

なろう小説API 入門講座

著：万灯あお

『小説家になろう』のデータを入手する唯一の方法である公式APIの『なろう小説API』をPythonで紹介コピペと改変だけで『小説家になろう』のデータを入手できるように！総合評価ポイントの分布を表示するソースコードなども紹介します！！

まえがき

この同人誌では「なろう小説 API」を Python で利用するためのソースコードを紹介・解説しています。「なろう小説 API」とは「小説家になろう」が提供している「なろうデベロッパー」における機能の一つで、「小説家になろう」に投稿された作品に関する様々な情報を入手することができます。取得できる情報の一例としては、作品の文字数、キーワード、総合評価ポイント、評価者数、挿絵の数、会話率などが挙げられます。この同人誌を読んで API の使い方を理解し、「小説家になろう」をデータ分析する第一歩を踏み出しましょう。

この同人誌の流れは次の通りです。

最初に、「なろう小説 API」の説明を行います。どういう流れで API を利用するのか、三つのパラメータの関係と中身について説明をします。

次に、プログラミングの経験がない人に向けて、この同人誌で紹介するソースコードを読めるようになるための基礎知識を紹介しています。ただ、説明は最低限なので検索ワードを紹介しているものと思ってください。

後半部分では、ソースコードの紹介・解説を行います。まず、「なろうデベロッパー」で紹介されている PHP で書かれた二つのソースコードを掲載し、その解説を行います。公式が用意しているソースコードを読み解き、どのように API を利用するのかを確認していきます。その後、筆者が Python で制作したソースコードの紹介・解説を行います。これにより、コピペと簡単な改変だけで自分の欲しい情報を取得することができるようになります。

基本的：「条件を指定して検索結果を 3 件表示する」では、機能を API の利用に絞り、API を利用する際の基本を確認します。

発展的：「指定した期間内の総合評価の分布を調べる（なろう版）」では、指定した期間に投稿された全ての短編作品の情報を取得し、総合評価ポイントの分布を調べます。「なろう小説 API」は同時に 2000 件以上の作品情報を取得するとエラーになるため、それをどう回避するのも紹介します。また、「なろう R18 小説 API」を利用したノクターン版の紹介も行います。

その他に、「なろうユーザ検索 API」を使用したソースコードも紹介します。

この同人誌を読むことで、なろう小説 API を利用する時に大部分をコピペで済ませることができます。また、自分の欲しいデータのためにソースコードを改変する際の手助けとなる知識も提供しています。「小説家になろう」をデータ分析する際の手助けになれば幸いです。

目次

まえがき	… 1	第5章：公式のサンプルコード解説	… 31
第1章：なろうデベロッパーについて	… 3	5-1：なろう小説 API サンプルプログラム「作品簡易一覧(PHP)」	… 31
1-1：各 API の説明	… 4	5-2：なろう小説 API サンプルプログラム「ページ分割対応作品簡易一覧(PHP)」	… 35
第2章：なろう小説 API について	… 5	第6章：Python で作成したソースコード群	… 42
2-1：出力見本	… 5	6-1：基本：「条件を指定して検索結果を3件表示する」	… 42
2-2：出力	… 7	6-2：発展：「指定した期間内の総合評価の分布を調べる（なろう版）」	… 46
2-3：of パラメータ	… 11	6-2-1：統計の知識	… 58
2-4：条件抽出 GET パラメータ	… 12	6-3：発展：「指定した期間内の総合評価の分布を調べる（ノクターン版）」	… 59
2-5：出力 GET パラメータ	… 17	6-4：その他：「なろうユーザ検索 API」	… 60
第3章：なろう R18 小説 API について	… 19	6-5：これから	… 61
3-1：出力見本	… 19	第7章：あとがき	… 62
3-2：出力	… 20	7-1：あとがき	… 62
3-3：of パラメータ	… 20	7-2：参考にしたサイト	… 62
3-4：条件抽出 GET パラメータ	… 20	第8章：その他	… 63
3-5：出力 GET パラメータ	… 21	8-1：作成・編集メモ	… 63
第4章：基礎知識	… 22	8-2：装丁情報	… 63
4-1：プログラミング全般の知識	… 22	8-3：既刊一覧	… 64
4-2：PHP のルール	… 28	8-4：奥付	… 64
4-3：Python のルール	… 28		
4-4：出力と JSON 形式について	… 29		
4-5：Google Colaboratory について	… 30		

第1章：なろうデベロッパーについて

「なろう小説 API」とは、「なろうデベロッパー」から提供されている「小説家になろう」の公式 API です。API とは、「指定の構文を入力すると特定のデータが出力される機能」とここでは理解してください。なろう小説 API を使うことで作品の検索や作品情報の取得などを行うことができます。

なろうデベロッパー：<https://dev.syosetu.com/>

作品情報の取得という点で注意が必要なのは、小説家になろうではスクレイピングが禁止されていることが挙げられます。利用規約の第 14 条禁止事項に以下の文言があります。

23.なろうデベロッパーで提供している API を利用する以外の方法で、本サイトに自動化された手段を用いてアクセスしたり、データを収集したりすること。

URL：<https://syosetu.com/site/rule/>

つまり、この同人誌で紹介する、なろうデベロッパーで提供されている API を使わずにデータを収集することは禁じられています。なお、なろうデベロッパーで提供されているツールは全て商用サイトでの利用が可能となっています。

なろうデベロッパーでは、大きく分けて 3 つの機能が提供されています。

1 つ目が、Atom フィードです。これは、更新情報をすばやく把握するためのもので、RSS と言った方が馴染みはあるかもしれません。提供されている Atom フィードは以下の 4 つです。

- ・活動報告 Atom
- ・小説 Atom
- ・X 活動報告 Atom
- ・R18 小説 Atom

2 つ目が、なろう小説インラインサーチウィジェットです。これは、作品の検索結果をホームページやブログに埋め込めるようにするためのものです。

3 つ目が、API です。提供されている API は以下の 5 つです。

- ・なろう小説 API
- ・なろう小説ランキング API
- ・なろう殿堂入り API
- ・なろうユーザー検索 API

・なろう R18 小説 API

この同人誌では、3 つ目の API のうち、「なろう小説 API」についてのみ説明します。他の API については適宜読み替えと改変を行うことで対応が可能です。

1-1：各 API の説明

以下では、具体的にそれぞれの API についてざっくりと説明していきます。

1 つ目の「なろう小説 API」では、小説家になろうに掲載されている作品の情報を取得できます。後ほど詳細に見ていきますが、作品名や作者名、作品文字数、総合評価ポイントなどの情報を取得できます。R18 作品については、なろう R18 小説 API を使って取得することになります。

2 つ目の「なろう小説ランキング API」では、制約があるものの日付を指定することでその日の日間、週間、月間、四半期ランキングについて、上位 300 位までの作品の N コードとポイント数を取得することができます。制約としては、週間は火曜日の日付のみ、月間・四半期は毎月 1 日のみとなっています。なお、R18 ランキングは取得できません。

3 つ目の「なろう殿堂入り API」では、指定した作品が過去にどのランキングに載ったかを取得することができます。具体的には、作品の N コードを渡すことで、何年何月何日に日間、週間、月間、四半期ランキングの何位にあり、何ポイント獲得していたのかを取得することができます。なお、R18 ランキングは取得できません。

なろう小説ランキング API となろう殿堂入り API の違いですが、なろう小説ランキング API は日付を指定することでその日のランキングを取得します。一方で、なろう殿堂入り API は作品を指定することで、どの日のどのランキングに載っていたのかを取得するという違いがあります。

4 つ目の「なろうユーザー検索 API」では、指定した条件を満たすユーザーの情報を取得することができます。取得できる情報としては、ユーザー ID やユーザー名、総合評価ポイントの合計などがあります。

5 つ目の「なろう R18 小説 API」では、ノクターンノベルズ、ムーンライトノベルズおよびミッドナイトノベルズに掲載されている R18 小説の情報を取得することができます。なろう小説 API と同様に作品名、作者名、キーワード、総合評価ポイントなどが取得できます。

それぞれの API の説明は「なろうデベロッパー」のサイトを確認してください。本同人誌では、「なろう小説 API」と「なろう R18 小説 API」についてのみ解説します。

第2章：なろう小説 API について

なろう小説 API : <https://dev.syosetu.com/man/api/>

なろう小説 API は小説家になろうで掲載されている作品の情報を取得することができる API です。ここでは、なろう小説 API について詳しく説明していきます。

なろう小説 API を利用する際には、大きく分けて以下の3つの要素が関係してきます。

- ・出力 GET パラメータ
- ・条件抽出 GET パラメータ
- ・出力と of パラメータ

大まかなイメージとしては、API を利用する際の全体的な設定を「出力 GET パラメータ」で行い、「条件抽出 GET パラメータ」でどんな作品を取得するかを条件を指示し、条件を満たした作品の情報が「出力」されます。そして、出力される項目を「of パラメータ」で指示します。

まず、出力見本でどのような出力がなされるのかを確認した後、出力される項目と of パラメータについて見ていきます。次に、どのような条件を設定できるのかを条件抽出 GET パラメータで確認します。最後に API の設定を行うパラメータである出力 GET パラメータを確認します。

2-1：出力見本

具体的なパラメータを見る前にどのような出力が行われるのかのイメージを掴みましょう。

例えば、すべての項目を出力した場合、以下のようになります。ここでは小説家になろうに投稿された作品の中で総合評価ポイントが一番高い作品の作品情報を取得するように API を叩いています。

- ・叩いた API の URL

<https://api.syosetu.com/novelapi/api/?gzip=5&out=json&lim=1&order=hyoka>

- ・出力

```
[{'allcount': 1062274},
{'all_hyoka_cnt': 43453,
'all_point': 410915,
'biggenre': 2,
'daily_point': 438,
'end': 1,
'fav_novel_cnt': 255483,
'general_all_no': 654,
'general_firstup': '2016-01-05 01:24:12',
'general_lastup': '2025-01-10 00:48:16',
'genre': 201,
'gensaku': ''}]
```

'global_point': 921881,
'impression_cnt': 33892,
'isbl': 0,
'isgl': 0,
'isr15': 1,
'isstop': 0,
'istenni': 1,
'istensei': 0,
'iszankoku': 1,
'kaiwaritu': 37,
'keyword': 'ほのぼの 異世界 ネットスーパー 勇者召喚 巻き込まれ ペットが最強 料理 のほほん
旅 R15 残酷な描写あり 異世界転移',
'length': 2184343,
'monthly_point': 2818,
'ncode': 'N2710DB',
'novel_type': 1,
'novelupdated_at': '2025-01-10 00:48:16',
'quarter_point': 9146,
'review_cnt': 181,
'sasie_cnt': 1,
'story': '★㊦アニメ 2 期決定！★\n'
 '◆◆◆オーバーラップノベルス様より書籍 1 4 巻まで発売中！\u3000'
 '本編コミックは 9 巻まで、外伝コミック「スイの大冒険」は 7 巻まで発売中です！◆◆◆\n'
 '異世界召喚に巻き込まれた俺、向田剛志（むこうだ\u3000つよし）。\n'
 'ステータスの鑑定で俺以外の 3 人の召喚者は職業欄に『異世界からやって来た勇者』となっ
るのに、俺だけ『巻き込まれた異世界人』となっていた。\\n'
 'スキルも聖剣術とか聖槍術とか聖魔法とかいろいろすごいのがそろっている。\\n'
 'それなのに俺のは固有スキルで『ネットスーパー』だけだった。\\n'
 '戦闘スキルじゃない俺はすぐさま役立たず的な扱いに。\\n'
 'しかも、召喚した国の王様の言い分がいかにも胡散臭い。\\n'
 'ネット小説を読むのを趣味にしていた俺は、諸々を総合した結果『ダメだ。これはあかんタイ
プの異世界召喚だ。』という結論に至った。\\n'
 'そして俺は召喚されたその日のうちに何やかや言いくるめて城から逃げ出したのだった。\\n',
'time': 4369,
'title': 'とんでもスキルで異世界放浪メシ',
'updated_at': '2025-01-10 11:19:09',
'userid': 679640,
'weekly_point': 744,

```
'writer': '江口\u3000連',  
'yearly_point': 53082}]
```

