

## 業務効率化⑤「API 活用して重要情報を収集・分析しよう！」

最初に API ってなんぞ？

API=Application Programming Interface の略で、企業や公共団体等が提供するサービス内の情報や機能を使えるようにしてくれる、いわば一歩間違えば違法となるスクレイピングの許可版と思えば良き！ただ、API もタダで使用できるものお金がかかるもの、企業が承認した人・企業だけが使用できると様々ある。

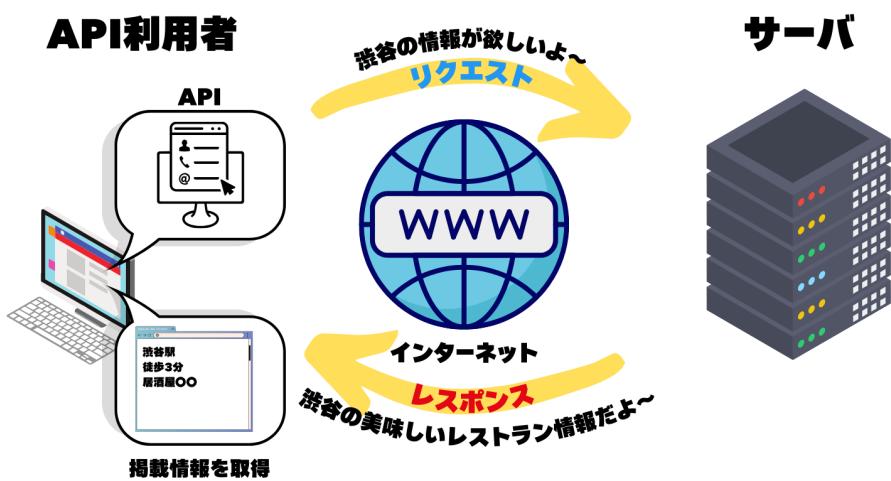
- ・スクレイピング=グレー寄りのグレー

※ 使用するエンジニア次第

- ・API=白寄りのグレー

※ ほぼ大丈夫

スクレイピングは自分で調査して情報を取得してくるってことは②でやったね、今回の API はサーバーに蓄積されている情報に対してリクエストを送るとレスポンスで返してくれるんだよ。



API には、LINE が提供している API 「LINE Notify」 という LINE アカウントから通知を送る仕組みったり、ホットペッパーでは「グルメサーチ API」 で掲載されている場所の方をほぼ取得できたりと、楽に実装できるんだよ！

では、今回は何の API 使うだと、Web API を使用する。

Web API とは、HTTP/HTTPS で「Web=インターネットを介して使用する API」ってことだ!

Web API の他にはマイクロソフトが独自 API として定めた Win32 とかあるね。

API 活用には 3 つのメリットがあり、

- ① 開発の効率化・コスト削減
- ② セキュリティ・顧客満足度の向上
- ③ システム変更による影響を最小化

- ① は、もちろんゼロイチじや無いから時間とコストが削減できる。また、特定の領域に特化した企業や団体が API を提供してくれているから、クオリティが高いんだよ。
- ② は、自前でデータや情報を保有しないからセキュリティ面が向上されるし、低コストで API を使用できるため価格を抑えサービス提供が可能になる。
- ③ は、何かのきっかけで仕様が変更されたりするとコードの大部分を修正する必要が出てくるけど、API から情報を取得しているので、仕様が変更されても修正量が劇的に少なくなる。

