

GakuNin RDMとChatGPTを使ってプログラミングを手軽に業務に活用する一試案

芦北, 卓也
九州大学附属図書館

<https://hdl.handle.net/2324/6796164>

出版情報 : 2023-09-02
バージョン :
権利関係 : Creative Commons Attribution 4.0 International

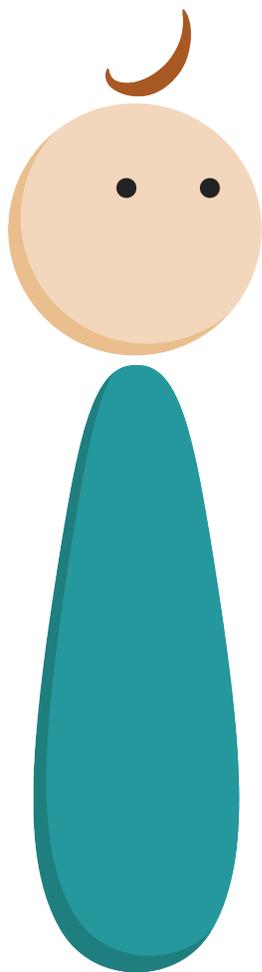
GakuNin RDMとChatGPTを使って プログラミングを手軽に業務に活用する 一試案

Code4Lib JAPAN Conference 2023
2023年9月2日(土)

芦北 卓也

<https://orcid.org/0000-0003-1853-7108>

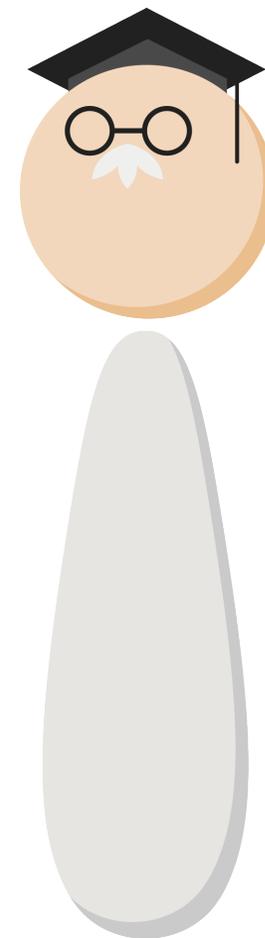




医学研究院のCCライセンスつきオープンアクセス論文を
どうか機関リポジトリに登録してください

図書館で勝手に登録してくれるならいいよ
APIでデータを取得すれば簡単でしょ

API.....



九州大学で 2023年7月に NIIへの申請により GakuNin RDMの データ解析機能が 利用可能に

<https://rds.dx.kyushu-u.ac.jp/news/1012>

The screenshot shows the GakuNin RDM website interface. At the top, there is a blue header with the Kyushu University logo and the text "Research Data Service 研究データ管理支援". A navigation menu includes "Services", "Guides", "System", and "About". The main content area features a "Recent Posts" section with a list of articles. The highlighted article is titled "GakuNin RDMでデータ解析が可能に" (Data analysis is possible with GakuNin RDM) and is dated 2023.07.13. The article text states that the university participated in a proof-of-concept experiment for data analysis functionality, allowing users to run Python or R programs online. Below the text is a screenshot of a JupyterLab interface showing a code cell with the command `print("Hello, world")` and its output, "Hello, world".

九州大学
KYUSHU UNIVERSITY

Research Data Service
研究データ管理支援

日本語 | English

Services ▾ Guides ▾ System ▾ About ▾

Recent Posts

- 部局等における研究データ管理・公開実施要領作成のための「ガイドライン」と「サンプル」を作成しました
- GakuNin RDMでデータ解析が可能に
- 国際シンポジウム「大学における研究データ管理の意義と支援人材育成」の動画と資料を公開
- 国際シンポジウム「大学における研究データ管理の意義と支援人材育成」(2023年6月19日(月))のご案内
- 国際シンポジウム「大学における研究データ管理の意義と支援人材育成」(2023年6月19日(月))開催のお知らせ

GakuNin RDMでデータ解析が可能に

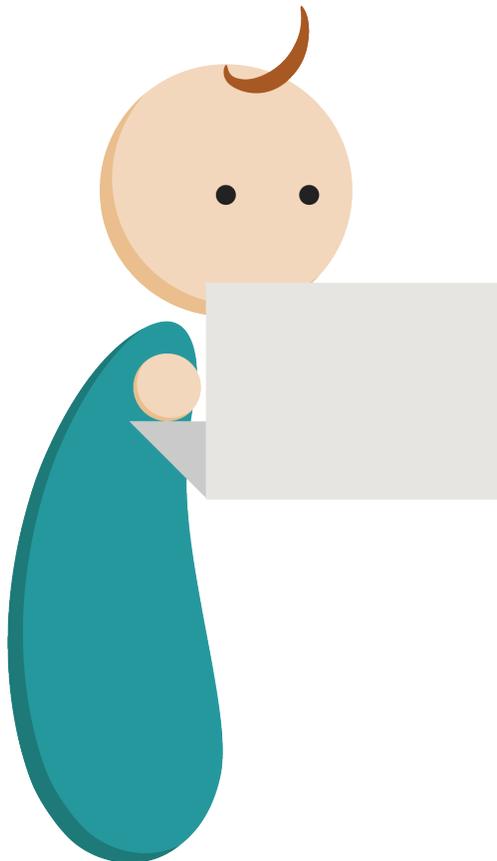
2023.07.13

九州大学では国立情報学研究所(NII)によるGakuNin RDMデータ解析機能の実証実験に参加し、その試用版を利用できるようになりました。オンライン上に解析環境を構築し、PythonやRのプログラムを開発・実行できます。

```
[2]: print("Hello, world")  
Hello, world
```

事前にGakuNin RDMへのログインとプロジェクトの作成が必要です。

Chat GPTに Crossref APIの 使い方を教わる



Crossref APIを使って論文のDOIからタイトルと出版年を取得するPythonのコードを教えてください。



Crossref APIを使用して論文のDOIからタイトルと出版年を取得するPythonコードを以下に示します。このコードでは、Requestsライブラリを使用してAPIリクエストを行います。