ここで注意が必要なのは、1 番最初にある allcount は作品情報ではなく検索した際の HIT 数を意味しています。all_hyoka_cnt から始まり、yearly_point で終わる {} で囲まれたものが作品の情報です。具体的な見方は「4-4：出力と JSON 形式について」を確認してください。

以降で出力の各項目について説明していきます。

2-2：出力

出力、つまり取得することのできる作品情報は以下の通りです。

要素	説明
allcount	条件抽出 GET パラメータで指定した条件を満たす作品が何作品あるのかを取得します。出力 GET パラメータで触れますが、ヒットした作品数が最大出力数の 2000 以上の場合でも 2000 という表記にはならずヒットした作品数をきちんと表示します。
title	作品名を取得します。
ncode	作品の N コードを取得します。N コードとは、作品に発行される一意のコードです。作品閲覧ページの URL「 https://ncode.syosetu.com/n〇〇/ 」の n〇〇に入っています。また、作品情報のページでも確認することができます。ちなみに、S コードとは作品のシリーズごとに発行される一意のコードのことです。
userid	作者のユーザーIDを取得します。ID はユーザー詳細ページから確認できます。
writer	作者名を取得します。
story	作品のあらすじを取得します。
biggenre	大ジャンルを数字として取得します。数字の意味は以下の通りです。 <ul style="list-style-type: none">・0：未選択・1：恋愛・2：ファンタジー・3：文芸・4：SF・99：その他・98：ノンジャンル
genre	ジャンルを数字として取得します。数字の意味は以下の通りです。 <ul style="list-style-type: none">・0：未選択〔未選択〕・101：異世界〔恋愛〕・102：現実世界〔恋愛〕・201：ハイファンタジー〔ファンタジー〕

	<ul style="list-style-type: none"> ・202：ローファンタジー〔ファンタジー〕 ・301：純文学〔文芸〕 ・302：ヒューマンドラマ〔文芸〕 ・303：歴史〔文芸〕 ・304：推理〔文芸〕 ・305：ホラー〔文芸〕 ・306：アクション〔文芸〕 ・307：コメディ〔文芸〕 ・401：VR ゲーム〔SF〕 ・402：宇宙〔SF〕 ・403：空想科学〔SF〕 ・404：パニック〔SF〕 ・9901：童話〔その他〕 ・9902：詩〔その他〕 ・9903：エッセイ〔その他〕 ・9904：リプレイ〔その他〕 ・9999：その他〔その他〕 ・9801：ノンジャンル〔ノンジャンル〕
gensaku	現在使用されていません。空文字を取得します。
keyword	キーワードを取得します。他サイトではタグと言われていることもあります。
general_firstup	初回掲載日を取得します。最初に作品を掲載した日時のことを指します。形式はYYYY-MM-DD HH:MM:SS です。具体的に書くと 2016-10-27 23:59:32 となります。
general_lastup	最終掲載日を取得します。連載作品は、次話を投稿することで更新されます。
novel_type	作品が連載か短編かを数字として取得します。1 の場合は連載、2 の場合は短編を意味します。
end	連載中か完結済みかを数字として取得します。0 の場合は短編作品と完結済み作品を、1 の場合は連載中を意味します。
general_all_no	掲載エピソード数を取得します。短編の場合は1になります。
length	作品文字数を取得します。スペースや改行は文字数としてカウントしません。
time	作品を読むのに掛かる時間（分）を取得します。計算方法は、作品文字数÷500 を切り上げた数字です。
isstop	長期連載停止中かを数字として取得します。1 の場合は長期連載停止中を、0 の場合はそれ以外を意味します。なお、長期連載停止中の定義は、「未完結の連載中作品で最終掲載日から 63 日以上経過しているもの」となっています。
isr15	小説を投稿する際に設定で、セルフレイティングの「R15 (15 歳以上推奨)」にチェックを付けたかを数字として取得します。1 の場合はチェックがあることを、

	0 の場合はチェックが付けられていないことを意味します。ちなみに、チェックが入っている場合、自動的に「R15」がキーワードに追加されます。
isbl	小説を投稿する際に設定する「作品に含まれる要素」のうち「ボーイズラブ」にチェックを付けたかを数字として取得します。1 の場合はチェックがあることを、0 の場合はチェックが付けられていないことを意味します。ちなみに、チェックが入っている場合、自動的に「ボーイズラブ」がキーワードに追加されます。
isgl	小説を投稿する際に設定する「作品に含まれる要素」のうち「ガールズラブ」にチェックを付けたかを数字として取得します。1 の場合はチェックがあることを、0 の場合はチェックが付けられていないことを意味します。ちなみに、チェックが入っている場合、自動的に「ガールズラブ」がキーワードに追加されます。
iszankoku	小説を投稿する際に設定する「作品に含まれる要素」のうち「残酷な描写あり」にチェックを付けたかを数字として取得します。1 の場合はチェックがあることを、0 の場合はチェックが付けられていないことを意味します。ちなみにチェックが入っている場合、自動的に「残酷な描写あり」がキーワードに追加されます。
istensei	小説を投稿する際に設定する「作品に含まれる要素」のうち「異世界転生」にチェックを付けたかを数字として取得します。1 の場合はチェックがあることを、0 の場合はチェックが付けられていないことを意味します。ちなみに、チェックが入っている場合、自動的に「異世界転生」がキーワードに追加されます。
istenni	小説を投稿する際に設定する「作品に含まれる要素」のうち「異世界転移」にチェックを付けたかを数字として取得します。1 の場合はチェックがあることを、0 の場合はチェックが付けられていないことを意味します。ちなみに、チェックが入っている場合、自動的に「異世界転移」がキーワードに追加されます。
global_point	総合評価ポイントを取得します。総合評価ポイントの計算方法は(ブックマーク数×2)+評価ポイントです。
daily_point	日間ポイントを取得します。日間ポイントは、ランキングを集計（更新）した時点から過去 24 時間以内に新しく獲得した総合評価ポイントのことです。日間ランキングの決定に使われます。なお日間ランキングは 1 日に 3 回更新されます。
weekly_point	週間ポイントを取得します。週間ポイントは、ランキングを集計（更新）した時点から過去 7 日以内に新しく獲得した総合評価ポイントのことです。週間ランキングの決定に使われます。なお、週間ランキングは毎日の朝に 1 回更新されます。
monthly_point	月間ポイントを取得します。月間ポイントは、ランキングを集計（更新）した時点から過去 30 日以内に新しく獲得した総合評価ポイントのことです。月間ランキングの決定に使われます。なお月間ランキングは毎日の朝に 1 回更新されます。
quarter_point	四半期ポイントを取得します。四半期ポイントは、ランキングを集計（更新）した時点から過去 90 日以内に新しく獲得した総合評価ポイントのことです。四半期ランキングの決定に使われます。なお、四半期ランキングは毎日の朝に 1 回更新されます。

yearly_point	年間ポイントを取得します。年間ポイントは、ランキングを集計（更新）した時点から過去 365 日以内に新しく獲得した総合評価ポイントのことです。年間ランキングの決定に使われます。なお、年間ランキングは毎週火曜日の朝に 1 回更新されます。
fav_novel_cnt	ブックマーク数を取得します。ブックマーク数は作品の末にある「ブックマークに追加」を押されることで増えます。ブックマークはログインしたユーザーのみが行うことができます。なお、ブックマークは 1 ユーザーにつき 4000 件まで登録することができます。
impression_cnt	感想数を取得します。感想は作者の設定により受け付けるか受け付けないかを定めることができます。受け付ける場合もログインしたユーザーのみから受け付けるか、ログインしなくても受け付けるかを定めることができます。
review_cnt	レビュー数を取得します。レビューは小説家になろうではイチオシレビューという呼び方をしています。イチオシレビューはログイン済みのユーザーが 1 つの作品に対して 1 つまで書けます。また、タイトルは 8 文字以上 100 字以内、本文は 150 文字以上 400 字以内で入力することが求められています。作者はイチオシレビューを受け付けるかを定めることができます。
all_point	評価ポイント数を取得します。評価ポイントは 5 段階評価で星一つが 2pt を意味します。つまり、評価は 1 ユーザーにつき 2,4,6,8,10pt のいずれかのポイントになります。なお、評価ポイントはログインしたユーザーのみが行うことができます。また、作者が評価を受け付けない設定をしているとポイントを入れることはできません。
all_hyoka_cnt	評価者数を取得します。評価者数は評価ポイントを入れたユーザーの数です。
sasie_cnt	挿絵の数を取得します。
kaiwaritu	<p>会話率を取得します。会話率とは、文章のうち会話行が占める割合を示すものです。具体的な計算方法は、「会話行 ÷ (地の文 + 会話行) × 100」です。それぞれの定義は以下の通りです。</p> <p>会話行：鉤括弧（」）で終わる行のこと。</p> <p>地の文：会話行ではなく、読点（、）・句点（。）・三点リーダー（…）・疑問符（?）・感嘆符（!）のいずれかを含む行のこと。</p> <p>無効行：会話行でも地の文でもない行のこと。計算対象外になる。</p> <p>例：</p> <ul style="list-style-type: none"> ・会話行：「吾輩は猫である」 ・地の文：「吾輩は猫である」そうその不審者は言った。 ・無効行：吾輩は猫であると不審者は名乗った
novelupdated_at	作品データが最後に更新された時刻を取得します。

updated_at	最後にデータが更新された日時を取得します。ここでいうデータは作品データ（本文）だけでなく総合評価ポイントなども含みます。なろう小説 API で取得できる情報のいずれかが更新された日時だという理解で問題ないかと思います
------------	--

以上の項目をなろう小説 API では出力（取得）することができます。

2-3：of パラメータ

情報を出力する際にすべての項目を出力する必要はありません。必要な項目だけを出力することで、通信データ量の削減やデータの管理が楽になるメリットがあります。小説家になろう公式からもデータ量の削減という点から必要な項目のみの出力が推奨されています。

出力する項目を選択する方法が of パラメータを指定することです。以下で紹介するパラメータを指定することで、指定した項目のみが出力されるようになります。複数項目を出力する際はハイフン(-)記号で区切ります。of パラメータが未指定の場合は全項目が指定されます。

