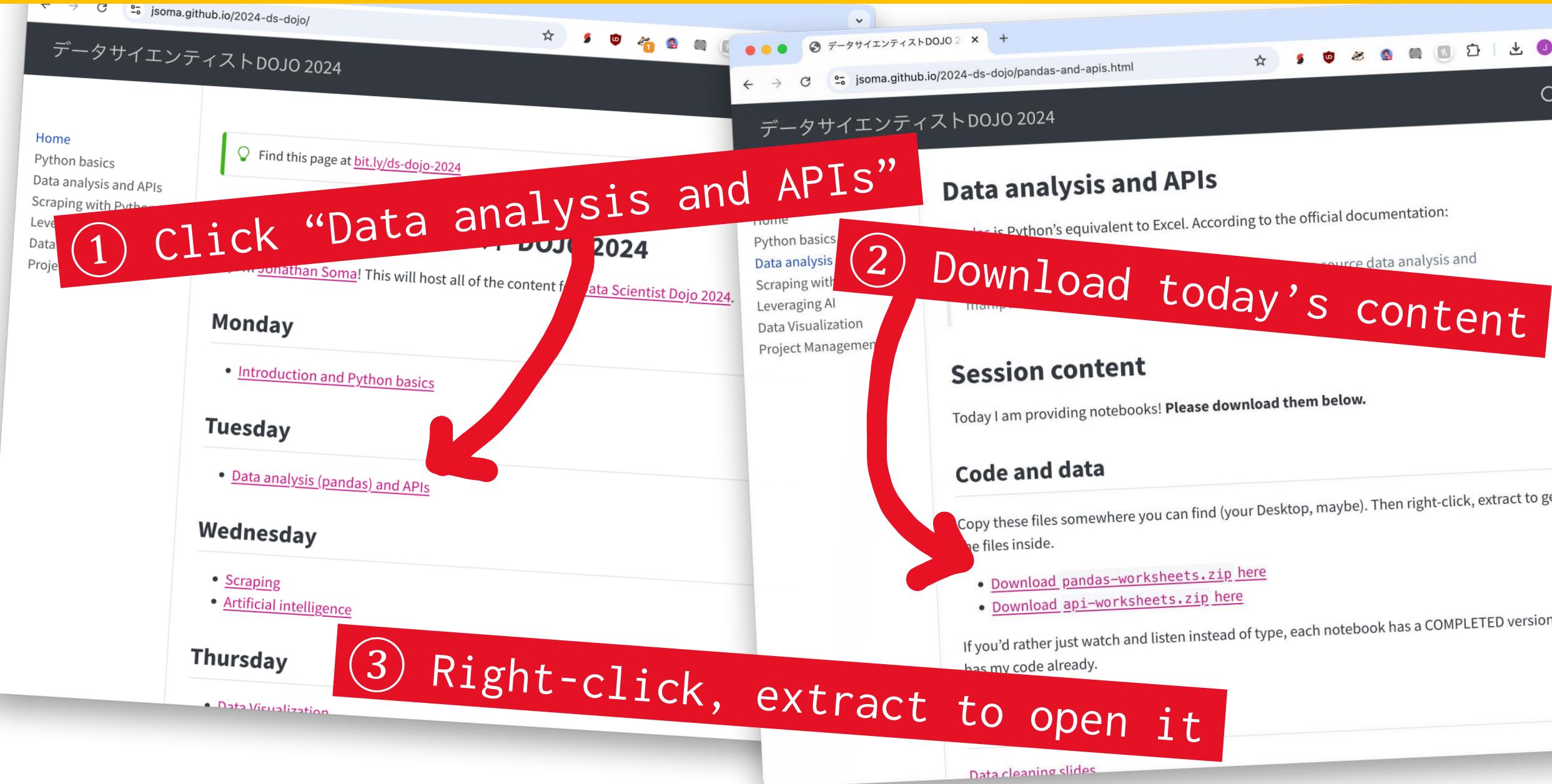


Visit <https://bit.ly/ds-dojo-2024> for materials