まず、Requestsライブラリをインストールします（もし未インストールの場合）：

```
bash
```

Copy code

```
pip install requests
```

次に、以下のPythonコードを実行して、論文のDOIからタイトルと出版年を取得します：

```
python
```

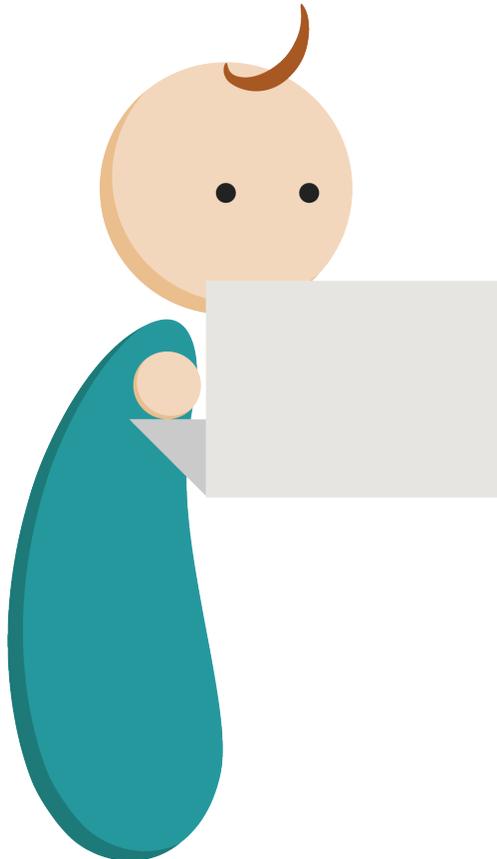
Copy code

```
import requests

def get_paper_info_from_doi(doi):
    base_url = "https://api.crossref.org/works/"
    url = f"{base_url}{doi}"

    try:
        response = requests.get(url)
```

Chat GPTに Unpaywall API の使い方を教わる



Unpaywallで特定のDOIから該当論文の"is_oa"及び"oa_status"を取得するPythonコード



Unpaywallは、オープンアクセスの論文情報を提供するAPIです。特定のDOIから該当論文の"is_oa"（オープンアクセスであるかどうか）と"oa_status"（オープンアクセスの状態）を取得するPythonコードを以下に示します。このコードを実行するには、リクエストを送信するための`requests`ライブラリが必要です。



python

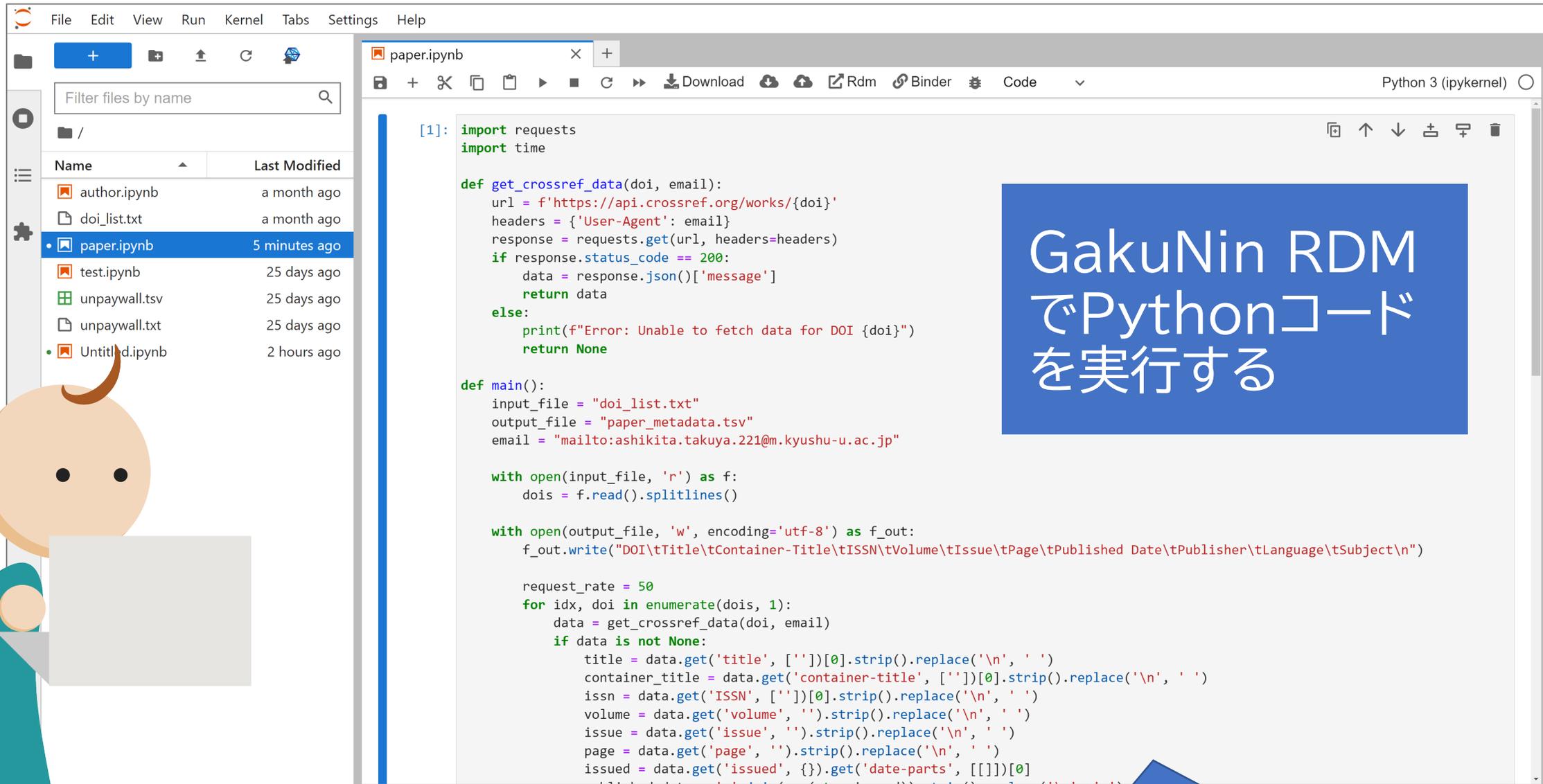
Copy code

```
import requests

def get_unpaywall_data(doi):
    base_url = "https://api.unpaywall.org/v2/"
    endpoint = f"resolve/{doi}"
    api_key = "YOUR_UNPAYWALL_API_KEY" # UnpaywallのAPIキーを入力してください

    headers = {"Authorization": f"Bearer {api_key}"}
    response = requests.get(base_url + endpoint, headers=headers)

    if response.status_code == 200:
        data = response.json()
        is_oa = data.get("is_oa", False)
        oa_status = data.get("oa_status", "unknown")
        return is_oa, oa_status
    else:
```