パラメータ	出力要素	説明
t	title	作品名
n	ncode	N コード
u	userid	作者のユーザーID
w	writer	作者名
s	story	作品のあらすじ
bg	biggenre	大ジャンル
g	genre	ジャンル
k	keyword	キーワード
gf	general_firstup	初回掲載日
gl	general_lastup	最終掲載日
nt	noveltype	連載か短編か ※「novel_type ではございません」とサイトには書いてあるが、これは出力する際に使用する連想配列のキーが novel_type ではなく noveltype だということである。
e	end	短編・完結済みか連載中か
ga	general_all_no	掲載エピソード数
l	length	作品文字数
ti	time	読了時間
i	isstop	長期連載停止中か
ir	isr15	「R15」が含まれるか否か
ibl	isbl	「ボーイズラブ」が含まれるか否か

igl	isgl	「ガールズラブ」が含まれるか否か
izk	iszankoku	「残酷な描写あり」が含まれるか否か
its	istensei	「異世界転生」が含まれるか否か
iti	istenni	「異世界転移」が含まれるか否か
gp	global_point	総合評価ポイント
dp	daily_point	日間ポイント
wp	weekly_point	週間ポイント
mp	monthly_point	月間ポイント
qp	quarter_point	四半期ポイント
yp	yearly_point	年間ポイント
f	fav_novel_cnt	ブックマーク数
imp	impression_cnt	感想数
r	review_cnt	レビュー数
a	all_point	評価ポイント
ah	all_hyoka_cnt	評価者数
sa	sasie_cnt	挿絵の数
ka	kaiwaritu	会話率
nu	novelupdated_at	作品の更新日時
ua	updated_at	最終更新日時

2-4：条件抽出 GET パラメータ

ここでは、条件抽出 GET パラメータについて見ていきます。

条件抽出 GET パラメータとは、作品を検索する際に指定できる条件のことを指します。

以下が条件抽出 GET パラメータの一覧です。

- ・単語で抽出する：文字列を指定して抽出します。「何の文字列で検索するか」と「どこを検索するか」の二つに分けられます。
- ・検索単語指定：検索・除外する単語（文字列）を指定します。半角または全角スペースで区切ると AND 抽出になります。部分一致で検索・除外します。

パラメータ	値	説明
word	string	指定した単語（文字列）を検索します。
notword	string	指定した単語（文字列）を除外検索します。この単語が含まれている文章は HIT しないということです。

・抽出対象の範囲：word、notword で指定した単語を使ってどの項目を検索するのかを指定します。以下の4項目すべてを指定しない場合は4項目すべてを抽出対象に指定します。なお、word が N コードだと判断される文字列の場合、この指定は無視され N コードから作品を検索します。

title	int	1 の場合はタイトルを抽出対象にします。
ex	int	1 の場合はあらすじを抽出対象にします。
keyword	int	1 の場合はキーワードを抽出対象にします。
wname	int	1 の場合は作者名を抽出対象にします。

・大ジャンルで抽出する：大ジャンルを指定して抽出するパラメータです。複数の大ジャンルを指定する場合はハイフン(-)記号で区切ります。ハイフン(-)記号は string 型でしか使えないことに注意しましょう。

biggenre	int, string	指定した大ジャンルで検索します。
notbiggenre	int, string	指定した大ジャンルを除外検索します。

指定できるパラメータは以下の通りです。

0：未選択

1：恋愛

2：ファンタジー

3：文芸

4：SF

99：その他

98：ノンジャンル

・ジャンルで検索する：ジャンルを指定して抽出するパラメータです。複数のジャンルを指定する場合はハイフン(-)記号で区切ります。genre が string のみとなっていますが int でも検索できることを確認しました。

genre	string	指定したジャンルで検索します。
notgenre	int, string	指定したジャンルを除外検索します。

指定できるパラメータは以下の通りです。

0：未選択 [未選択]

101：異世界 [恋愛]

102：現実世界 [恋愛]

201：ハイファンタジー [ファンタジー]

202：ローファンタジー [ファンタジー]

301：純文学 [文芸]

302：ヒューマンドラマ [文芸]

303：歴史 [文芸]

304：推理 [文芸]

305：ホラー〔文芸〕
 306：アクション〔文芸〕
 307：コメディ〔文芸〕
 401：VR ゲーム〔SF〕
 402：宇宙〔SF〕
 403：空想科学〔SF〕
 404：パニック〔SF〕
 9901：童話〔その他〕
 9902：詩〔その他〕
 9903：エッセイ〔その他〕
 9904：リプレイ〔その他〕
 9999：その他〔その他〕
 9801：ノンジャンル〔ノンジャンル〕

・ユーザ ID で検索する：ユーザ ID を指定して抽出するパラメータです。複数のユーザ ID を指定する場合はハイフン(-)記号で区切ります。

userid	int, string	指定したユーザーを検索します。
--------	-------------	-----------------

・「作品に含まれる要素」を指定して検索する：「作品に含まれる要素」を指定もしくは除外指定して抽出するパラメータです。

isr15	int	1 を指定した場合、「R15」が含まれている作品を検索します。
notr15	int	1 を指定した場合、「R15」が含まれている作品を除外検索します。
isbl	int	1 を指定した場合、「ボーイズラブ」が含まれている作品を検索します。
notbl	int	1 を指定した場合、「ボーイズラブ」が含まれている作品を除外検索します。
isgl	int	1 を指定した場合、「ガールズラブ」が含まれている作品を検索します。
notgl	int	1 を指定した場合、「ガールズラブ」が含まれている作品を除外検索します。
iszankoku	int	1 を指定した場合、「残酷な描写あり」が含まれている作品を検索します。
notzankoku	int	1 を指定した場合「残酷な描写あり」が含まれている作品を除外検索します。
istensei	int	1 を指定した場合、「異世界転生」が含まれている作品を検索します。
nottensei	int	1 を指定した場合、「異世界転生」が含まれている作品を除外検索します。
istenni	int	1 を指定した場合、「異世界転移」が含まれている作品を検索します。
nottenni	int	1 を指定した場合、「異世界転移」が含まれている作品を除外検索します。
istt	int	1 を指定した場合、「異世界転生」または「異世界転移」が含まれている作品を検索します。

・文字数を指定して検索する：ここでのいう文字数とは、作品から一部のタグ記号、ルビ、ルビのふりがな部分、みてみんの画像挿入コード、スペース、改行を抜いた値のことです。

minlen	int	作品の最小文字数を指定します。つまり、指定した文字数以上の作品を抽出します。
maxlen	int	作品の最大文字数を指定します。つまり、指定した文字数以下の作品を抽出します。
length	int, string	作品の文字数を指定します。つまり、1000 を指定した場合、文字数が1000文字の作品を抽出します。minlen または maxlen との併用はできません。範囲指定する場合は、最小文字数と最大文字数をハイフン(-)記号で区切ります。また、minlen と maxlen を使わなくても「1000-」で1000文字以上、「-1500」で1500文字以下の作品を抽出することができます。

・会話率を指定して検索する：

kaiwaritu	int, string	作品の会話率を指定します。範囲指定する場合は、最低数と最大数をハイフン(-)記号で区切ります。「10-50」であれば会話率が10%~50%の作品を抽出します。また、「50-」で50%以上、「30」で30%の作品を抽出します。
-----------	----------------	--

・挿絵数を指定して検索する：

sasie	int, string	作品の挿絵の数を指定します。範囲指定する場合は、最低数と最大数をハイフン(-)記号で区切ります。「1-」であれば挿絵が1以上の作品を、「1-5」であれば挿絵が1~5の作品を抽出します。
-------	----------------	--

・読了時間を指定して検索する：読了時間とは作品文字数÷500を切り上げた数字のことです。単位は分です。なお、文字数を指定して検索する(minlen、maxlen、length)とは併用できません。

mintime	int	作品の最低読了時間を指定します。
maxtime	int	作品の最大読了時間を指定します。
time	int, string	作品の読了時間を指定します。mintime または maxtime との併用はできません。範囲指定する場合は、最小文字数と最大文字数をハイフン(-)記号で区切ります。「30-50」で読了時間が30~50分の作品を、「-40」で40分以内の作品を抽出します。

・Nコードを指定して検索する：Nコードとは作品に割り振られるコードのことです。

ncode	string	Nコードを指定します。ハイフン(-)記号で区切ればOR検索になります。開示設定が「検索除外中です」の作品は抽出できません。
-------	--------	---

・作品タイプを指定して検索する：

type	string	作品タイプを指定します。指定できるパラメータは以下の通りです。
------	--------	---------------------------------

		t：短編 r：連載中 er：完結済連載作品 re：すべての連載作品(連載中および完結済) ter：短編と完結済連載作品
--	--	---

・文体を指定して検索する：本機能は試験的に提供されているものなので正確に抽出できるとは限りません。ハイフン(-)記号で区切れれば OR 検索できます。

buntai	int, string	文体を指定します。指定できるパラメータは以下の通りです。 1：字下げされておらず、連続改行が多い作品 2：字下げされていないが、改行数は平均な作品 4：字下げが適切だが、連続改行が多い作品 6：字下げが適切でかつ改行数も平均な作品
--------	-------------	---

ちなみに、字下げとは行の先頭に空白を入れて行の開始位置をずらすことです。なお、具体的な改行数の基準は公開されていません。

・連載停止中かを指定して検索する：長期連載停止中作品についての機能です。なお、長期連載停止中の定義は「未完結の連載中作品で最終掲載日から 63 日以上経過しているもの」です。

stop	int	連載停止中について指定します。指定できるパラメータは以下の通りです 1：長期連載停止中を除きます 2：長期連載停止中のみ取得します
------	-----	---

・最終掲載日を指定して検索する：最終掲載日 general_lastup で抽出します。

lastup	string	最終掲載日について指定します。指定できるパラメータは以下の通りです。 thisweek：今週(日曜日の午前 0 時はじまり) lastweek：先週 sevenday：過去 7 日間(7 日前の午前 0 時はじまり) thismonth：今月 lastmonth：先月 タイムスタンプ：開始日と終了日をハイフン(-)記号で区切ることで UNIX タイムスタンプで抽出できます。
--------	--------	---

例えば、「thisweek」を指定することで今週新規投稿または次話投稿された作品を抽出します。

・最終更新日を指定して検索する：最終更新日 novelupdated_at で抽出します。

lastupdate	string	最終更新日について指定します。指定できるパラメータは以下の通りです。 thisweek：今週(日曜日の午前 0 時はじまり) lastweek：先週 sevenday：過去 7 日間(7 日前の午前 0 時はじまり)
------------	--------	---

		thismonth：今月 lastmonth：先月 タイムスタンプ：開始日と終了日をハイフン(-)記号で区切ることで UNIX タイムスタンプで抽出できます。
--	--	--

・ピックアップについて指定して検索する：小説ピックアップの対象となっている作品とは小説 PickUp！に掲載されている 1000 作品のことを指します。選出されるのは、オリジナルの短編作品か連載作品（完結済、連載中（文字数が 10 万文字以上））で、週間ユニークアクセス数や文字数等の情報を元に注目されていると判断された作品です。

ispickup	int	1 を指定すると小説ピックアップの対象となっている作品を抽出します。
----------	-----	------------------------------------