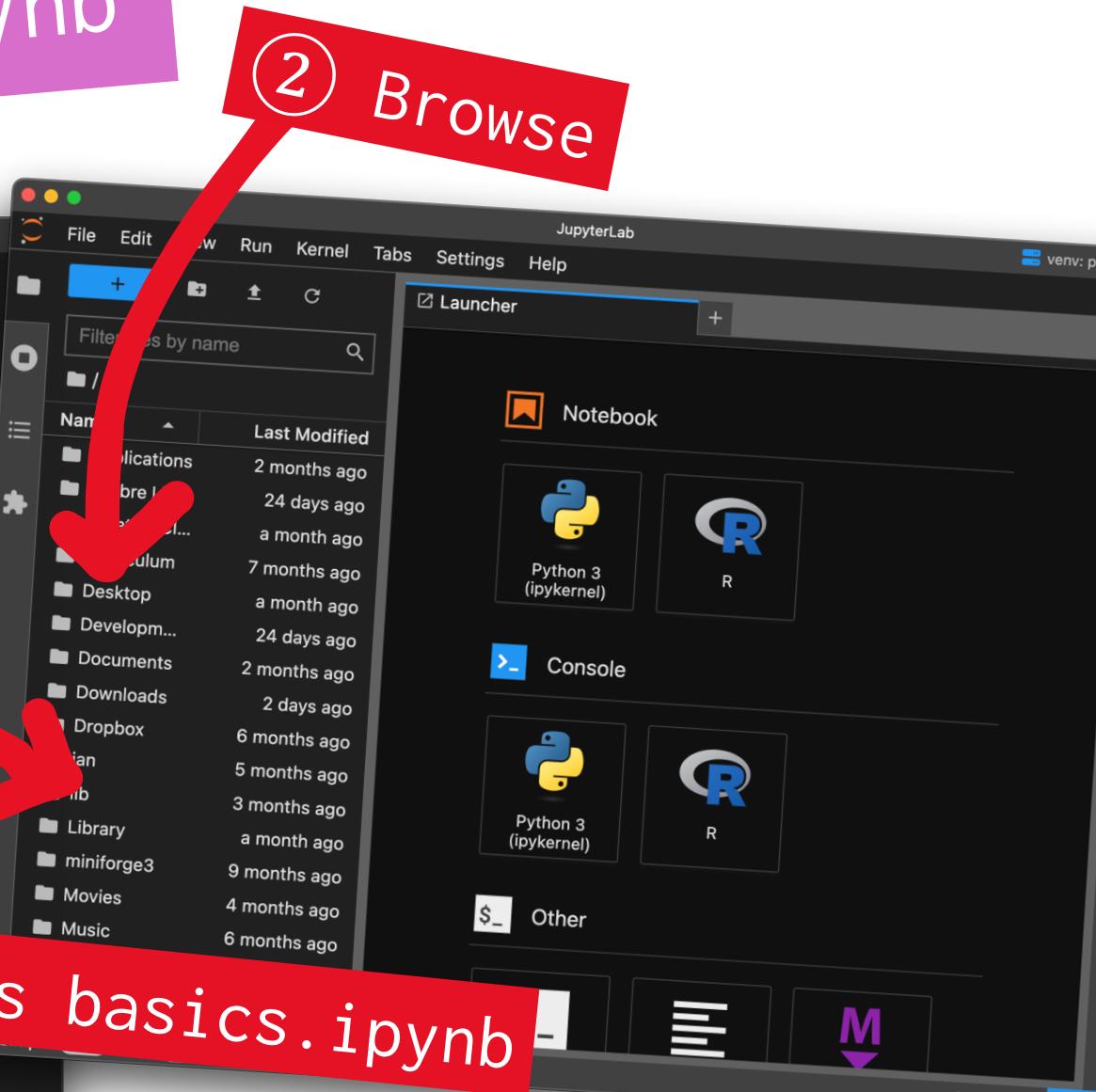
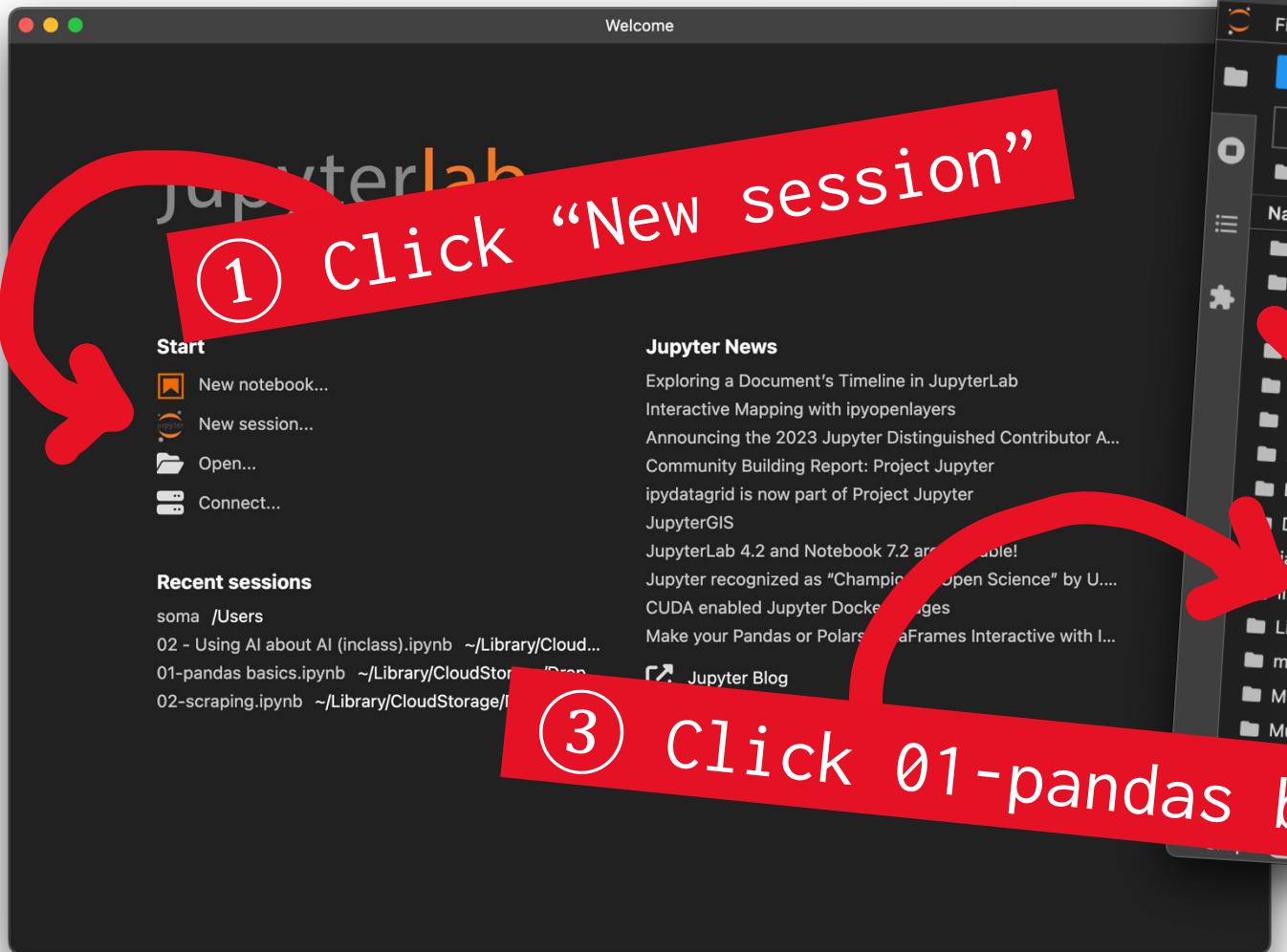
# Structured data

