW of Idyllwild, CA",  
        "time": 1388620296020,  
        "updated": 1457728844428,  
        "tz": null,  
        "url": "https://earthquake.usgs.gov/earthquakes/eventpage/ci11408890",  
        "detail": "https://earthquake.usgs.gov/fdsnws/event/1/query?eventid=ci11408890&format=geojson",  
        "felt": null,  
        "cdi": null,  
        "mmi": null,  
        "alert": null,  
        "status": "reviewed",  
        "tsunami": 0,  
        "sig": 26,  
        "net": "ci",  
        "code": "11408890",  
        "ids": ",ci11408890,",  
        "sources": ",ci,",  
        "types": ",cap,focal-mechanism,nearby-cities,origin,phase-data,scitech-link,",  
        "nst": 39,  
        "dmin": 0.06729,  
        "rms": 0.09,  
        "gap": 51,  
        "magType": "ml",  
        "type": "earthquake",  
        "title": "M 1.3 - 10km SSW of Idyllwild, CA"  
      },  
      "geometry": {  
        "type": "Point",  
        "coordinates": [-116.7776667, 33.6633333, 11.008]  
      },  
      "id": "ci11408890"  
    },  
    {  
      "type": "Feature",  
      "properties": {  
        "mag": 1.1,  
        "place": "Central Alaska",  
        "time": 1388620046501,  
        "updated": 1689800179099,  
        "tz": null,  
        "url": "https://earthquake.usgs.gov/earthquakes/eventpage/ak01421ig3u",  
        "detail": "https://earthquake.usgs.gov/fdsnws/event/1/query?eventid=ak01421ig3u&format=geojson"  
      }  
    }  
  ]  
}
```

API Documentation - Earthquake Catalog

earthquake.usgs.gov/fdsnws/event/1/



Earthquake Hazards Program

← Feeds and Notifications

Real-time Notifications

Earthquake Notification Service

Tweet Earthquake Dispatch

Real-time Feeds

ATOM

KML

Spreadsheet

QuakeML

GeoJSON Summary

GeoJSON Detail

For Developers

API Documentation - EQ Catalog

Change Log

Feed Lifecycle Policy

Developer's Corner

API Documentation - Earthquake Catalog

This is an implementation of the [FDSN Event Web Service Specification](#), and allows custom searches for earthquake information using a variety of parameters.

i Please note that automated applications should use [Real-time GeoJSON Feeds](#) for displaying earthquake information whenever possible, as they will have the best performance and availability for that type of information.

URL

<https://earthquake.usgs.gov/fdsnws/event/1/>

request known enumerated parameter values for the interface.

- <https://earthquake.usgs.gov/fdsnws/event/1/application.json>

application.wadl

request WADL for the interface.

- <https://earthquake.usgs.gov/fdsnws/event/1/application.wadl>

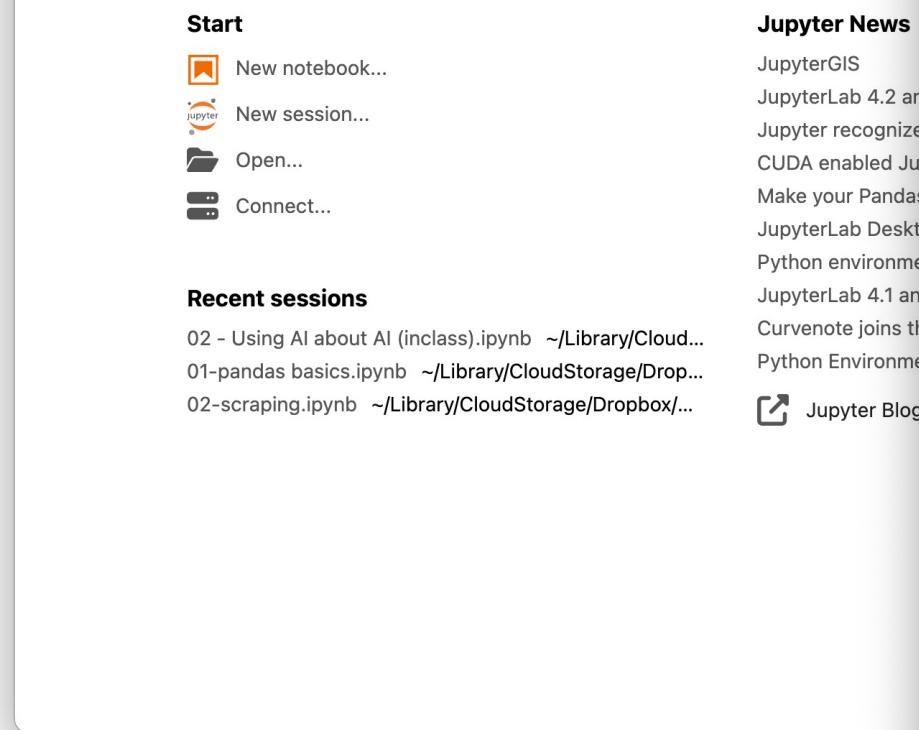
catalogs

request available catalogs.

- <https://earthquake.usgs.gov/fdsnws/event/1/catalogs>

contributors

jupyterlab



Open JupyterLab

The screenshot shows a JupyterLab notebook titled "Untitled2.ipynb". The code in the notebook uses the Playwright library to scrape data from a table. It starts by importing `async_playwright` and creating a browser instance. Then it waits for the table to become visible and queries all rows. For each row, it extracts the first three cells and creates a dictionary with keys "License Name", "Expiration Date", and "License Holder". Finally, it creates a DataFrame from the data and prints its head.

```
[ ]: from playwright.async_api import async_playwright

# Start Playwright
playwright = await async_playwright().start()
browser = await playwright.chromium.launch(headless=True)
page = await browser.new_page()

# Wait for the results to be visible after form submission
await page.wait_for_selector("table")

# Scrape table rows
rows = await page.query_selector_all("table tr")

data = []
for row in rows:
    cells = await row.query_selector_all("td")
    if len(cells) > 0:
        license_name = await cells[0].inner_text()
        expiration_date = await cells[1].inner_text()
        license_holder = await cells[2].inner_text()

        data.append({
            "License Name": license_name,
            "Expiration Date": expiration_date,
            "License Holder": license_holder
        })

df = pd.DataFrame(data)
df.head()
```

Open 01-Earthquakes.ipynb

Visit <https://bit.ly/ds-dojo-2024> for materials

APIs

structured data from the
internet