2-5：出力 GET パラメータ

最後に出力 GET パラメータについて見ていきます。出力 GET パラメータとは、なろう小説 API 自体の設定を行うパラメータのことです。

パラメータ	値	説明
gzip	int	ファイルに掛ける gzip 圧縮のレベルを 1～5 で指定できます。なお、1 は圧縮レベルが低く、5 は圧縮レベルが高いです。転送量を減らすために使用が推奨されています。
out	string	出力するファイルの形式を指定します。以下のパラメータが指定できます。 yaml：YAML 形式(デフォルト) json：JSON 形式 PHP：PHP の serialize() atom：Atom フィード jsonp：JSONP 形式 サンプルコードでは JSON 形式が使用されています。そのため、この同人誌では JSON 形式のみ説明します。JSON 形式については「4-4：出力と JSON 形式について」で説明します。
of	string	出力する項目を指定します。未指定時は全項目を出力します。転送量軽減のため、使用が推奨されています。複数項目を出力する場合は-で区切ります。詳しい項目は「2-3：of パラメータ」を確認してください。
lim	int(1～500)	最大出力数を指定します。1～500 を指定できます。指定しない場合は 20 になります。
st	int(1～2000)	表示開始位置を指定します。例として 3 を指定すれば 3 作品目以降の作品情報を取得します。一度に出力できる作品数は lim の最大値である 500 作品が限度です。そこで、st で 501 番目を指定することで 501 番～1000 番までの作品を取得することができるようになります。これを繰り返すことで最

		<p>大 2000 作品まで取得することができます。なお、一度に 2000 作品以上取得しようとするエラーになってしまうため、ひと工夫が必要になります。発展的なソースコードでその件について触れます。</p> <p>※2000+500 で 2500 まででは？ と思うかもしれませんが、1 回に取得できる数が 2000 までのようで、allcount が 2078 の場合でも取得しようするとエラーになることを確認しています。</p>
order	string	<p>出力順序を指定します。指定しない場合は新着更新順になります。指定できるパラメータは以下の通りです。</p> <p>new：新着更新順</p> <p>favnovelcnt：ブックマーク数の多い順</p> <p>reviewcnt：レビュー数の多い順</p> <p>hyoka：総合ポイントの高い順</p> <p>hyokaasc：総合ポイントの低い順</p> <p>dailypoint：日間ポイントの高い順</p> <p>weeklypoint：週間ポイントの高い順</p> <p>monthlypoint：月間ポイントの高い順</p> <p>quarterpoint：四半期ポイントの高い順</p> <p>yearlypoint：年間ポイントの高い順</p> <p>impressioncnt：感想の多い順</p> <p>hyokacnt：評価者数の多い順</p> <p>hyokacntasc：評価者数の少ない順</p> <p>weekly：週間ユニークユーザの多い順。毎週火曜日早朝リセット(前週の日曜日から土曜日分)</p> <p>lengthdesc：作品本文の文字数が多い順</p> <p>lengthasc：作品本文の文字数が少ない順</p> <p>generalfirstup：初回掲載順</p> <p>ncodedesc：N コード降順</p> <p>old：更新が古い順</p>

以上で、なろう小説 API の各パラメータの説明は終わりました。ソースコードを見ながら実際にどうやって API を PHP と Python で実行するのかを確認していきましょう。

第3章：なろう R18 小説 API について

なろう R18 小説 API：<https://dev.syosetu.com/xman/api/>

なろう R18 小説 API ではノクターンノベルズ（男性向け）、ムーンライトノベルズ（女性向け・BL）、ミッドナイトノベルズ（大人向け）に掲載されている R18 作品の情報を取得できます。ムーンライトノベルズは女性向けと BL に分けて作品情報を取得できます。

今後の説明では、なろう小説 API と同じパラメータについては省略し、特筆事項のみ説明します。全体の情報については URL から公式の説明を確認してください。

3-1：出力見本

API の URL：<https://api.syosetu.com/novel18api/api/?gzip=5&out=json&lim=1&order=hyoka>

```
[{'allcount': 132321},
{'all_hyoka_cnt': 9154,
'all_point': 86415,
'daily_point': 56,
'end': 1,
'fav_novel_cnt': 67596,
'general_all_no': 390,
'general_firstup': '2016-10-13 21:32:11',
'general_lastup': '2023-08-15 05:00:00',
'gensaku': '',
'global_point': 221607,
'impression_cnt': 12593,
'isbl': 0,
'isgl': 0,
'isstop': 1,
'istenni': 1,
'istensei': 0,
'iszankoku': 0,
'kaiwaritu': 21,
'keyword': '異世界転移 オリジナル戦記 冒険 チート 男主人公 紳士 異世界娼館',
'length': 2273143,
'monthly_point': 364,
'ncode': 'N8321DO',
'nocgenre': 1,
'novel_type': 1,
```

```

'novelupdated_at': '2024-11-17 09:06:23',
'quarter_point': 1054,
'review_cnt': 26,
'sasie_cnt': 0,
'story': '工事現場で事故死した主人公は、謎の存在から回復系チートを与えられ、異世界へと転移する。\\n'
        '転移した先は、剣と魔法のファンタジー世界。\\n'
        '彼はその地で人々と触れ合い、救い救われ、時に戦う。\\n'
        '様々な経験を積み成長して行く主人公。\\n'
        'そんな彼と人々との深いつながりを綴った一大叙事詩が、今ここに幕を上げる。',
'time': 4547,
'title': 'せっかくチートを貰って異世界に転移したんだから、好きなように生きてみたい',
'updated_at': '2025-01-31 12:06:35',
'weekly_point': 118,
'writer': 'ムンムン',
'yearly_point': 4720}}

```

3-2：出力

なろう R18 小説 API では 3 つの R18 サイト（ノクターンノベルズ、ムーンライトノベルズ、ミッドナイトノベルズ）を一つのデータベースとしてみて検索します。そのため、どのサイトで掲載されたのかを見るために nocgenre が用意されています。一方で、なろう小説 API で存在した isr15 や biggenre（大ジャンル）、genre（ジャンル）が削除されていることに注意してください。

要素	説明
nocgenre	掲載サイトを意味する数値を返します。数値は以下のとおりです。 1：ノクターンノベルズ(男性向け) 2：ムーンライトノベルズ(女性向け) 3：ムーンライトノベルズ(BL) 4：ミッドナイトノベルズ(大人向け)

3-3：of パラメータ

出力する項目を個別に指定するためのパラメータです。

パラメータ	出力要素	説明
ng	nocgenre	掲載サイト

3-4：条件抽出 GET パラメータ

出力の項で触れましたが、なろう R18 小説 API は 3 つのサイトをまとめて検索します。そのため、個々のサイトについて検索したい場合は、nocgenre か notnocgenre を使用します。

- ・掲載サイトを指定して検索する

nocgenre	int,string	検索する掲載サイトを指定します。
notnocgenre	int,string	指定した掲載サイトを除外検索できます。

以下のパラメータが使用できます。どちらも、ハイフン(-)記号で区切ることで複数サイトを指定できます。

- 1：ノクターンノベルズ(男性向け)
- 2：ムーンライトノベルズ(女性向け)
- 3：ムーンライトノベルズ(BL)
- 4：ミッドナイトノベルズ(大人向け)

- ・XID を指定する

xid	int,string	検索したい XID を指定します。ハイフン(-)記号で区切ることで XID の OR 検索を行います。
-----	------------	---

XID とは R18 用のアカウントに設定されるもので、なろう小説 API におけるユーザーID とは別のものです。

- ・ピックアップ条件に当てはまるかどうかを指定して検索する

ispickup	int	<p>ピックアップ条件に当てはまるかを指定します。指定できるパラメータは以下の通りです。</p> <p>1：ピックアップ条件である最終掲載日 general_lastup から 60 日以内なおかつ「短編または完結済または 10 万文字以上の連載中」を満たす作品</p> <p>0：上記ピックアップ条件を満たさない作品</p>
----------	-----	---

3-5：出力 GET パラメータ

なろう小説 API と比べて order パラメータで利用できるパラメータが減っています。

order	string	<p>出力順序を指定します。指定しない場合は新着更新順になります。指定できるパラメータは以下の通りです。</p> <p>new：新着更新順</p> <p>favnovelcnt：R18 ブックマーク数の多い順</p> <p>reviewcnt：レビュー数の多い順</p> <p>hyoka：総合ポイントの高い順</p> <p>hyokacnt：評価者数の多い順</p> <p>lengthdesc：作品本文の文字数が多い順</p> <p>dailypoint：日間ポイントの高い順</p> <p>weeklypoint：週間ポイントの高い順</p> <p>monthlypoint：月間ポイントの高い順</p> <p>quarterpoint：四半期ポイントの高い順</p> <p>yearlypoint：年間ポイントの高い順</p> <p>lengthasc：作品本文の文字数が少ない順</p> <p>old：更新が古い順</p>
-------	--------	--

第4章：基礎知識

この同人誌では PHP と Python を使います。そのため、本章ではプログラミング初心者に向けて、この同人誌で紹介するソースコードの内容を読み解くために必要な最低限の知識について触れていきます。とはいえ、ざっくりとした説明になるので勉強するための検索ワードを紹介しているものだと捉えてください。なお、説明中に下線部が引かれているものは構文ではないことを意味しています。

最初に、なぜこの二つの言語を紹介するのかを説明します。PHP に関しては、なろうデベロッパーで掲載されているサンプルコードが PHP で記述されているためです。このサンプルコードを読み解くには PHP の知識が必要になります。次に Python ですが、これは開発環境（開発ソフト）が関係しています。本来はサンプルコードに従って PHP で書くのが順当なのですが、以下の3つの理由でこの同人誌では Python を使ってソースコードを作成していきます。1：（私の場合）PHP の開発環境をなろう小説 API でしか使わない。2：Python は Google Colaboratory を使えば開発環境を構築しなくていい。3：個人的には PHP よりも Python の練習をしたい。

4-1：プログラミング全般の知識

この節ではプログラミング初心者に向けて PHP と Python に共通することについて説明します。

・アルゴリズム

ソースコードは一行ずつ処理されますが、その処理する順番（＝我々が読む順番）には3つのルールがあります。

1つ目が順次構造です。これは、上から下に順番に処理していくもので、基本的なルールです。

2つ目が選択構造（分岐処理）です。これは、条件を満たすか満たさないかで処理を飛ばしたり飛ばさなかったりするものです。例えば、 $20 \geq A$ という条件の際、A が 30 だったらこちらのブロックを処理するけどあちらのブロックは飛ばして処理しない。A が 10 だったらあちらのブロックを処理するけどこちらのブロックは飛ばして処理しない。みたいな感じです。色々種類がありますが、今回は if 文しか使いません。

3つ目が反復構造（繰り返し処理）です。これは、条件を満たしている間は処理を繰り返す行うというものです。こちらも色々種類がありますが、今回は while 文と for 文を使います。

・変数

変数とは文字や数字を入れる箱です。正確には違いますがそういうことにしておいてください。

$A = 30$

この場合、A は変数で 30 を値と言い、変数 A に値 30 を代入すると言います。代入なのですが、今回は格納という言い方を多用します。

A に 30 が格納されているため次のようなこともできます。

$B = A + 5$

B には 35 が代入されます。ソースコードを読む際は、この変数には今何が格納されているのかに注意しましょう。書き出すのも手です。

・型

変数という箱には種類があり、それに合わせて入れることのできるモノも違います。この種類を型と言い、異なる型の変数に異なる型の値を入れるとエラーになります。異なる型を入れたい場合は型変換を行った上で代入します。