Pandas, CSVs, Excel and APIs

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



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

take the final 12 letters



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

match anything  
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)



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?

Deforestation of the Amazon rainforest[9][10][11]		
Year	Deforested area	Accumulated loss
2005	1,4 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)
	1,11 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)

just ask!  
then TEST IT

< 5/5 >

To extract the "Deforested area" values in km<sup>2</sup> 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 km<sup>2</sup> from each line.



Now let's do a  
little more pandas

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

# APIs

structured data from the  
internet

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

① Click “Data analysis and APIs”

② Download today’s content

Monday

- [Introduction and Python basics](#)

Tuesday

- [Data analysis \(pandas\) and APIs](#)

Wednesday

- [Scraping](#)
- [Artificial intelligence](#)

Thursday

- [Data Visualization](#)

Find this page at [bit.ly/ds-dojo-2024](https://bit.ly/ds-dojo-2024)

① Click “Data analysis and APIs”

② Download today’s content

## Data analysis and APIs

Pandas is Python’s equivalent to Excel. According to the official documentation:

Pandas is fast, powerful, flexible and easy to use open source data analysis and manipulation library. It is built on top of the Python programming language.

### Session content

Today I am providing notebooks! Please download them below.

### Code and data

Copy these files somewhere you can find (your Desktop, maybe). Then right-click, extract to get files inside.

- [Download pandas-worksheets.zip here](#)
- [Download api-worksheets.zip here](#)

If you’d rather just watch and listen instead of type, each notebook has a COMPLETED version with my code already.