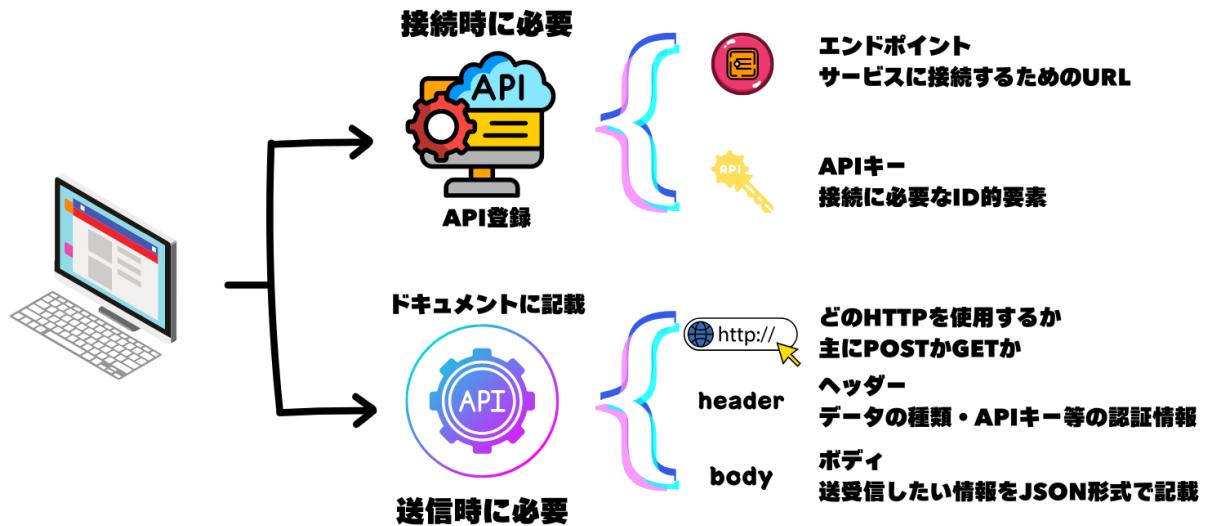
ここまで聞いたら API だけ使えばいいじゃんって思うかもしれないけど、注意が必要!

それは「API 提供サービスに依存してしまう」ってこと、要は API 提供が中止、制限等により情報が取得できなくなってしまう危険性があるってこと!

API を使用するためには 2つ必要だよ。

- ① API に接続するための情報
- ② API に送信する情報

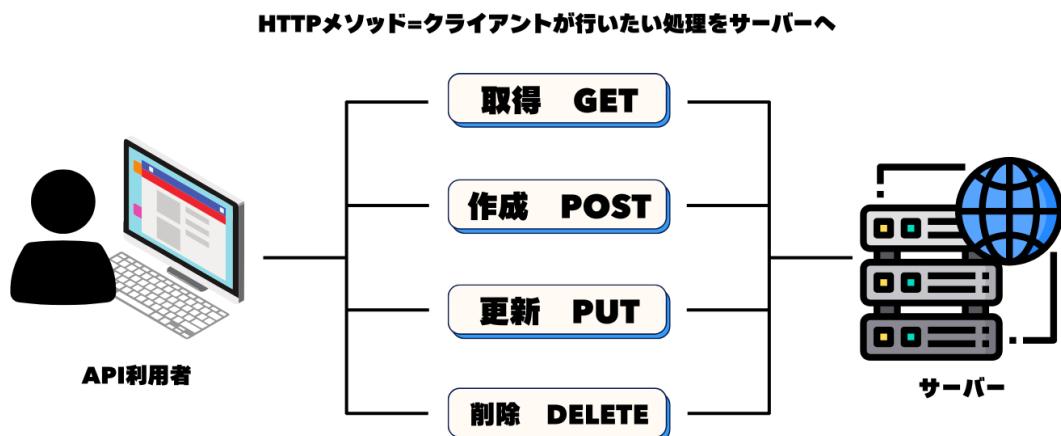
- ① は、URL の API エンドポイントと接続に使用する API キーが必要
- ② は、HTTP メソッド (POST か GET) 、ヘッダーのデータの種類や API キーなどの認証情報、ボディの送受信したい情報を JSON 形式で記載となる。



初めて聞く用語もあるかもしれないけど、スクレイピングよりめっちゃ簡単だよって覚えておけば今は大丈夫!

## HTTP メソッド!

HTTP メソッドには、ユーザーが行いたい処理をサーバーに伝えるという役割がある。



### ① データを取得する GET

Web ページや API からのデータ取得、css ファイルや Js ファイルといったデータ取得に使用する。

### ② 作成する POST

フォームからデータを送信したり、新規アカウントやデータを作成したりする際に使用する。

### ③ 更新する PUT

アカウントや SNS での投稿といった既存データを更新する際に使用する。

### ④ 削除する DELETE

アカウントや記事等のデータ削除する際に使用する。

アカウントを登録して API キーを取得！

今回使用する API は、飲食店検索・予約サイト「ホットペッパーグルメ」の API で、店舗情報を取得してみよう！

<https://webservice.recruit.co.jp/>にアクセス

The screenshot shows the Recruit WEB SERVICE homepage. A red box highlights the '新規登録' (New Registration) button. Another red box highlights the 'WEBサービス一覧' (List of Web Services) section, which includes a link to 'ホットペッパーグルメ'. On the right, there's a 'WHAT'S NEW' section for July 2025, a 'HOT PEPPER グルメ' logo, and maintenance information for the Hot Pepper Guru service.