File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name	Last Modified
author.ipynb	a month ago
doi_list.txt	a month ago
paper.ipynb	5 minutes ago
test.ipynb	25 days ago
unpaywall.tsv	25 days ago
unpaywall.txt	25 days ago
Untitled.ipynb	2 hours ago

```
[1]: import requests
import time

def get_crossref_data(doi, email):
    url = f'https://api.crossref.org/works/{doi}'
    headers = {'User-Agent': email}
    response = requests.get(url, headers=headers)
    if response.status_code == 200:
        data = response.json()['message']
        return data
    else:
        print(f"Error: Unable to fetch data for DOI {doi}")
        return None

def main():
    input_file = "doi_list.txt"
    output_file = "paper_metadata.tsv"
    email = "mailto:ashikita.takuya.221@m.kyushu-u.ac.jp"

    with open(input_file, 'r') as f:
        dois = f.read().splitlines()

    with open(output_file, 'w', encoding='utf-8') as f_out:
        f_out.write("DOI\tTitle\tContainer-Title\tISSN\tVolume\tIssue\tPage\tPublished Date\tPublisher\tLanguage\tSubject\n")

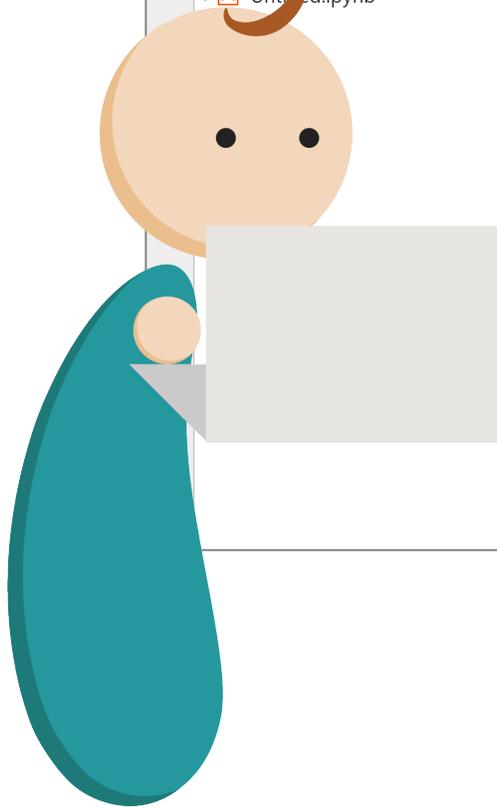
    request_rate = 50
    for idx, doi in enumerate(dois, 1):
        data = get_crossref_data(doi, email)
        if data is not None:
            title = data.get('title', []).strip().replace('\n', ' ')
            container_title = data.get('container-title', []).strip().replace('\n', ' ')
            issn = data.get('ISSN', []).strip().replace('\n', ' ')
            volume = data.get('volume', '').strip().replace('\n', ' ')
            issue = data.get('issue', '').strip().replace('\n', ' ')
            page = data.get('page', '').strip().replace('\n', ' ')
            issued = data.get('issued', {}).get('date-parts', [[]])[0]
```

GakuNin RDM でPythonコード を実行する

セルの枠内にコードを貼り付けて
上の▶ボタンを押すか、Ctrl-Enter
またはShift-Enterで実行

The screenshot shows a JupyterLab environment. On the left is a file browser with a search bar and a list of files including 'paper_metadata.tsv'. The main area displays a table of paper metadata with columns for DOI, Title, Container-Title, ISSN, Volume, Issue, Page, Published Date, and Publisher. A blue box in the bottom right contains the text: 実際に論文のメタデータが取得できる

	DOI	Title	Container-Title	ISSN	Volume	Issue	Page	Published Date	Publisher
1	10.1016/j.ijregi.20...	Seroepidemiological...	IJID Regions	2772-7076	2		126-129	2022-3	Elsevier BV
2	10.1093/mr/roac030	Antibody response t...	Modern Rheumatol...	1439-7595	33	2	367-372	2022-3-29	Oxford University Pres...
3	10.1186/s12884-02...	Lack of catch-up in ...	BMC Pregnancy and...	1471-2393	22	1		2022-3-12	Springer Science and B...
4	10.2169/internalme...	Cerebral Tuberculo...	Internal Medicine	0918-2918	61	19	2941-2...	2022-10-1	Japanese Society of In...
5	10.21873/anticanre...	Neuroectodermal R...	Anticancer Research	0250-7005	42	9	4337-4...	2022-8-23	Anticancer Research U...
6	10.9758/cpn.2022...	Prescription Pattern...	Clinical Psychophar...	1738-1088	20	1	61-69	2022-2-28	Korean College of Neur...
7	10.1111/liv.15085	On-treatment gam...	Liver International	1478-3223	42	1	59-68	2021-11-29	Wiley
8	10.1038/s41430-02...	Critical vitamin defi...	European Journal of...	0954-3007	76	11	1618-1...	2022-6-10	Springer Science and B...
9	10.1002/cam4.4939	Distribution and fav...	Cancer Medicine	2045-7634	12	1	49-60	2022-6-13	Wiley
10	10.1016/j.ajhg.202...	A multi-layer functi...	The American Journ...	0002-9297	109	8	1366-1...	2022-8	Elsevier BV