*Does anyone have a  
favorite pokemon?*

This is an important question

Google Search x +

google.com/search?q=snorlax&oq=snorlax&gs\_lcrp=EgZjaHJvbWUyDwgAEEUYORiRAhiABBiKBTIMCAEQABhDGIAEGloFMg0IahAAGJECGIAEGloFMgwIAxAA... ...

snorlax

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?

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

In commemoration of Nintendo's 135<sup>th</sup> 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, their remakes, 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  
カビゴン  
*Kabigon*



Snorlax

**Gigantamax Snorlax**

Images on the Bulbagarden Archives

<a href="#">2.5.1 Base stats</a>		<a href="#">Gigantamax Snorlax</a>
<a href="#">2.5.2 Pokéathlon stats</a>		<a href="#">Images on the Bulbagarden Archives</a>
<a href="#">2.6 Type effectiveness</a>		<b>Type</b>
<a href="#">2.7 Learnset</a>		Normal
<a href="#">2.7.1 By leveling up</a>		
<a href="#">2.7.2 By TM</a>		
<a href="#">2.7.3 By breeding</a>		
<a href="#">2.7.4 By a prior Evolution</a>		
<a href="#">2.8 Side game data</a>		
<a href="#">2.9 Form data</a>		
<a href="#">2.9.1 Gigantamax</a>		
<a href="#">2.10 Evolution data</a>		
<a href="#">2.11 Sprites</a>		
<b>3 In animation</b>		
<a href="#">3.1 Main series</a>		
<a href="#">3.1.1 Major appearances</a>		
<a href="#">3.1.1.1 Ash's Snorlax</a>		
<a href="#">3.1.1.2 Other</a>		
<a href="#">3.1.2 Minor appearances</a>		
<a href="#">3.1.3 Pokédex entries</a>		
<a href="#">3.2 Pokémon Origins</a>		
<a href="#">3.3 Pokémon Masters Animated Trailer</a>		
<a href="#">3.4 GOTCHA!</a>		
<a href="#">3.5 Pokémon Evolutions</a>		
<a href="#">3.6 The Adventures of Snorlax &amp; Cubone</a>		
<b>4 In the manga</b>		
<a href="#">4.1 Ash &amp; Pikachu</a>		
<a href="#">4.2 The Electric Tale of Pikachu</a>		
<a href="#">4.3 Pokémon Adventures</a>		
<a href="#">4.4 Pokémon Battle Frontier</a>		
<a href="#">4.5 Pokémon: Yeah! I Got Pokémon!</a>		
<a href="#">4.6 Pokémon Mystery Dungeon: Ginji's Rescue Team</a>		
<a href="#">4.7 Pokémon Journeys</a>		
<a href="#">4.8 Pokémon Pocket Monsters</a>		
<a href="#">4.9 Pokémon Ruby-Sapphire</a>		
<a href="#">4.10 Snorlax's Dream Gourmet</a>		

# 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

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**

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"  
      }  
    }  
  ]  
}
```

A screenshot of a web browser displaying the PokéAPI homepage. The page has a dark header with a pink bar containing the word "APIS". Below the header is a large red arrow pointing from a red button labeled "Structured data" to the central content area. The main content features the PokéAPI logo and the text "The RESTful Pokémon API" and "Serving over 2.5 billion API calls each month!". A green button below the text says "Check out the docs!". At the bottom, there's a search bar with the URL "https://pokeapi.co/api/v2/" and the search term "pokemon/ditto", followed by a "Submit" button.

# APIS

PokéAPI

# APIS

Home About API v2 GraphQL v1beta

**PokéAPI**

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

# jupyterlab

The screenshot shows the JupyterLab interface. On the left, there's a sidebar with sections for "Start" and "Recent sessions". The "Start" section includes options like "New notebook...", "New session...", "Open...", and "Connect...". The "Recent sessions" section lists three notebooks: "02 - Using AI about AI (inclass).ipynb", "01-pandas basics.ipynb", and "02-scraping.ipynb". To the right of the sidebar is a main content area titled "doc - JupyterLab". It contains a code editor with two cells of Python code. The first cell uses the `playwright` library to navigate to a URL and scrape data from a table. The second cell imports `pandas` and processes the scraped data. The top right of the window shows a "venv: python" indicator.

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

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

# Navigate to the URL
await page.goto("https://www.tdlr.texas.gov/LicenseSearch/")

[ ]: import pandas as pd

# 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 JupyterLab

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

# APIs

structured data from the  
internet