お問い合わせ 利用規約 プライバシーポリシー(2021年4月改定)  
退会申請

新規登録

WEBサービス一覧  
・ホットペッパーグルメ

WHAT'S NEW

2025/07/07  
リクルートウェブサービス システムメンテナンスのお知らせ

以下の日程でシステムメンテナンス作業が予定されています。  
システムメンテナンス期間中は、リクルートウェブサービスをご利用いただくことができません。

メンテナンス予定日時

2025年8月6日(水) 14時30分頃 ~ 20時00分頃  
※時間は前後する場合がございます。

ご利用中の方にはご迷惑をおかけしますが、何卒ご理解いただきますようお願い申し上げます。

ホットペッパー HOT PEPPER グルメ  
PRODUCED BY RECRUIT

The screenshot shows the 'New Registration' process. A red box highlights the '新規登録' (New Registration) button. Another red box highlights the 'WEBサービス一覧' (List of Web Services) section, which includes a link to 'ホットペッパーグルメ'. The registration steps shown are: ユーザー登録 (User Registration), 認証メール送信 (Send Verification Email), 登録完了 (Registration Complete), and APIキー送信 (Send API Key).

お問い合わせ 利用規約 プライバシーポリシー(2021年4月改定)  
退会申請

新規登録

WEBサービス一覧  
・ホットペッパーグルメ

II 新規登録

ユーザー登録 認証メール送信 登録完了 APIキー送信

APIを利用するにあたって必要なAPIキーを発行します。利用規約をご確認・ご承諾の上、下記のフォームにメールアドレスを記入後送信ボタンを押して下さい。  
APIキー発行後に、登録いただいたメールアドレスを変更される場合は、新しいメールアドレスを改めて登録し、新しい API キーを発行した上で、利用終了した旧メールアドレス（旧 API キー）の退会申請を行ってください。

[プライバシーポリシー](#)  
[リクルートWEBサービス 利用規約](#)

上記の利用規約およびプライバシーポリシーに同意する

メールアドレスを入力してください 送信

登録したメールアドレスに通知が来て、URL部分をクリックすると再度APIキーの通知がくるよ！

そのAPIキーは大切に保管し他人に教えないようにしましょう！

APIドキュメントを理解しよう！

使用するAPIの種類は？また、どういったものを提供してくれるのか？を把握しましょう！

The screenshot shows the Recruit WEB SERVICE API Reference page. The main navigation bar includes links for Home, About the Service, Common Questions, Contact, Terms of Use, Privacy Policy (last updated April 2021), and Logout. On the left, there's a sidebar with links for New Registration, Service Catalog (highlighted), Usage Examples, API Reference (highlighted), and Requests. The main content area is titled "API Reference" and shows two sections: "Gourmet Search API" and "Store Name Search API". The "Gourmet Search API" section is detailed with a red box around its URL: "http://webservice.recruit.co.jp/hotpepper/gourmet/v1/". Below it is a table of parameters:

パラメータ	項目名	説明	必須	値
key	APIキー	APIを利用するため割り当てられたキーを設定します。	<input type="radio"/>	
id	お店ID	お店に割り当てられた番号で検索します。		(例) J999999999 20個まで指定可。 <sup>*2</sup>
name	掲載店名	お店の名前で検索(部分一致)します。		
name_kana	掲載店名かな	お店の読みかなで検索(部分一致)します。		
name_any	掲載店名 OR かな	お店の名前または読みかな両方をOR検索(部分一致)します。		
tel	電話番号	お店の電話番号で検索します。半角数字(ハイフンなし)		(例) 035550000
address	住所	お店の住所で検索(部分一致)します。		

### ▶ サンプルクエリ

大エリアコード=Z011(東京)のお店を検索

```
http://webservice.recruit.co.jp/hotpepper/gourmet/v1/?key=[APIキー]&large_area=Z011
```

■地点を指定して、その範囲にあるお店をオススメ順に取得

```
http://webservice.recruit.co.jp/hotpepper/gourmet/v1/?key=[APIキー]&lat=34.67&lng=135.52&range=5&order=4
```

※実際にご利用いただくには新規登録をした上で、取得したご自身のAPIキーをGETパラメータに加えていただく必要があります。

[▲TOP](#)