ChatGPTを使うメリット

- プログラミングのスキルがなくてもコードが書ける
- 1回で正常に実行できるとは限らないが、具体的な仕様を細かく要求すれば成功率が高まる
- 実行エラーが発生してもそのことを指摘すればコードを改善してもらえる
- 理解できなかった部分について掘り下げて質問すれば詳しく説明してもらえる
- 成功するまで何度でも対話を繰り返すことができ、試行錯誤していくうちにだんだんプログラミングの要領がつかめてくる

API利用の注意点



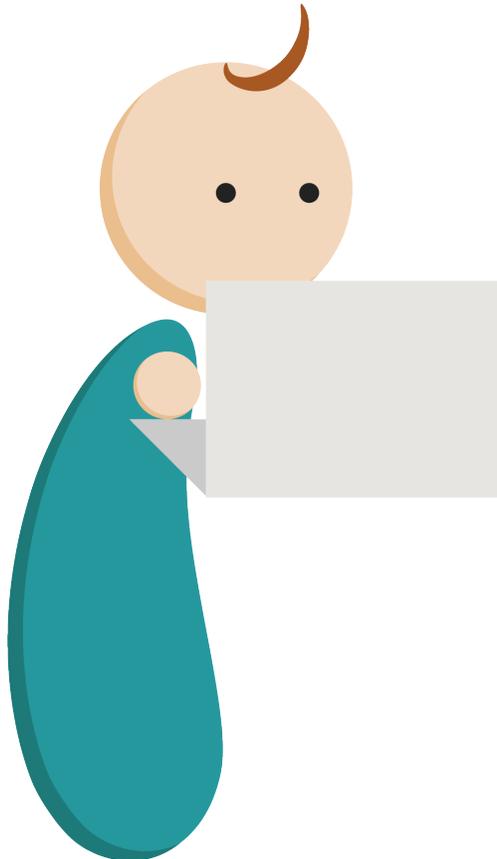
公開されているAPIを利用する上で気を付けることはありますか。

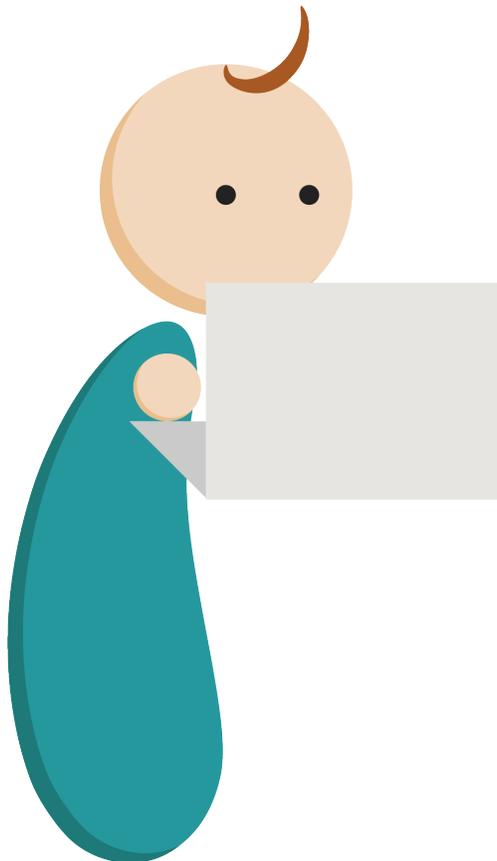


公開されているAPIを利用する際にはいくつか注意すべきポイントがあります。以下にいくつか重要な点を挙げてみましょう。



1. **利用規約の確認:** APIプロバイダーが提供する利用規約を必ず確認しましょう。利用規約には、APIの使用に関する制約、制限、課金ポリシーなどが記載されています。
2. **アクセス制限と認証:** APIのアクセスには適切な認証が必要です。APIキー、トークン、OAuth認証などのメカニズムを使用してアクセス制御を行います。認証情報を適切に保護し、不正なアクセスを防ぐようにしましょう。
3. **利用制限:** APIプロバイダーは通常、利用制限を設けています。1分あたりのリクエスト数や1日あたりのトータルリクエスト数などが制限されている場合があります。自身のアプリケーションの予想されるトラフィックに合わせて適切に計画を立てましょう。
4. **エラーハンドリング:** APIから返されるエラーコードやエラーメッセージを適切にハンドリングすることが重要です。エラーが発生した場合にユーザーに適切なメッセージを表示し、問題解決をサポートする仕組みを構築しましょう。
5. **データセキュリティ:** APIを介して送信されるデータは暗号化されるべきです。また、APIプロバイダーが提供するセキュリティ対策についても確認し、適切な対策を講じることが重要です。
6. **エンドポイントの変更:** APIはバージョンアップや仕様変更が行われることがあります。自身のアプリケーションが依存しているAPIエンドポイントが変更される可能性に備え、柔軟性を持たせる設計を心掛けましょう。
7. **データの利用許諾:** APIを通じて取得したデータを自身のアプリケーションで利用する場合、そのデータの利用許諾に注意が必要です。著作権やライセンスに関する事項を遵守しましょう。
8. **モニタリングとメンテナンス:** APIの利用状況をモニタリングし、異常なトラフィックやエラーを追跡できるよう





以下の手順で論文の書誌情報を取得するPythonコードを教えてください。



その際、クエリに「mailto:hogehoge@example.org」パラメータを含めるか、または、User-Agentヘッダーに「mailto:hogehoge@example.org」を含めます。

また、1秒間に50回のリクエストレートとしてください。

1. 1行に1つDOIが記録されている"doi_list.txt"という名称のテキストファイルを読み込む。
2. Crossref APIから以下の書誌情報を取得する。クエリに「mailto:」パラメータを含めます。
 - ・タイトル(title)
 - ・ジャーナル名(container-title)
 - ・ISSN
 - ・巻(volume)
 - ・号(issue)
 - ・ページ(page)
 - ・発行年(issued) YYYY-MM-DD形式で
 - ・出版社(publisher)
 - ・件名(subject)
3. 取得した情報から改行コードを除外する。また、前後の空白文字も除外する(トリム)。
4. 取得した情報を"paper_matadata.tsv"という名称のタブ区切りのテキストファイルに出力する。

細かな要求の例



以下は、Crossref APIを使用して論文の書誌情報を取得し、指定された形式で出力するPythonコードの例です。このコードでは、User-Agentヘッダーに「mailto:hogehoge@example.org」を含め、1秒間に50回のリクエストレートを制御します。また、取得した情報から改行コードと前後の空白文字を除外しています。