今回、覚えておかないといけない型は以下の 3 つです。

- ・int 型：整数型です。整数のみを入れることができ、1.5 のように小数点以下があるものは代入できません。

- ・float 型：浮動小数点型です。2.553 のように小数点以下があるものを代入できます。

- ・string 型：文字列型です。”こんにちは”のように文字列を代入できます。int 型、float 型は何もつけずに代入できますが、文字列の場合はダブルクォーテーション(” ”)かアポストロフィー(‘ ’)で文字列を囲むのがルールです。

・変数の四則演算

int 型、float 型は一般的な算数と同じです。//は PHP では使えません。

A : 5 B : 3

C = A + B C : 8 足し算

C = A - B C : 2 引き算

C = A * B C : 15 掛け算

C = A / B C : 1.6666666666666667 割り算

C = A // B C : 1 割り算の整数部分

C = A % B C : 2 割り算のあまり

ちょっと変わったものに次のようなものがあります。

B += A B : 8

これは、B = B + A と同じ処理です。

string 型でも足し算は使えます。

A : ”こんにちは” B : ”世界”

C = A + B C : ”こんにちは世界” Python の場合。

C = A . B C : ”こんにちは世界” PHP の場合。

string 型と int 型・float 型をつなげる時は int 型・float 型を string 型に合わせます。

A : 18 B : ”歳”

C = str(A) + B C : 18 歳 Python の場合。

C = (string)A . B C : 18 歳 PHP の場合。

これの応用としてよくこういう書き方を見ます。

A : 28

C = “今日で” + A + “日目” C : 今日で 28 日目 Python の場合。

これの何が嬉しいのかというと、変数 A の中身を変えることで 28 の部分だけを変更することができるようになるのです。そのため、“今日で 50 日目”のように状況に合わせた表示を簡単に実装することができるようになります。

• 配列

変数を複数つなげて一つの変数としたものが配列です。Python ではリストと言います。

配列名 = [値, 値]

と書き、具体的には次のようになります。

A = ["yokohama", "kawasaki", "shinbashi", "ueno"]

値のことを要素と言います。配列から要素を取り出したい場合には要素番号を指定することで取り出せます。要素番号は 0 番目から数えます。

B = A[0] B: "yokohama"

B = A[3] B: "ueno"

• 連想配列

配列の変化形で、要素番号を自由に付けられる配列です。Python では辞書型と言います。

連想配列名 = {キー:値, キー:値}

と書き、具体的には次のようになります。

A = {"りんご": "赤", "みかん": "オレンジ色", "バナナ": "黄色"}

値の取り出しは要素番号の代わりにキーを指定することでできます。

B = A["りんご"] B: "赤"

B = A["バナナ"] B: "黄色"

注意が必要なのは、キーを 0 から始まる連番にした場合には配列と扱い方が同じになることです。つまり、次のような配列と連想配列はどちらも同じコードで処理できます。

A = ["yokohama", "kawasaki", "shinbashi", "ueno"]

A = {"0": "yokohama", "1": "kawasaki", "2": "shinbashi", "3": "ueno"}

ちなみに、Python では、

配列名 = [値, 値]

連想配列名 = {キー:値, キー:値}

A = 配列名[要素番号]

A = 連想配列名[キー]

PHP では、

配列名 = [値, 値]

連想配列名 = [キー=>値, キー=>値]

A = 配列名[要素番号]

A = 連想配列名[キー]

と書きます。

• True と False

この二つを真偽値と言います。条件式のときに使い、条件式が正しければ True に、間違っていれば False になります。例えば、 $5 > 3$ は True で $3 > 5$ は False になります。

• if 文

条件を満たしたらこっちのブロックを、満たさなかったらあっちのブロックを処理してね。というような条件分岐処理を行うものです。

Python では、

```
if 条件式:
```

```
    処理 1
```

```
else:
```

```
    処理 2
```

PHP では、

```
if(条件式){
```

```
    処理 1
```

```
}else{
```

```
    処理 2
```

```
}
```

と書きます。

条件式が True だったら処理 1 を、条件式が false だったら処理 2 を実行します。

具体的に書くと次のようになります。

```
if A >= 30 :
```

```
    B = A + 5
```

```
else:
```

```
    B = A - 10
```

変数 A が 35 のとき B には 40 が、変数 A が 15 のとき B には 10 が代入されます。

最小の構成は次の通りです。

```
if 条件式:
```

```
    処理 1
```

この場合、条件式を満たさない時は何も処理を行いません。

条件式に使用できる比較演算子の中で特筆すべきものとして == があります。

$x == y$: x と y が等しければ True

• while 文

条件を満たしている間、付属しているブロック内の処理を繰り返します。基本的に条件式に使う変数をループ内で +1 などして変化させないと無限ループに入ります。延々と処理が終わらないなと思った場合は無限ループに入っていないかを確認したほうがいいです。

Python では以下のように書きます。

while 条件式:

処理

処理

条件式が True の場合に処理を実行しブロックの末尾まできたら条件式の判定を再び行います。よって、条件式が True であれば延々と処理を繰り返します。

具体的に書くと次のようになります。

A = 5

B = 0

while A > 0:

B += A * 2

A = A - 1

最終的に A は 0、B は 30 になります。

・for 文・foreach 文

指定回数だけ繰り返し処理を行いたい時や配列の長さ（要素の数）だけ繰り返し処理を行いたい時に使うのが for 文です。今回は回数を指定するものはでてこないため、配列の長さ分の繰り返し処理を行う方法のみ説明します。

Python では以下のように書きます。

for 変数 in 配列:

処理

具体的に書くと次のようになります。配列の要素を一つずつ取り出して処理を行い、全ての要素を取り出すとループを終了します。

A = [3,5,7]

for i in A:

i += 1

print(i)

print(i)は変数 i の中身を出力するものです。末尾で改行するので、実行結果は

4

6

8

になります。配列 A の要素が一つずつ取り出されて処理されているのがわかります。

PHP では for 文と foreach 文の二つがあります。上で説明したような使い方をするのが foreach 文です。一方、for 文は配列の要素数を取得してから個数を数えながらループを回すモノです。今回は foreach 文しか使いません。PHP では以下のように書きます。

foreach(配列名 as 値){

処理

}

連想配列の場合は以下のように書きます。

```
foreach(配列名 as キー名 => 値){  
    処理  
}
```

Python に foreach 文はなく for 文を使います。

・関数

入力を与えると中でなんらかの処理を行った後に出力を返す機能の塊のことです。数学の関数をイメージするといいいかもしれません。x が決まれば y が確定するというやつです。

Python では以下のように書きます。

```
def 関数名(引数 1,引数 2):  
    処理  
    return 戻り値
```

使用方法是次の通りです。関数を呼び出すと言います。

```
A = 関数名(引数 1,引数 2)
```

A には戻り値が代入されます。

具体的に書くと次のようになります。

```
def Tashizan(A,B):
```

```
    C = A + B
```

```
    return C
```

使用する際は次のようになり、

```
Sum = Tashizan(3,6)
```

Sum には 9 が代入されます。

関数は自分で書くよりも他人が書いた関数を利用する場面が非常に多いです。というのも、Python のプログラミングでは他人が書いた関数の集まりであるライブラリを使うことが前提になっているからです。ライブラリは複雑な処理を関数にまとめてくれているので、自分で書く手間をなくし関数を呼び出すだけで問題を処理してくれます。例えば、

```
data = gzip.decompress(data)
```

という行が今後出てきます。gzip.decompress() は gzip 圧縮されたデータを引数に与えることで解凍したデータを戻り値として返してくれる関数です。私は gzip 圧縮されたデータをどう処理すれば解凍できるのかは知りません。しかしながら、gzip.decompress() の引数に圧縮されたデータを渡せば、関数が良い感じに処理してくれて解凍されたデータが返ってくることは知っています。

ちなみに、読み方というか捉え方ですが、ピリオド(.) は「の」と捉えると飲み込みやすくなります。つまり、gzip.decompress() は gzip の decompress 関数（機能）なんだなと捉えるということです。ただし、PHP の文字列の結合にもピリオド(.) が使われるため、混同しないようにしましょう。

・変数のスコープ

関数に関する話ですが、変数には影響範囲というものがあります。要するに、関数内で宣言した変数は関数の中でしか扱えないということです。以下で具体的に見ていきます。

```
A = "ABC"
def Syori():
    A = "123"
    return
Syori()
print(A)
```

とした場合、print(A)はABCを出力します。関数Syori内の変数Aは関数内で新しく宣言された変数のため、"ABC"を更新しません。ちなみに、if文やfor文ではスコープを考える必要はありません

・コメントアウト

コメントアウトをすることで、その行は処理されなくなります。Pythonでは#で、PHPは//で使えます。使用目的としては、ソースコードにメモを書き込んだり、後で使うかもしれないのでその行を削除したくない場合に一時的にその行を無効にする目的で使います。

具体例を挙げると、

```
A = 5
B = 2
C = 0
#処理を始める
#C = A * B
C = C + B  #終了
```

とすると、Cに代入されるのは12ではなく2になります。また、#より後ろが無効化されるので処理と同じ行にコメントアウトすることもできます。

4-2：PHPのルール

PHPのソースコードを書く際にはソースコード本体を「<?php」と「?>」で囲みます。PHPはHTMLの中で使用されることもあるため、タグを付けてPHPの範囲を示す必要があるのです。また、PHPは行の終わりを「;」で終える必要があったり、変数や配列などの名前には\$をつけます。

4-3：Pythonのルール

Pythonでは行の前に入るインデント（空白）が重要になります。文末に印をつけない代わりにインデントの個数で同じ行かを判断しているためです。

Pythonにはライブラリというものがあります。ライブラリとは、関数をまとめたものでユーザーから見れば新しく機能を追加するものです。イメージとしては、ゲーム本体に対するDLCやMODのようなものです。ライブラリの他にモジュールやパッケージという言葉がありますが、これらは、ライブラリ⇄パッケージ⇄モジュールという関係です。

4-4：出力と JSON 形式について

JSON 形式とは、データを記述するルールのこと、似たようなものに CSV や XML などがあります。JSON 形式の記述の仕方は以下の通りです。

値が一つの場合、

```
{“key” : “value”}
```

複数の値を持つ場合の場合、

```
{“key1” : “value1”, “key2” : “value2”}
```

複数の行を持つ場合、

```
[  
{“key1” : “value1”, “key2” : “value2”}  
{“key1” : “value1”, “key2” : “value2”}  
]
```