### ▶ レスポンスフィールド

フィールド	説明	例	lite*1
results			○
api_version	APIのバージョン	1.20	○
results_available	クエリー条件にマッチする、検索結果の全件数	47	○
results_returned	このXMLに含まれる検索結果の件数	47	○
results_start	検索結果の開始位置	1	○
shop	[複数要素]		○
↑ id	お店ID	J9999999999	○
↑ name	掲載店名	居酒屋 ホットペッパー	○

The screenshot shows the Recruit WEB SERVICE API Reference page. On the left, there's a sidebar with links like '新規登録' (New Registration), 'WEBサービス一覧' (List of WEB Services), and 'APIリファレンス'. The main content area is titled 'APIリファレンス' and lists various API endpoints. Numbered callouts point to specific sections: 1 points to 'グルメサーチAPI' (Gourmet Search API); 2 points to '店名サーチAPI' (Shop Name Search API); 3 points to '検索用ディナー予算マスタAPI' (Search Dinner Budget Master API); 4 points to '大サービスエリアマスタAPI' (Large Service Area Master API); 5 points to 'サービスエリアマスタAPI' (Service Area Master API); 6 points to '大エリアマスタAPI' (Large Area Master API); 7 points to '中エリアマスタAPI' (Medium Area Master API); 8 points to '小エリアマスタAPI' (Small Area Master API); 9 points to 'ジャンルマスタAPI' (Genre Master API); 10 points to 'クレジットカードマスタAPI' (Credit Card Master API); 11 points to '特集マスタAPI' (Special Feature Master API); 12 points to '特集カテゴリマスタAPI' (Special Feature Category Master API); and 13 points to 'エラー時のレスポンス' (Response when an error occurs). The 'HOT PEPPER クルメ' logo is also visible.

API にリクエストを送ってみよう！

API にリクエストを送信する場合は、`requests` モジュールが使用される。

```
import requests
```

次に②のスクリエイピングでもやったように、エンドポイント=URL を変数 `url` に代入します。

```
url = 'https://webservice.recruit.co.jp/hotpepper/gourmet/v1/'
```

今回はとりあえず五反田駅周辺というキーワード (`keyword`) で検索してみよう。

そして最後にもう一つ指定をしておくよ。

それは、`format` パラメータになるんだけど、指定しなくてもレスポンスを受け取ることはできるんだけど、指定なしのデフォルト状態だと、XML という形式でデータが返ってきちゃうから、JSON 形式で返ってくるように指定しよう！

取得した API キーと 3 つのパラメータを `params` という辞書型で用意しよう！

```
params = {
    'key':'4eaf106c7c06de8e',
    'keyword':'五反田駅',
    'format':'json'
}
```

この、リクエストパラメータ `params` とエンドポイント `url` を用いて、API をたたくよ！  
ドキュメントにもあったように GET メソッドを使用し、変数 `res` に代入！

```
res = requests.get(url, params)
```

```
res
```

実行結果

```
<Response [200]>
```

成功してるね！

中身を確認しよう！

```
res.json()
```

```
{'results': {'api_version': '1.30',
  'results_available': 37,
  'results_returned': '10',
  'results_start': 1,
  'shop': [{}{'id': 'J004402152',
    'name': '海坊主五反田駅前店',
    'logo_image':
      'https://imgfp.hotp.jp/SYS/cmn/images/common/diary/custom/m30\_img\_noimage.gif',
    'name_kana': 'うみぼうずごたんだえきまえてん',
    'address': '東京都品川区西五反田2-7-8 誠實ビル 7 階',
    'station_name': '五反田',
    'ktai_coupon': 0,
    'large_service_area': {'code': 'SS10', 'name': '関東'},
    'service_area': {'code': 'SA11', 'name': '東京'},
    'large_area': {'code': 'Z011', 'name': '東京'},
    'middle_area': {'code': 'Y025', 'name': '品川・目黒・田町・浜松町・五反田'},
    'small_area': {'code': 'X086', 'name': '五反田'},
    'lat': 35.6263279137,
    'lng': 139.7224702104,
    'genre': {'name': '居酒屋', 'catch': '鮮魚とお肉料理を地酒と共に堪能できる', 'code': 'G001'},
    'sub_genre': {'name': '中華', 'code': 'G007'},
    'budget': {'code': 'B003',
      'name': '3001~4000 円',
      'average': 'ランチ:900 円~3000 ディナー:3000 円'},
    'catch': '気軽に楽しめるランチ 多彩なコースで満足◎',
    'capacity': 95,
```