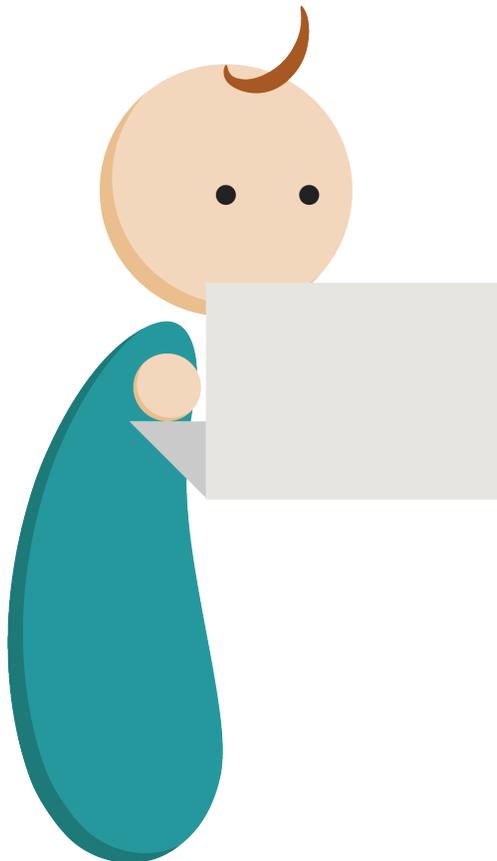


python

Copy code

```
import requests
import time

def get_crossref_data(doi, email):
    url = f'https://api.crossref.org/works/{doi}'
    headers = {'User-Agent': email}
```



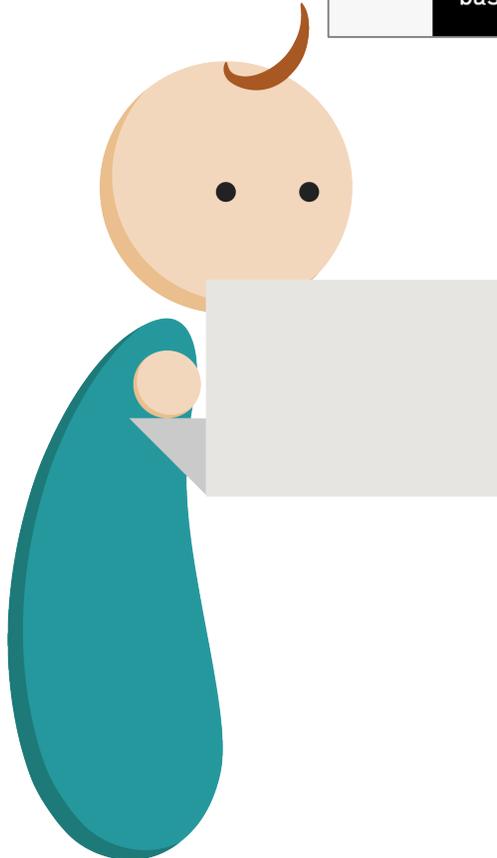
 実行すると、"NameError: name 'csv' is not defined"というエラーが発生しました。

 申し訳ありません、`csv`モジュールをインポートし忘れていたようです。エラーを修正するために、以下のようにコードを修正します：

```
python Copy code  
  
from bs4 import BeautifulSoup  
import pandas as pd  
import csv # csvモジュールをインポートする  
  
def html_tables_to_csv(html_file):  
    # HTMLファイルを開く  
    with open(html_file, 'r', encoding='utf-8') as file:  
        html_data = file.read()  
  
    # BeautifulSoupを使ってHTMLを解析  
    soup = BeautifulSoup(html_data, 'html.parser')  
  
    # テーブルをCSVに変換して保存するためのリスト  
    csv_data = []
```

発生したエラーの修正

試行錯誤の例



Crossref APIを使ってISSNのリストから学術雑誌のタイトル、PISSN、EISSNを取得するPythonコード

Crossref APIを使用してISSNのリストから学術雑誌のタイトル、Print ISSN (PISSN)、Electronic ISSN (EISSN)を取得するPythonコードを以下に示します。

```
python
import
# Cross
base_u
```

Crossref APIを使ってISSNのリストから学術雑誌のタイトルとissn-typeを取得するPythonコードを教えてください。

issn-typeは複数ある場合があります、次の形式で提供されます。

```
"issn-type":[{"value":"2410-4477","type":"print"},{"value":"2305-3925","type":"electronic"}]
```

Crossref APIを使用してISSNのリストから学術雑誌のタイトルとissn-typeを取得するPythonコードを以下に示します。

```
python
import
# Cross
base_u
```

issn-typeはリストで提供されることがあります。例は以下の通りです。

```
"issn-type":[{"value":"2410-4477","type":"print"},{"value":"2305-3925","type":"electronic"}]
```

申し訳ありません、誤解がありました。ISSNタイプがリストで提供される場合のコードを以下に示します。

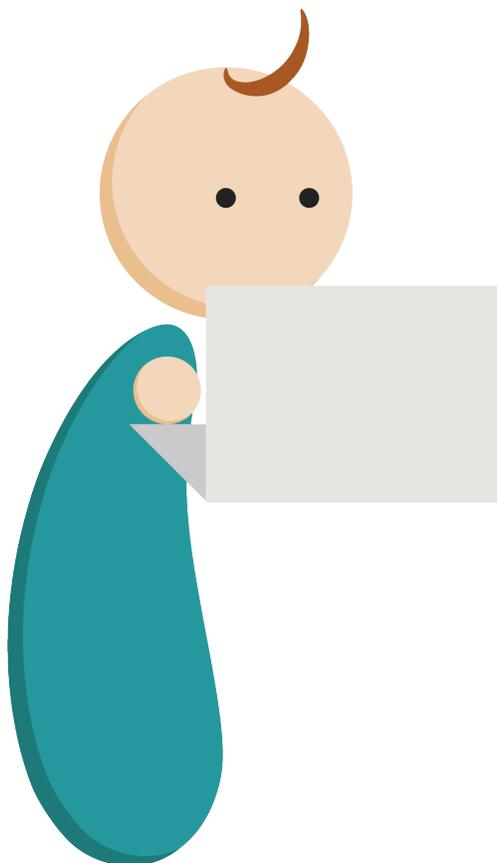
```
python
import
# Cross
base_u
# ISSN
issn_li
def get
url
```