複数の行を持つ場合のイメージとしては、配列の値として連想配列が格納されている感じです。

では、実際のなろう小説 API の出力を見てみましょう。以下は 2023 年 1 月 1 日から 2024 年 1 月 1 日の期間で総合ポイント上位 3 作品の情報を一部出力したものです。分かりやすくするために適宜改行を入れています。具体的なパラメータは次の通りです。

```
"gzip":5, "out":"json", "lim":3, "order":"hyoka", "lastup":（省略）, "of":"t-n-k-nt"
```

```
[
```

```
{'allcount': 88184},
```

```
{'keyword': 'R15 残酷な描写あり 異世界転生 異世界転移 オリジナル戦記 冒険 男主人公 学園 ハー  
レム 魔法 チート ハッピーエンド 冒険者 貴族 アニメ化 自重知らず 内政 書籍化',
```

```
'ncode': 'N4936DP',
```

```
'noveltype': 1,
```

```
'title': '転生貴族の異世界冒険録～自重を知らない神々の使徒～【アニメ化決定】'},
```

```
{'keyword': '男主人公 ハーレム 冒険 日常 ゲーム ダンジョン VRMMO 主人公最強 スキル 女騎士  
獣人 ダークエルフ メイド 奴隷 精霊 R15 残酷な描写あり 異世界転生 異世界転移',
```

```
'ncode': 'N6774EH',
```

```
'noveltype': 1,
```

```
'title': '元・世界 1 位のサブキャラ育成日記 ～廃プレイヤー、異世界を攻略中！～'},
```

```
{'keyword': 'R15 残酷な描写あり 異世界転生 異世界 転生 日進月歩の成長 アイテムチート？ いつ  
かハーレム？\u3000迷宮 治癒士\u3000ドM\u3000ゾンビ 物体X 祓魔師\u3000聖変 精霊\u3000  
転生龍 ネット小説大賞 おっさんハーレム',
```

```
'ncode': 'N8697CX',
```

```
'noveltype': 1,
'title': '聖者無双\u3000～サラリーマン、異世界で生き残るために歩む道～'}

]
```

一つずつ見ていきましょう。まず、配列の 0 番目に allcount が格納された連想配列があります。これは、検索した条件で HIT した作品数を意味しています。2023 年 1 月 1 日から 2024 年 1 月 1 日の期間で 88184 作品が投稿されていることがわかります。最大出力数の 2000 より大幅に大きいですから、すべての作品情報を取得するためには一工夫が必要になります。

配列の 1 番目から作品情報が格納された連想配列が続きます。今回は of パラメータで t-n-k-nt を指定したため、t: タイトル、n: N コード、k: キーワード、nt: ノベルタイプ（連載か短編か）が出力されています。なお、\u3000は全角空白を意味します。ちなみに、\u200 は半角空白です。

JSON 形式から作品情報を取得する場合は、まず配列の要素番号を指定して各作品の情報が格納されている連想配列を取り出します。その後、連想配列のキーを指定して要素を取り出すことになります。つまり、配列名[1]['title'] とすれば、要素として「転生貴族の異世界冒険録～自重を知らない神々の使徒～【アニメ化決定】」を取得することができます。

4-5 : Google Colaboratory について

Google Colaboratory（以下 Google Colab）とは Google が提供しているオンラインの Python 実行環境のことです。Google ドライブで開発環境を作成することができます。無料で面倒な設定はせずに Python が実行できることとオンライン環境故に自分の PC スペックに制限されないのが魅力です。

スマホでも Google Colab は使用できます。ただし、Gogle ドライブアプリでは Google Colab は開きません。スマホで使う場合は、Gogle ドライブをアプリではなくブラウザから開くことで Google Colab を使用することができます。具体的な方法は以下の通りです。

各検索アプリで表示をスマホ用のサイトから PC 版のサイトに切り替えます（Safari: 「デスクトップ用 Web サイトを表示」、Chrome 「PC 版サイトを見る」）。この状態で検索から Gogle ドライブにログインすることでブラウザから Gogle ドライブを開くことができます。

Google ドライブ上で新規>その他>Google Colaboratory で Google Colab を新規作成することができます。開くとブロックがあり、そこにソースコードを書いていくことになります。ソースコードを書いたらブロックの左上にある再生ボタン (▶) みたいなアイコンをクリックすることでソースコードを実行できます。初回実行時は接続作業があるため少し時間が掛かります。

他の詳しいことはググってください。

第5章：公式のサンプルコード解説

なろうデベロッパーのサイトには二つのサンプルコードが掲載されています。具体的には、「作品簡易一覧(PHP)」と「ページ分割対応作品簡易一覧(PHP)」です。この章ではこの二つのサンプルコードの解説を行います。

なろうデベロッパーにおけるサンプルコードの位置は以下の通りです。

なろうデベロッパー>技術マニュアル>API 利用マニュアル[要プログラミング]>なろう小説 API>サンプルコード：作品簡易一覧(PHP)とサンプルコード：ページ分割対応作品簡易一覧(PHP)

PHP をオンラインで実行できるサイト (paiza.io、3v4l.org) があるのですが、どちらのサンプルコードもエラーが発生したことを報告させていただきます。

5-1：なろう小説 API サンプルプログラム「作品簡易一覧(PHP)」

「作品簡易一覧(PHP)」はなろう小説 API を利用して作品のタイトル、あらすじ、URL を表示するシンプルな PHP スクリプトです。」(公式サイトより引用)

URL：<https://dev.syosetu.com/man/sample01/>

```
01:  <?php
02:
03:  // なろう小説 API サンプルプログラム「作品簡易一覧(PHP)」
04:
05:  // API のオプションパラメータを連想配列で指定
06:  $params["out"] = "json";
07:  $params["lim"] = 100;
08:  $params["gzip"] = 5;
09:
10:  // API のエンドポイント(URL)を指定
11:  $url = "https://api.syosetu.com/novelapi/api/" . http_build_query($params, null, "&");
12:
13:  // ユーザエージェントが設定されていないクライアントからのアクセスは 403 エラーを返す
14:  // そのため、未設定の場合は PHP を名乗る
15:  if(empty(ini_get("user_agent"))){
16:      $options["http"]["header"][] = "User-Agent: PHP";
17:      $context = stream_context_create($options);
18:  }else{
```



```

19 :     $context = null;
20 : }
21 :
22 : // API を利用してデータを取得
23 : $file = file_get_contents($url, false, $context);
24 :
25 : // 取得した圧縮データを解凍
26 : $file = gzdecode($file);
27 :
28 : // JSON 形式のデータをデコード
29 : $listarray = json_decode($file, true);
30 :
31 : echo "▼抽出した作品の一覧<br /><br /><br />" . PHP_EOL;
32 :
33 : // 要素 0 には対象作品の総数が入るので破棄
34 : unset($listarray[0]);
35 :
36 : // 作品毎に表示
37 : foreach($listarray as $key => $value){
38 :     // タイトルを表示
39 :     echo "<b>" . $value["title"] . "</b><br />" . PHP_EOL;
40 :
41 :     // あらすじを表示
42 :     echo nl2br($value["story"]) . "<br />" . PHP_EOL;
43 :
44 :     // 作品の URL を指定
45 :     $novelurl = "https://ncode.syosetu.com/" . strtolower($value["ncode"]) . "/";
46 :
47 :     // 作品のリンクを表示
48 :     echo "<a href=\"\" . $novelurl . \"\">" . $novelurl . "</a>" . PHP_EOL;
49 :
50 :     echo "<hr />" . PHP_EOL;
51 : }
52 :
53 : ?>

```

01: PHP のお約束でソースコードの前後を「<?php」と「?>」で囲みます。

03: //はコメントアウトです。以下//の行の説明は飛ばします。

06~08：出力 GET パラメータの設定を行います。params は連想配列型です。

06：連想配列 params にキー：out、値：json を追加しています。つまり、出力 GET パラメータの out に json を指定しています。よって、出力の形式が JSON 形式になります。

07：連想配列 params にキー：lim、値：100 を追加しています。つまり、出力 GET パラメータの lim に 100 を指定しています。よって、出力される作品情報の数が 100 個になります。※HIT 数である allcount が先頭に追加されるため、連想配列 params の要素数は 101 個になります。

08：連想配列 params にキー：gzip、値：5 を追加しています。つまり、出力 GET パラメータの gzip に 5 を指定しています。よって、gzip 圧縮レベル 5 で圧縮された gzip ファイルを出力します。

11：なろう小説 API を使用する場合、URL を使って情報を取得します。この行では API の利用に必要な URL を作成しています。具体的には、連想配列を URL エンコードされたクエリ文字列に変換し、API の URL を作成しています。

URL エンコードとは URL で使用できない文字を URL で使用できる文字に変換することです。また、クエリ文字列とは、URL の後ろに追加される「?項目 1=値 1&項目 2=値 2」形式の文字列のことです。役割としては、クエリ文字列を URL の後ろに追加することで API に情報を伝えることができます。今回は、出力 GET パラメータ等の情報を API に伝える役割を果たします。

ソースコードを見ていきましょう、ピリオド (.) は PHP においては文字列連結でした。そのため、"https://api.syosetu.com/novelapi/api/?" と http_build_query(\$params, null, "&") を足し合わせたものを変数 url に格納しています。前半の "https://api.syosetu.com/novelapi/api/?" はなろう小説 API の URL の末尾に ? を追加したものです。

そして、http_build_query(\$params, null, "&") は、与えられた連想配列から URL エンコードされたクエリ文字列を生成する関数です。第一引数に変換したい連想配列 params を渡しています。第二引数は渡した配列に数値が使われている場合に指定した数値を付け足すかどうかを意味します。今回は null なので何も付け足しません。第三引数にはデータの区切り文字を指定します。初期値は & ですが、明示的に & を指定したほうが環境依存にならず安全だそうです。第四引数もあるのですが、省略されています。結果として変数 url には以下の文字列が格納されることになります。

```
https://api.syosetu.com/novelapi/api/?gzip=5&out=json&lim=100
```

15~20：API を叩くときにユーザーエージェントという値が設定されていない場合、API は 403 エラーを返します。そのため、ユーザーエージェントが未設定の場合は PHP を設定することにします。ちなみに、ユーザーエージェントとは、ホームページを見る際にユーザーから web サーバーに送られる情報の一つであり、どんなソフトや機械を使ってアクセスしているのかを送ります。

15：if 文です。empty(ini_get("user_agent")) が True の場合に 16~17 行目が実行され、False の場合に 19 行目が実行されます。empty(ini_get("user_agent")) を見ていきましょう。

まずは、(ini_get("user_agent"))の方から見ていきます。ini_get()は設定オプションの値を得る関数ですので、引数 user_agent から user_agent の情報を取得していることが分かります。

次に、empty()は、変数が存在しない場合や、値が空と見なされる場合に True を返します。つまり、empty(ini_get("user_agent"))は user_agent が設定されていない場合に True を返します。

16~17: user_agent を PHP に設定しています。

16: 連想配列 options に文字列"User-Agent: PHP"を格納しています。

17: 変数 context に stream_context_create(\$options)の値を格納しています。

stream_context_create()は、データの入出力を管理するストリームフィルタの振る舞いの設定を変更できるストリームコンテキストを作成する関数です。※よく分かってないです。

19: 変数 context に null を格納しています。

23: API を利用してデータを取得しています。変数 file に file_get_contents(\$url, false, \$context) で取得したデータを格納しています。file_get_contents()は変数 url で指定した場所にあるファイルを取得します。これでデータの取得は終わりました。残りは取得したデータの処理と表示に関する処理になります。