情報がいっぱい取得できているね！

「shop」に店舗情報が格納されているけど、そのままキーを shop で指定してしまうとエラーになるよ！

results の中の shop だということを認識しておこう！

```
res.json()['results']['shop']
```

```
[{'id': 'J004402152',
  'name': '海坊主五反田駅前店',
  'logo_image':
https://imgfp.hotp.jp/SYS/cmn/images/common/diary/custom/m30\_img\_noimage.gif,
  'name_kana': 'うみぼうずごたんだえきまえてん',
  'address': '東京都品川区西五反田2-7-8 誠實ビル7階',
  'station_name': '五反田',
  'ktai_coupon': 0,
  'large_service_area': {'code': 'SS10', 'name': '関東'},
  'service_area': {'code': 'SA11', 'name': '東京'},
  'large_area': {'code': 'Z011', 'name': '東京'},
  'middle_area': {'code': 'Y025', 'name': '品川・目黒・田町・浜松町・五反田'},
  'small_area': {'code': 'X086', 'name': '五反田'},
  'lat': 35.6263279137,
  'lng': 139.7224702104,
  'genre': {'name': '居酒屋', 'catch': '鮮魚とお肉料理を地酒と共に堪能できる', 'code': 'G001'},
  'sub_genre': {'name': '中華', 'code': 'G007'},
  'budget': {'code': 'B003',
    'name': '3001~4000円',
    'average': 'ランチ:900円~3000円 ディナー:3000円'},
  'catch': '気軽に楽しめるランチ 多彩なコースで満足◎',
  'capacity': 95,
  'access': 'JR五反田駅西口より徒歩約2分',
  'mobile_access': 'JR五反田駅西口より徒歩約2分',
  'urls': {'pc': https://www.hotpepper.jp/strJ004402152/?vos=nhppalsa000016},
  'photo': {'pc': {'l':
https://imgfp.hotp.jp/IMGH/64/95/P047976495/P047976495\_238.jpg,
    ...
  },
  'pet': '不可',
  'child': 'お子様連れ不可 :平日ディナータイム小学生以下のお子様同席はご遠慮いただいております。ご了承の程お願ひいたします。',
  'wifi': 'なし',
  'coupon_urls': {'pc':
https://www.hotpepper.jp/strJ001281501/map/?vos=nhppalsa000016,
    'sp': https://www.hotpepper.jp/strJ001281501/scoupon/?vos=nhppalsa000016}}]
```

ちゃんと店舗情報が取得できているね！

この中には複数の店舗情報が格納されているから、変数 `shops` に代入しておこう！

```
shops = res.json()['results']['shop']
shops[:3]
```

では、今は何店舗の情報が取得できているか確認してみよう！

```
len(shops)
```

```
10
```

これでとりあえず 10 店舗の情報を取得することができた！

今回は、`keyword` に「五反田駅」を指定しただけだから 10 店舗くらいになっているけど、検索範囲を広げることで多くの情報を取得することができるようになる。

コードを一つにまとめよう！

より多くのデータを取得してみよう！

まとめたコードに追記をするのは、count と start を指定する。

count は、1 ページあたりの取得数で、start は、検索の開始位置となるよ！

```
import requests

url = 'https://webservice.recruit.co.jp/hotpepper/gourmet/v1/'

start = 1
count = 100

params = {
    'key':'4eaf106c7c06de8e',
    'keyword':'五反田駅',
    'format':'json',
    'start': start,
    'count': count
}
res = requests.get(url, params)

shops = res.json()['results']['shop']
shops[:3]
```

確認してみよう！

```
len(shops)
```

```
37
```

・・・意外と少ないね

API で取得したデータを加工しよう！

必要なデータを抽出して整形する。

必要なデータを変数 shops の中からちゅ出して、データを整形していく！

取得するデータは・・・・

- ① name 店舗名
- ② address 住所
- ③ capacity 総席数
- ④ access 交通アクセス
- ⑤ urls > pc 店舗 URL
- ⑥ open 営業時間
- ⑦ budget > name ディナー予算