今回はISSNのリストを"issn_list.txt"から取得してください。"issn_list.txt"はISSNが改行で区切られています。また、出力先をTSVファイル"output.tsv"にしてください。

ISSNのリストを"issn_list.txt"から読み取り、結果をTSVファイル"output.tsv"に出力するPythonコードを以下に示します。

```
python
```

コードの理解できなかった部分について掘り下げて質問



以下の部分についてもう少し詳しく教えてください。
例えば日本の場合、'iso_a3'はJPN、'ISO_Code'はJAと対応づけられますか。

データと地図データの結合

```
merged_data = world.merge(data, left_on='iso_a3', right_on='ISO_Code', how='left')
```



もちろん、ご質問いただいた内容に関して詳しく説明いたします。



GeoPandasの`merge`関数は、2つのデータフレームを結合する際に使用されます。引数として指定された`left_on`と`right_on`は、それぞれ結合されるデータフレームのカラムを指定します。`how`引数は結合方法を指定します。この場合、左側（`world`）のデータフレームの`iso_a3`カラムと、右側（`data`）のデータフレームの`ISO_Code`カラムを基準に結合を行っています。

例えば、日本の場合、`iso_a3`が`'JPN'`であり、`ISO_Code`が`'JA'`のように異なるコードで表されていると仮定します。

コード例の結合部分は以下ようになります。

```
python
```

Copy code

```
merged_data = world.merge(data, left_on='iso_a3', right_on='ISO_Code', how='left')
```

このコードでは、`iso_a3`カラムと`ISO_Code`カラムの値を比較し、一致する場合にデータが結合されま

GakuNin RDMを使うメリット

- オンライン上でPythonとRが使える
 - JupyterLab と RStudio が利用可能
 - 業務用パソコンの構成を変更せずに済む
- プロジェクトに同僚の職員をメンバー登録すれば同一の環境で業務の分担やデータ共有が可能
 - 学認の認証システムを使って、所属機関のアカウントのID、パスワードでログイン可能
- その気になれば、他機関の職員とも協働可能
 - 所属機関が規約に同意
 - 学認の認証システムを使って、所属機関がユーザ認証

GakuNin RDMを使うメリット

- GakuNin RDMの標準ストレージを利用可能(1人100GB)
退職してアカウントが削除されてもデータは残存
 - アカウントを削除されると本人はデータにアクセスできなくなるが、後任者・管理者を当該プロジェクトメンバーに加えて引継ぎ可能
- Google DriveやOneDrive、Dropbox等、アドオン接続した外部ストレージを連携可能
 - 分散したデータをGakuNin RDMに集約

TIPS

```
# ホームディレクトリに共有ストレージのシンボリックリンクを作成
! ln -s /mnt/user ~
# ホームディレクトリに追加ストレージのシンボリックリンクを作成
! ln -s /mnt/rdm/osfstorage ~
! ln -s /mnt/rdm/googledrive ~
! ln -s /mnt/rdm/onedrive ~
```

- シンボリックリンクにより追加ストレージ上のJupyter Notebookが実行可能になる。
- 解析環境は30日以上使用しないと自動的に削除されるため、作成したhogehoge.ipynbを解析環境ごと削除されないようにするには共有ストレージや追加ストレージに移動しておく。

GakuNin RDMでデータ解析機能を使うには



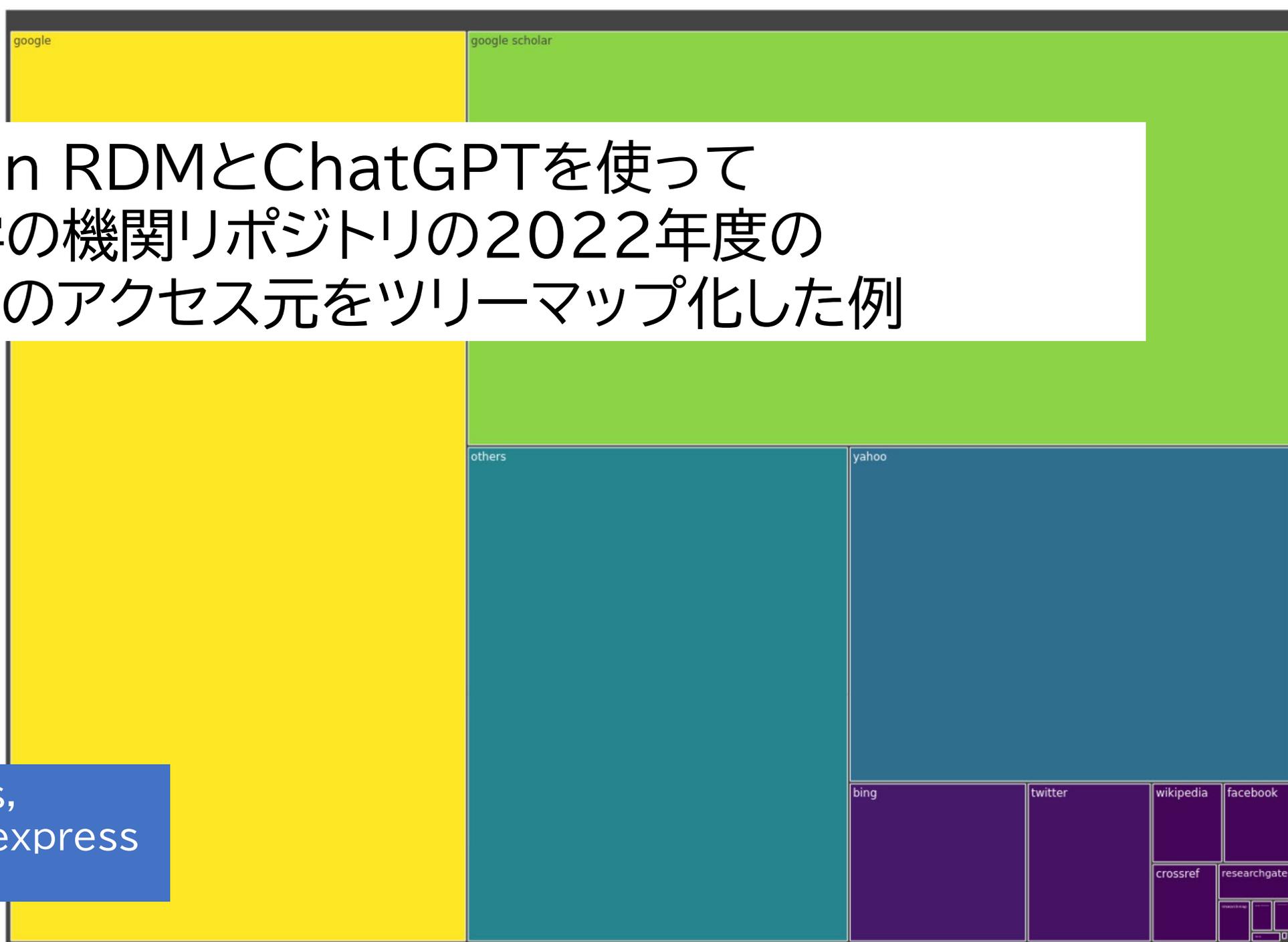
The screenshot shows the 'Add-ons' menu in GakuNin RDM. The 'Add-ons' menu item is circled in orange. Below it, the 'GakuNin Federated Computing Services (Jupyter)' add-on is selected and circled in orange, with a '有効にする' (Enable) button next to it. A blue arrow points from this '有効にする' button towards the right-hand screenshot.