26: gzip 圧縮されたファイルを解凍しています。出力 GET パラメータの gzip を指定した場合 (08 行目)、なろう小説 API は gzip 圧縮された gzip ファイルを出力します。gzip ファイルはそのままでは扱えないため、gzdecode()関数で gzip ファイルを解凍しています。解凍したファイルは出力 GET パラメータの out で指定した (06 行目) 形式になります。正確には out で指定した出力形式が gzip 圧縮されているため、それを解凍するという流れです。

29: JSON 形式のデータを配列に変換しています。JSON 形式そのままでは扱いづらいため、json_decode()で適切な PHP の型 (今回は配列) に変換しています。変換した配列を変数 listarray に格納しています。

31: echo は文字列を出力する機能です。Python における print()と同じです。ここでは、.によって "▼抽出した作品の一覧

"と PHP_EOL が結合されたものを出力しています。
は html における改行コードタグです。
が3個あるため、html 上での表示では「▼抽出した作品の一覧」の下に改行が3回入ることになります。また、PHP_EOL は、OS や環境に依存しない改行コードのことです。コンソール出力の時に改行しブラウザ上 (html) では改行しません。

34: 出力されたデータの1番目 (プログラム上では0番目) には作品情報ではなく、allcount が格納されています。for 文などで繰り返し処理を行う場合、一つだけ例外が存在すると処理が面倒くさいことになります。そこで、以後使わない allcount を削除し、作品情報のみのデータにしています。

unset()関数は指定した要素を削除する機能で、今回は配列 listarray の0番目の要素 (= allcount) を削除しています。

37~51: 取得した作品情報のタイトル、あらすじ、リンクを作品ごとに出力しています。

37: foreach 文は連想配列の要素の数だけ処理を繰り返します。

foreach(\$listarray as \$key => \$value)は、配列 listarray の要素を一つずつ取り出し、要素のキーを変数 key に値を変数 value に格納します。連想配列として処理を行っていますが、キーが0から始

まる連番なので配列と区別がつかないタイプです。また、配列 listarray の要素は値として作品情報を格納した連想配列であるため、変数 value には連想配列が代入されます。

39：作品のタイトルを太文字で出力しています。

\$value["title"]でキーに title（タイトル）を指定して値を取得しています。

html ではで囲むと文字列が太字になります。
と PHP_EOL は改行でした。

それらが、で結合され echo によって「タイトル名
」が出力されます。

42：作品のあらすじを出力しています。

value["story"]でキーに story（あらすじ）を指定して値を取得しています。

nl2br()は html 以外の改行コードを html の改行コードに変換する関数です。例えば、\n など改行を意味するコードなのですが、これを
に置き換えるということです。

それらが、で結合され echo によって「あらすじ
」が出力されます。

45：作品の N コードを取得し、作品の url を作成しています。

変数 novelurl に"https://ncode.syosetu.com/"と strtolower(\$value["ncode"])と"/"を結合したものを格納しています。

value["ncode"]でキーに ncode を指定して値（N コード）を取得しています。

strtolower()は文字列内の英文字をすべて小文字にする関数です。例えば、『とんでもスキルで異世界放浪メシ』の N コードを取得すると N2710DB です。これを n2710db に変換するということです。

小説家になろうにおいて作品ページの URL は

https://ncode.syosetu.com/N コード/

で管理されています。そこで取得した N コードを組み合わせて作成した作品ページの URL を変数 novelurl に格納しています。

48：作品の URL の文字列に作品のリンクをつけて出力しています。

HTML 上では、とで挟んだ文字列にリンクをつけることができます。

50：HTML の<hr />タグを出力しています。<hr />タグは横線を画面上に引きます。

51：37 行の foreach 文を閉じています。

53：PHP のルールでソースコードの前後を「<?php」と「?>」で囲みます。

5-2：なろう小説 API サンプルプログラム「ページ分割対応作品簡易一覧(PHP)」

「ページ分割対応作品簡易一覧(PHP)はなろう小説 API を利用して作品のタイトル、作者名、N コード、作品種別(エピソード数)、文字数、あらすじ、URL を表示するシンプルな PHP スクリプトです。」(公式サイトより引用)

URL：<https://dev.syosetu.com/man/sample02/>

001： <?php

002：

003： // なろう小説 API サンプルプログラム「ページ分割対応作品簡易一覧(PHP)」

```

004 :
005 : // API のオプションパラメータを連想配列で指定
006 : $params["out"] = "json";
007 : $params["word"] = "女主人公";
008 : $params["gzip"] = 5;
009 : $params["order"] = "generalfirstup";
010 :
011 : // 1 ページあたりの表示件数
012 : $lim = 20;
013 :
014 : // ページ番号を取得
015 : $p = intval($_GET["p"]);
016 :
017 : // もしページ番号が1未満なら1を代入
018 : if($p < 1){
019 :     $p = 1;
020 : }
021 :
022 : // 位置を決定
023 : $st = ($p - 1) * $lim;
024 :
025 : // st は 0 はじまりではなく1始まりなので1を加算
026 : $st++;
027 :
028 : // API のオプションパラメータに st を追加
029 : $params["st"] = $st;
030 :
031 : // API のオプションパラメータに lim を追加
032 : $params["lim"] = $lim;
033 :
034 : // API のエンドポイント(URL)を指定
035 : $url = "https://api.syosetu.com/novelapi/api/" . http_build_query($params, null, "&");
036 :
037 : // ユーザエージェントが設定されていないクライアントからのアクセスは 403 エラーを返す
038 : // そのため、未設定の場合は PHP を名乗る
039 : if(empty(ini_get("user_agent"))){
040 :     $options["http"]["header"][] = "User-Agent: PHP";

```

```

041 :     $context = stream_context_create($options);
042 : }else{
043 :     $context = null;
044 : }
045 :
046 : // API を利用してデータを取得
047 : $file = file_get_contents($url, false, $context);
048 :
049 : // 取得した圧縮データを解凍
050 : $file = gzdecode($file);
051 :
052 : // JSON 形式のデータをデコード
053 : $listarray = json_decode($file, true);
054 :
055 :     echo "全抽出作品数：" . number_format($listarray[0]["allcount"]) . "<br />" .
        PHP_EOL;
056 :     echo "ページ番号：" . $p . "<br />" . PHP_EOL;
057 :     echo "▼抽出した作品の一覧<br /><br /><br />" . PHP_EOL;
058 :
059 : // 全抽出作品数を別の変数に退避し、不要になった要素 0 を破棄
060 : $allcount = $listarray[0]["allcount"];
061 : unset($listarray[0]);
062 :
063 : // 作品毎に表示
064 : foreach($listarray as $key => $value){
065 :     // タイトルを表示
066 :     echo "<b>" . $value["title"] . "</b><br />" . PHP_EOL;
067 :
068 :     // 作者を表示
069 :     echo "作：" . $value["writer"] . " N コード：" . $value["ncode"] . "<br />" .
        PHP_EOL;
070 :
071 :     // 作品種別と連載の場合は全掲載エピソード数を表示
072 :     if($value["novel_type"] == 2){
073 :         echo "短編<br />" . PHP_EOL;
074 :     }else{
075 :         // novel_type が 2 以外、つまり 1 なら連載
076 :         if($value["end"] == 1){

```

```

077 :         echo "連載中(全" . $value["general_all_no"] . "エピソード)<br />" . PHP_EOL;
078 :     }else{
079 :         echo "完結済(全" . $value["general_all_no"] . "エピソード)<br />" . PHP_EOL;
080 :     }
081 : }
082 :
083 : // 文字数を表示
084 : echo "文字数：" . $value["length"] . "文字<br /><br />" . PHP_EOL;
085 :
086 : // あらすじを表示
087 : echo nl2br($value["story"]) . "<br />" . PHP_EOL;
088 :
089 : // 作品の URL を指定
090 : $novelurl = "https://ncode.syosetu.com/" . strtolower($value["ncode"]) . "/";
091 :
092 : // 作品のリンクを表示
093 : echo "<a href=\"\" . $novelurl . "\">" . $novelurl . "</a>" . PHP_EOL;
094 :
095 : echo "<hr />" . PHP_EOL;
096 : }
097 :
098 : // 2 ページ目以降を表示中の場合は前のページへのリンクを表示
099 : if($p >= 2){
100 :     echo "<a href=\"\" . $_SERVER["PHP_SELF"] . "?p=" . ($p - 1) . "\">前のページ</a>" .
101 :     PHP_EOL;
102 : }
103 : echo "<br />" . PHP_EOL;
104 :
105 : // 次のページで表示する作品があれば次のページへのリンクを表示
106 : if(($p * $lim) < $allcount){
107 :     echo "<a href=\"\" . $_SERVER["PHP_SELF"] . "?p=" . ($p + 1) . "\">次のページ</a>" .
108 :     PHP_EOL;
109 : }
110 : ?>

```

5-1：なろう小説 API サンプルプログラム「作品簡易一覧(PHP)」で説明したものは簡潔に述べます。

001: PHP のお約束です。

005~032: 連想配列 params に出力 GET パラメータと条件抽出 GET パラメータを格納しています。

006: 「作品簡易一覧(PHP)」 06 行目と同じです。

007: 連想配列 params にキー: word、値: 女主人公を追加しています。つまり、条件抽出 GET パラメータの word に女主人公を指定しています。よって、タイトル、あらすじ、キーワード、作者名に「女主人公」が含まれる作品を検索ようになります。

008: 「作品簡易一覧(PHP)」 08 行目と同じです。

009: 連想配列 params にキー: order、値: generalfirstup を追加しています。つまり、出力 GET パラメータの order に generalfirstup を指定しています。出力の順番が初回掲載順になります。

012~032: 最大出力数を指定する lim と表示開始位置を指定する st を指定しています。

012: 変数 lim を宣言し、20 を代入しています。

015: 変数 p を宣言し、値を代入しています。

`$_GET["p"]`でページ番号を取得しています。`intval()`は渡された変数の中身の型を数値型に変換します。なお、null の場合は 0 に変換します。

018~020: 変数 p が 1 未満の場合、変数 p に 1 を格納します。

018: if 文で変数 p が 1 未満かを判定します。1 未満の場合 19 行を実行します。

019: 変数 p に 1 を格納しています。

023~026: 変数 p の数値 (ページ数) に合わせて出力の開始位置を調整しています。

動作のイメージとしては、1 ページ目は 1 番から変数 lim が意味する最大出力数までリストを出力します。例えば、lim が 20 なら、1 ページ目は 1~20 番目までの作品情報を出力するということです。すると、2 ページ目は 21~40 番目を出力することになります。そこで、1 ページ目には 1 を 2 ページ目には 21 を指定したいわけです。

023: 変数 st は出力 (表示) 開始位置を指定するパラメータです。変数 lim は最大出力数ですから、変数 p (ページ数) から 1 引いた数と掛け算することでそのページにおける出力開始位置が分かります。1 ページ目なら $(1 - 1) * 20 = 0$ 、2 ページ目なら $(2 - 1) * 20 = 20$ ですね。

026: ただ、st は 0 始まりではなく 1 始まりなので `st++` で変数 st に 1 を追加しています。

`st++` は `st = st + 1` と同じです。

029: 連想配列 params にキー: st、値: 変数 st を追加しています。つまり、出力 GET パラメータの st に変数 st の値を指定しています。変数 st は可変でページ数によって変化します。