まずは、1 店舗から必要なデータを取り出してみよう！

```
shop = shops[0]
```

```
datum = {  
    'name': shop['name'],  
    'address': shop['address'],  
    'capacity': shop['capacity'],  
    'access': shop['access'],  
    'open': shop['open'],  
    'budget': shop['budget']['name'],  
    'urls': shop['urls']['pc']  
}
```

```
datum
```

```
{'name': '海坊主五反田駅前店',  
 'address': '東京都品川区西五反田2-7-8 誠實ビル 7階',  
 'capacity': 95,  
 'access': 'JR五反田駅西口より徒歩約 2 分',  
 'open': '月～日、祝日、祝前日：11:00～14:30 (料理 L.O. 14:00 ドリンク L.O. 14:00) 16:30～翌 0:00  
(料理 L.O. 23:00 ドリンク L.O. 23:00)',  
 'budget': '3001～4000 円',  
 'urls': 'https://www.hotpepper.jp/strJ004402152/?vos=nhppalsa000016'}
```

全店舗から必要なデータのみを取り出そう！

変数 data に格納！

```
data = []
for shop in shops:
    datum = {
        'name': shop['name'],
        'address': shop['address'],
        'capacity': shop['capacity'],
        'access': shop['access'],
        'open': shop['open'],
        'budget': shop['budget']['name'],
        'urls': shop['urls']['pc']
    }
    data.append(datum)
```

```
data[:5]
```

```
[{'name': '海坊主五反田駅前店',
  'address': '東京都品川区西五反田2-7-8 誠實ビル 7階',
  'capacity': 95,
  'access': 'JR五反田駅西口より徒歩約2分',
  'open': '月～日、祝日、祝前日：11:00～14:30 (料理L.O. 14:00 ドリンクL.O. 14:00)16:30～翌0:00 (料理L.O. 23:00 ドリンクL.O. 23:00)',
  'budget': '3001～4000円',
  'urls': 'https://www.hotpepper.jp/strJ004402152/?vos=nhppalsa000016'},

  {'name': '幻の和牛と三崎港直送鮮魚 MORGEN 五反田駅前店',
  'address': '東京都品川区西五反田1-18-1 五反田EXビル7F',
  'capacity': 70,
  'access': '五反田駅徒歩3分／大崎広小路駅徒歩1分／大崎駅徒歩10分',
  'open': '月、日、祝日：17:00～22:15 (料理L.O. 21:45 ドリンクL.O. 22:00)火～土、祝前日：17:00～23:00 (料理L.O. 22:30 ドリンクL.O. 22:45)',
  'budget': '3001～4000円',
  'urls': 'https://www.hotpepper.jp/strJ001278727/?vos=nhppalsa000016'},

  {'name': '王豚足家 オウトンソクヤ 五反田本店',
  'address': '東京都品川区西五反田1-4-2',
  'capacity': 54,
  'access': 'JR五反田駅西口から 徒歩1分 (旧野村証券ビル裏)/五反田駅から87m',
  'open': '月：11:30～14:30 (料理L.O. 14:00 ドリンクL.O. 14:00)17:00～23:00 (料理L.O. 22:30 ドリンクL.O. 22:30)火～木：11:30～14:30 (料理L.O. 14:00)17:00～翌0:00 (料理L.O. 23:30 ドリンクL.O. 23:30)金：11:30～14:30 (料理L.O. 14:00 ドリンクL.O. 14:00)17:00～翌4:00 (料理L.O. 翌3:30 ドリンクL.O. 翌3:30)土：17:00～翌4:00 (料理L.O. 翌3:30 ドリンクL.O. 翌3:30)'},
```

pandas の DataFrame 型変換して分析！

リストのデータを pandas に読み込ませてみよう！

```
import pandas as pd  
df = pd.DataFrame(data)
```

```
df.head()
```