The screenshot shows the 'Analysis' menu in GakuNin RDM. The 'Analysis' menu item is circled in orange. Below it, the '新しい解析環境' (New Analysis Environment) section is visible. A blue box highlights the '追加パッケージ' (Additional Packages) section, which lists 'apt-get', 'conda', 'pip', and 'R (MRAN)' with '+追加' (Add) buttons. To the right of this box, the text 'matplotlib, scikit-learn, pandas等、任意のパッケージを個別に追加可能' (matplotlib, scikit-learn, pandas, etc., any packages can be added individually) is displayed. At the bottom, the '新しい解析環境を作成' (Create New Analysis Environment) button is circled in orange, with the URL 'https://binder.cs.rcos.nii.ac.jp' next to it.

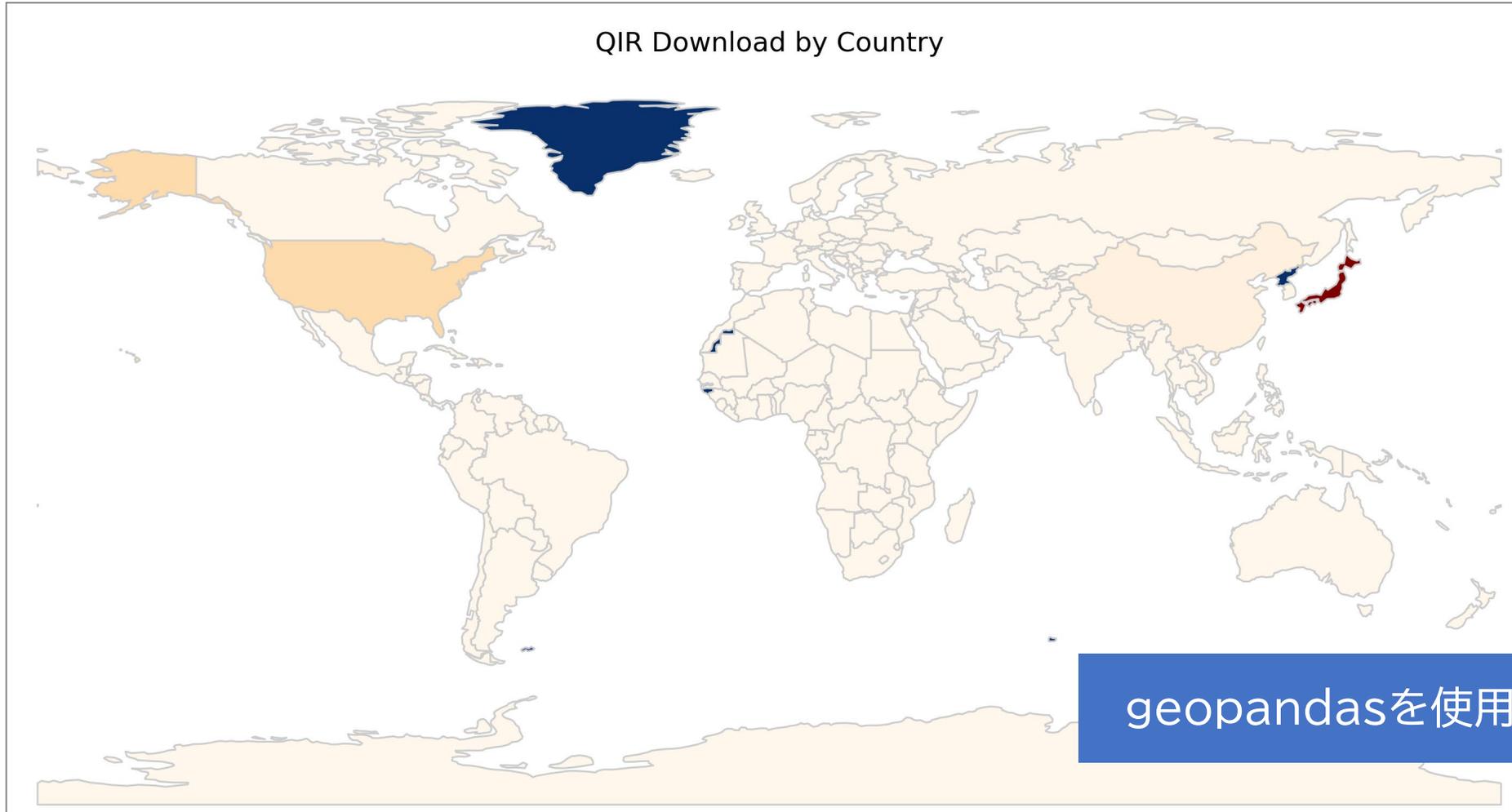
プロジェクトを作成後、
①アドオンメニューでデータ解析機能(jupyter)を有効にして
②解析メニューで「新しい解析環境を作成」ボタンをクリックするだけ