	name	address	capacity	access	open	budget	urls
0	海坊主五反田駅前店	東京都品川区西五反田2-7-8 誠實ビル7階	95	J R五反田駅西口より徒歩約2分	月～日、祝日、祝前日: 11:00～14:30 (料理L.O. 14:00 ドリンクL.O. ...)	3001～4000円	<a href="https://www.hotpepper.jp/strJ004402152/?vos=nhan...">https://www.hotpepper.jp/strJ004402152/?vos=nhan...</a>
1	幻の和牛と三崎港直送鮮魚 MORGEN 五反田駅前店	東京都品川区西五反田1-18-1 五反田EXビル7F	70	五反田駅徒歩3分／大崎広小路駅徒歩1分／大崎駅徒歩10分	月、日、祝日: 17:00～22:15 (料理L.O. 21:45 ドリンクL.O. 22:...)	3001～4000円	<a href="https://www.hotpepper.jp/strJ001278727/?vos=nhan...">https://www.hotpepper.jp/strJ001278727/?vos=nhan...</a>
2	王豚足家 オウトンソクヤ 五反田本店	東京都品川区西五反田1-4-2	54	J R五反田駅西口から 徒歩1分 (旧野村証券ビル裏)五反田駅から87m	月: 11:30～14:30 (料理L.O. 14:00 ドリンクL.O. 14:00) 17:...	4001～5000円	<a href="https://www.hotpepper.jp/strJ000132490/?vos=nhan...">https://www.hotpepper.jp/strJ000132490/?vos=nhan...</a>
3	居酒屋 せんや 五反田駅本店	東京都品川区東五反田1-17-7 大宗五反田ビル202	100	各線五反田駅東口徒歩2分／鹿児島黒牛や鮮度抜群の刺身を完全個室で堪能♪	月～金、日、祝前日: 16:00～翌0:00 (料理L.O. 23:00 ドリンクL.O. ...)	3001～4000円	<a href="https://www.hotpepper.jp/strJ001244947/?vos=nhan...">https://www.hotpepper.jp/strJ001244947/?vos=nhan...</a>
4	村ざ来 五反田店	東京都品川区西五反田2-18-3 グレイス五反田107	42	五反田駅西口から徒歩5分	月～金: 11:30～14:30 (料理L.O. 14:00 ドリンクL.O. 14:00) ...	2001～3000円	<a href="https://www.hotpepper.jp/strJ003498617/?vos=nhan...">https://www.hotpepper.jp/strJ003498617/?vos=nhan...</a>

文字列から必要な情報のみを抽出！

予算が入っている budget カラムに対して、文字列処理を行い、予算の下限、上限を数値 (int 型) で取り出し、予算の下限 lower\_budget カラム、上限を upper\_budget カラムに入れよう！

まずは、budget カラムから 1 つ目のデータを取り出して、\_budget という変数に代入！

```
_budget = df['budget'][0]  
_budget
```

```
'3001~4000 円'
```

続いて下限の金額を取り出す

```
_budget.split('~')
```

```
['3001', '4000 円']
```

そして下限の値を int 型に変換する。

```
int(_budget.split('~')[0])
```

```
3001
```

続けて上限も取り出そう！

```
_budget.split('~')[1]
```

```
'4000 円'
```

不要な部分を replace() メソッドで取り除こう！

```
_budget.split('~')[1].replace('円','")
```

```
'4000'
```

上限の値を int 型に変換する。

```
int(_budget.split('～')[1].replace('円',''))
```

```
4000
```

これで下限上限の値の取得ができたので、全ての値に対して同様の処理をしよう。

```
lower_budgets = []
upper_budgets = []

for _budget in df['budget']:
    if _budget != "":
        lower_budget=int(_budget.split('～')[0])
        upper_budget=int(_budget.split('～')[1].replace('円',''))
    else:
        lower_budget = 0
        upper_budget = 0

    lower_budgets.append(lower_budget)
    upper_budgets.append(upper_budget)
```

下限と上限のカラムを作成しよう！

```
df['lower_budget'] = lower_budgets
df['upper_budget'] = upper_budgets
```

```
df.head()
```

	name	address	capacity	access	open	budget	urls	lower_budget	upper_budget
0	海坊主五反田駅前店	東京都品川区西五反田2-7-8 誠實ビル7階	95	J R五反田駅西口より徒歩約2分	月～日、祝日、祝前日:11:00～14:30 (料理L.O. 14:00 ドリンクL.O....)	3001～4000円	<a href="https://www.hotpepper.jp/strJ004402152/?vos=nh...">https://www.hotpepper.jp/strJ004402152/?vos=nh...</a>	3001	4000
1	幻の和牛・三崎港直送鮮魚MORGEN 五反田駅前店	東京都品川区西五反田1-18-1 五反田EXビル7F	70	五反田駅徒歩3分／大崎広小路駅徒歩1分／大崎駅徒歩10分	月、日、祝日: 17:00～22:15 (料理L.O. 21:45 ドリンクL.O. 22:...)	3001～4000円	<a href="https://www.hotpepper.jp/strJ001278727/?vos=nh...">https://www.hotpepper.jp/strJ001278727/?vos=nh...</a>	3001	4000
2	王豚足家 オウトンソクヤ 五反田本店	東京都品川区西五反田1-4-2	54	J R五反田駅西口から徒歩1分 (旧野村証券ビル裏)/五反田駅から87m	月: 11:30～14:30 (料理L.O. 14:00 ドリンクL.O. 14:00) 17...	4001～5000円	<a href="https://www.hotpepper.jp/strJ000132490/?vos=nh...">https://www.hotpepper.jp/strJ000132490/?vos=nh...</a>	4001	5000
3	居酒屋 せんや 五反田駅本店	東京都品川区東五反田1-17-7大宗五反田ビル202	100	各線五反田駅東口徒歩2分／鹿児島黒牛や鮮度抜群の刺身を完全個室で堪能♪	月～金、日、祝前日:16:00～翌0:00 (料理L.O. 23:00 ドリンクL.O. ...)	3001～4000円	<a href="https://www.hotpepper.jp/strJ001244947/?vos=nh...">https://www.hotpepper.jp/strJ001244947/?vos=nh...</a>	3001	4000
4	村さま 五反田店	東京都品川区西五反田2-18-3 グレイス五反田107	42	五反田駅西口から徒歩5分	月～金: 11:30～14:30 (料理L.O. 14:00 ドリンクL.O. 14:00) ...	2001～3000円	<a href="https://www.hotpepper.jp/strJ003498617/?vos=nh...">https://www.hotpepper.jp/strJ003498617/?vos=nh...</a>	2001	3000

条件を指定して必要なデータを抽出しよう！

pandas ではデータの抽出も簡単にできるからやってみよう！

予算上限 upper\_budget から 4,000 円以下の店舗のみを抽出してみましょう！

```
df['upper_budget'] <= 4000
```

```
0    True
1    True
2   False
3    True
4    True
5    True
...
33   True
34   True
35   True
36   False
Name: upper_budget, dtype: bool
```

予算の上限が 4,000 円以下の店舗を変数 df\_filtered に代入

```
df_filtered = df[df['upper_budget'] <= 4000]
```

```
df_filtered.head()
```

	name	address	capacity	access	open	budget	urls	lower_budget	upper_budget
0	海坊主五反田駅前店	東京都品川区西五反田2-7-8 誠實ビル7階	95	J R五反田駅西口より徒歩約2分	月～日、祝日、祝前日: 11:00～14:30 (料理L.O. 14:00 ドリンクL.O....)	3001～ 4000円	<a href="https://www.hotpepper.jp/strJ004402152/?vos=nh...">https://www.hotpepper.jp/strJ004402152/? vos=nh...</a>	3001	4000
1	幻の和牛と三崎港直送鮮魚MORGEN五反田駅前店	東京都品川区西五反田1-18-1 五反田EXビル7F	70	五反田駅徒歩3分 ／大崎広小路駅徒歩1分／大崎駅徒歩10分	月、日、祝日: 17:00～22:15 (料理L.O. 21:45 ドリンクL.O. 22:....)	3001～ 4000円	<a href="https://www.hotpepper.jp/strJ001278727/?vos=nh...">https://www.hotpepper.jp/strJ001278727/? vos=nh...</a>	3001	4000
3	居酒屋せんや五反田駅本店	東京都品川区東五反田1-17-7 大宗五反田ビル202	100	各線五反田駅東口徒歩2分／鹿児島黒牛や鮮度抜群の刺身を完全個室で堪能♪	月～金、日、祝前日: 16:00～翌0:00 (料理L.O. 23:00 ドリンクL.O. ....)	3001～ 4000円	<a href="https://www.hotpepper.jp/strJ001244947/?vos=nh...">https://www.hotpepper.jp/strJ001244947/? vos=nh...</a>	3001	4000
4	村さ来五反田店	東京都品川区西五反田2-18-3 グレイス五反田107	42	五反田駅西口から徒歩5分	月～金: 11:30～ 14:30 (料理L.O. 14:00 ドリンク L.O. 14:00) ...	2001～ 3000円	<a href="https://www.hotpepper.jp/strJ003498617/?vos=nh...">https://www.hotpepper.jp/strJ003498617/? vos=nh...</a>	2001	3000
5	酒蔵 桜ちゃん五反田	東京都品川区西五反田1-4-2秀和五反田駅前レジデンス2階	150	JR五反田駅徒歩2分/都営浅草線五反田駅徒歩2分/東急池上線五反田駅徒歩2分	月～金: 16:00～ 23:30 (料理L.O. 22:30 ドリンク L.O. 23:00) ...	3001～ 4000円	<a href="https://www.hotpepper.jp/strJ003893833/?vos=nh...">https://www.hotpepper.jp/strJ003893833/? vos=nh...</a>	3001	4000