GakuNin RDMとChatGPTを使って 九州大学の機関リポジトリの2022年度の 外部からのアクセス元をツリーマップ化した例



pandas,
plotly.express
を使用

GakuNin RDMとChatGPTを使って 九州大学の機関リポジトリの2022年度の国・地域別 ダウンロード数を世界地図にプロットした例



Excelを使った例

教員が自由な形式で
DOIを入力して困る



ChatGPTに
Excel VBAの
コーディングをさせて
DOIを正規化



ExcelのアクティブシートのY列の2行目以降のセルに対して以下の処理を順番に実行するVBAコードを教えてください。大文字・小文字を区別しません。正規表現を用いてよいです。

1. セル内改行を削除
2. セル内の全角文字を半角文字に変換
3. 行頭及び行末のスペースを削除
4. 行頭の"doi:"および"doi: "および"doi "を大文字・小文字を区別せずに削除
5. 行末の"."を削除
6. 行頭の"https://doi.org/"および"http://doi.org/"および"https://dx.doi.org/"および"http://dx.doi.org/"および"doi.org/"および"org/"を削除
7. 値に"/"が含まれないセルを空値にする



以下のVBAコードは、正規表現を使用して指定された処理をアクティブシートのY列の2行目以降のセルに対して順番に実行します。このコードでは、`RegExp`オブジェクトを使用して正規表現検索と置換を行います。



vb

Copy code

```
Sub ProcessYColumn()  
    Dim ws As Worksheet  
    Dim lastRow As Long  
    Dim i As Long  
    Dim cellValue As String  
    Dim regEx As Object
```

Excelを使った例

論文のメタデータを
共著者ごとに複製
したい



ChatGPTに
Excel VBAの
コーディングをさせて
該当行を人数分コピー

著者のIDが入った
特定セルのみ分割



以下の処理を実行するVBAコードを教えてください。

1. ExcelのアクティブシートのF列のセルに"&&"で結合された複数の文字列がセットされている場合、以下の処理を実行します。
 - 2-1. 該当する行のF列を除く各セルの内容を下の行にコピーします。
 - 2-2. コピーする行数は、"&&"で結合された複数の文字列のカウント数とします。
 - 2-3. コピーする行のF列には、"&&"で結合された各々の文字列をセットします。
3. 以上の処理をアクティブシートの最初から最後の行まで繰り返します。



以下は、ExcelのVBAコードで要件に従って処理を行うサンプルコードです。



vba

Copy code

```
Sub ProcessRowsWithAmpersands()  
    Dim lastRow As Long  
    Dim i As Long, j As Long  
    Dim strArr() As String  
  
    ' アクティブシートの最終行を取得  
    lastRow = ActiveSheet.Cells(Rows.Count, "F").End(xlUp).Row  
  
    ' 各行の処理  
    For i = 1 To lastRow  
        ' F列の値を取得し、"&&"で分割  
        strArr = Split(CStr(ActiveSheet.Cells(i, "F").Value), "&&")
```

ChatGPTにCode4Lib カンファレンス応募に必要な 発表概要を添削させた例



以下の文章は、あるカンファレンスに発表応募する発表概要の案です。400字以内の制限があります。添削してください。

九州大学ではGakuNin RDMのデータ解析機能実証実験に参加し、同機能の試用版を使えるようになった。研究者ではない図書館の職員でも、業務用のパソコンに特別な環境構築をすることなく、クラウド上に仮想環境を構築し、Webブラウザ上でPythonやRのプログラムを開発・実行できる。

併せて最近話題のChatGPTを活用すれば、Crossref APIやUnpaywall APIを使ったデータ取得が簡単に実現できるとわかった。プログラミングの初心者でも、ChatGPTにやりたいことを簡潔に伝えるだけで実行可能なコードがわかり、実務に役立てることができる。

本報告では、PythonやRの可能性に関心はあったもののスキルとして身に付けられてはなかった大学図書館職員が、GakuNin RDMとChatGPTを併用することにより、プログラミングを誰でも手軽に業務に活用できるという例を示す。



以下は、カンファレンスに発表応募するための発表概要の案です。文字数制限を考慮し、より簡潔かつ明確にまとめました。



タイトル: GakuNin RDMとChatGPTを活用した大学図書館職員のための手軽なプログラミング活用法

要約:

九州大学では、GakuNin RDMのデータ解析機能実証実験に参加し、その試用版を利用できるようになりました。研究者ではない図書館の職員でも、特別な環境構築なしにクラウド上で仮想環境を作成し、Webブラウザ上でPythonやRのプログラムを開発・実行できることがわかりました。

さらに、話題のChatGPTを利用することで、Crossref APIやUnpaywall APIを用いたデータ取得が容易になりました。プログラミング初心者でも、ChatGPTにやりたいことを簡潔に伝えるだけで、実行可能なコードを生成し、実務に役立てることができます。

本報告では、プログラミングスキルを持たない大学図書館職員がGakuNin RDMとChatGPTの組み合わせを活用し、手軽にプログラミングを業務に活かす事例を示します。PythonやRの潜在的な可能性に関心を持っていたが、スキルを身につけることができなかった職員にとって、本手法は有望なアプローチとなるでしょう。

課題

- ChatGPTに書かせたコードを全面的に信用してよいか
 - 実行しても気づかないバグ、脆弱性の存在
- コードの仕様変更に対応できるか
 - 人事異動後の属人的サポートは回避したい
- ChatGPTやGakuNin RDMのサービス継続性
 - いつまで無料で使えるか

まとめ

- プログラミングに敷居の高さを感じる人でも GakuNin RDMとChatGPTを活用することで **手軽に**クラウド上の実行環境やAIによる支援が得られる
- とくに若手職員、若手研究者におすすめ
- 個人がプログラミングに親しみをもつことで、組織全体の業務DXを喚起