032: 連想配列 params にキー: lim、値: 変数 lim を追加しています。つまり、出力 GET パラメータの lim に変数 lim の値を指定しています。変数 lim は 20 だったので 20 個の作品情報を出力ようになります。

035~053：「作品簡易一覧(PHP)」の11~29行目とやっていることは同じなので省略します。

055：「全抽出作品数：allcount の数
」を出力します。

\$listarray[0]["allcount"]で作品情報が格納された連想配列から allcount を出力します。配列 listarray の 0 番目の要素である連想配列に allcount が格納されているのです。

number_format()は数値を 3 桁ごとのカンマ区切り文字に変換する関数です。例えば、数値の 1000000 を文字列の 1,000,000 に変換します。

056：「ページ番号：変数 p の中身
」を出力します。

057：「▼抽出した作品の一覧

」を出力します。

060~061：全作品出力数 allcount を別の変数に代入し、allcount を削除します。これにより、配列 listarray に作品情報だけが格納されるようになります。

060：変数 allcount に全作品出力数 allcount を代入しています。

061：allcount が格納されている配列 listarray の 0 番目の要素を削除しています。unset()は指定した配列の要素を削除する関数です。

064~096：取得した作品情報のタイトル、作者名、N コード、連載か短編か、連載なら完結済か連載中か、文字数、あらすじ、作品 URL を出力しています。なろう小説 API サンプルプログラム「作品簡易一覧(PHP)」の 37~51 行の処理と基本は同じです。

064：foreach()は配列の要素の数だけ処理を繰り返します。

foreach(\$listarray as \$key => \$value)は、配列 listarray の要素を一つずつ取り出し、要素のキーを変数 key に値を変数 value に格納します。配列 listarray の要素は値として作品情報を格納した連想配列を持っているため、変数 value には連想配列が代入されます。

066：作品のタイトルを太文字で出力しています。

\$value["title"]でキーに title (タイトル)を指定して値を取得しています。

html ではで囲むと文字列が太字になります。

と PHP_EOL は改行でした。

それらが、で結合され echo によって「タイトル名
」が出力されます。

069：作品の作者名と N コードを出力しています。

\$value["writer"]でキーに writer (作者名)を指定して値を取得しています。

\$value["ncode"]でキーに ncode (N コード)を指定して値を取得しています。

それらが、で結合され echo によって「作：作者名 N コード：N コード
」が出力されます。

072~080：novel_type (連載作品か短編作品か)を取得し、連載作品ならば end (完結済か連載中か)を取得・出力しています。

072：\$value["novel_type"]でキーに novel_type (連載作品は 1、短編作品は 2)を指定して値を取得しています。if 文で novel_type が 2 の場合は 073 行が実行されます。

073：echo によって「短編
」が出力されます。

074: novel_type がそれ以外=1 の場合は 076~080 行が実行されます。

076: \$value["end"]でキーに end (短編作品と完結済作品は 0、連載中は 1) を指定して値を取得しています。

if 文で end が 1 の場合は 077 行が実行されます。それ以外=0 の場合は 079 行が実行されます。

077: \$value["general_all_no"]でキーに general_all_no (エピソード数) を指定して値を取得しています。
.で結合され echo によって「連載中 (全エピソード数エピソード)
」が出力されます。

079: \$value["general_all_no"]でキーに general_all_no (エピソード数) を指定して値を取得しています。
.で結合され echo によって「完結済 (全エピソード数エピソード)
」が出力されます。

084: \$value["length"]でキーに length (作品文字数) を指定して値を取得しています。

.で結合され echo によって「文字数: 作品文字数文字

」が出力されます。

087: \$value["story"]でキーに story (あらすじ) を指定して値を取得しています。nl2br()は改行コードを
に変換するものでした。
.で結合され echo によって「あらすじ
」が出力されます。

090: \$value["ncode"]で N コードを取得しています。strtolower()は文字列の英文字をすべて小文字にする関数でした。変数 novelurl に「https://ncode.syosetu.com/N コード/」を代入しています。

093: 「https://ncode.syosetu.com/N コード/」に「https://ncode.syosetu.com/N コード/」のリンクを紐づけしています。

096: echo によって「<hr />」が出力されます。

099~108: 前のページと次のページのリンクを出力します。正直、あまり理解していません。

099: if 文で変数 p が 2 以上の場合 100 行を実行します。

100: echo によって「前のページ」が出力されます。「前のページ」に現在実行しているページの一つ前のリンクを紐づけしています。

103: echo によって「
」が出力されます。

106: まだ表示していない作品があるかを判定します。

if 文で変数 p と変数 lim を掛けたものが allcount 未満の場合 107 行を実行します。

要するに p (ページ数) と lim (表示件数) を掛けたものが allcount (HIT 数) より少ない=まだ表示していないものがある場合 107 行を実行するということです。

107: echo によって「次のページ」が出力されます。「次のページ」に現在実行しているページの一つ前のリンクを紐づけしています。

110: PHP のお約束です。

第 6 章：Python で作成したソースコード群

この章では、筆者が Python を使用して作成したソースコードを紹介していきます。

6-1 で紹介する基本：「条件を指定して検索結果を 3 件表示する」の機能は、基本的なもので、API のパラメータの設定を行った後、API を実行して返ってきた結果をそのまま表示するというものです。このソースコードで API を利用する際の基本を確認します。

6-2 で紹介する発展：「指定した期間内の総合評価の分布を調べる（なろう版）」では、指定した期間に投稿された短編作品の情報を取得し、総合評価ポイントの分布を調べます。「なろう小説 API」では同時に 2000 件以上の作品情報を取得するとエラーになるため、その回避法について述べます。

6-3 で紹介する発展：ソースコード「指定した期間内の総合評価の分布を調べる（ノクターンノベルズ版）」では、6-2 のソースコードをノクターンノベルズ用に改変して実行結果を確認します。

6-4 で紹介するその他：「なろうユーザ検索 API」では、「なろうユーザ検索 API」を使用したソースコードを紹介します。この同人誌では紹介しなかった他の API でもなろう小説 API の知識が使えることを確認します。

6-1：基本：「条件を指定して検索結果を 3 件表示する」

出力 GET パラメータ、条件抽出 GET パラメータ、of パラメータを指定して検索を行い、その検索結果を 3 つだけ表示するという基本的な動作を行うソースコードです。このソースコードを参考にして「なろう小説 API」を利用するための最低限の仕組みを理解しましょう。

全体の流れとしては、以下の通りです。

01：使用するライブラリを導入する。

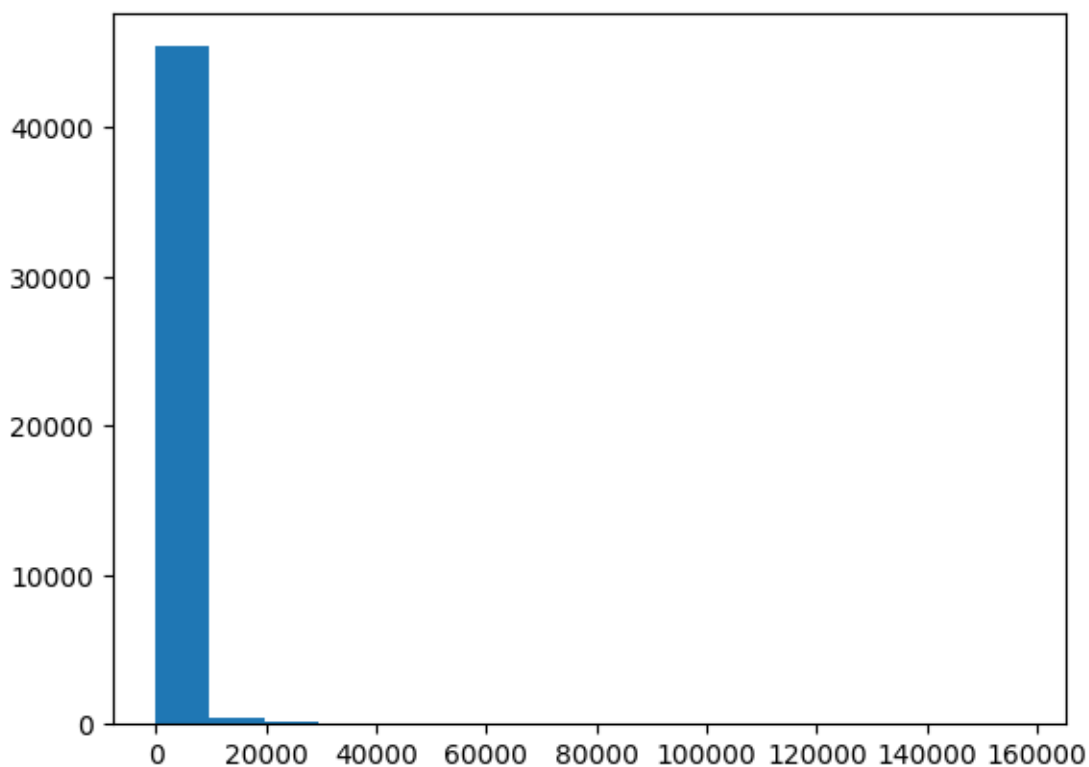
02：API の設定を行う（出力 GET パラメータ、条件抽出 GET パラメータ、of パラメータ）。

03：API を実行する。

04：API から返ってきたデータを表示する。

```
01:  #基本的なソースコード
02:  import datetime
03:  import requests
04:  import json
05:  import gzip
06:  import pprint
07:
08:  #API の URL
09:  url = "https://api.syosetu.com/novelapi/api/"
10:
11:  params = {}
```

転送量：36.42KB
作品数：46243
最大値：157182
最小値：0
平均値：615.43
中央値：10
最頻値：0
標準偏差：3640.57



下から 39790.5 番目
下から 86.05%
偏差値：48.79

以下でソースコードの詳細な説明を行いますが、基本的なソースコードで説明したものは簡潔に述べます。

003~012：import モジュール名でこのソースコードで使用するモジュール（ライブラリ）を導入しています。

003：datetime モジュールを導入しています。日時を扱うモジュールです。

004：requests ライブラリを導入しています。ウェブサイト情報を問い合わせるライブラリです。

005：json モジュールを導入しています。JSON 形式のファイルを扱うモジュールです。

006：gzip モジュールを導入しています。gzip 圧縮を扱うモジュールです。

007：pprint モジュールを導入しています。出力の print()を強化したものでリスト型や辞書型を見やすく出力してくれます。

6-3：発展的：「指定した期間内の総合評価の分布を調べる（ノクターンノベルズ版）」

「なろう小説 API」ではなく「なろう R18 小説 API」を使用してノクターンノベルズの総合評価ポイントの分布について調べます。基本的には、発展的：「指定した期間内の総合評価の分布を調べる（なろう版）」と変わりありません。なろう版からの変更点は以下の通りです。

- ・ API の URL の変更をするために 27 行目を
url = "https://api.syosetu.com/novel18api/api/"に変更します。
- ・ 掲載サイトを指定するため、42 行目の条件抽出 GET パラメータに("nocgenre","1")を追加します。

以上のソースコードを実行することで以下の出力を得ます（2025.04.08）。

2

1

一致している

取得件数：2695

転送量：4.08KB

作品数：2695

最大値：24958

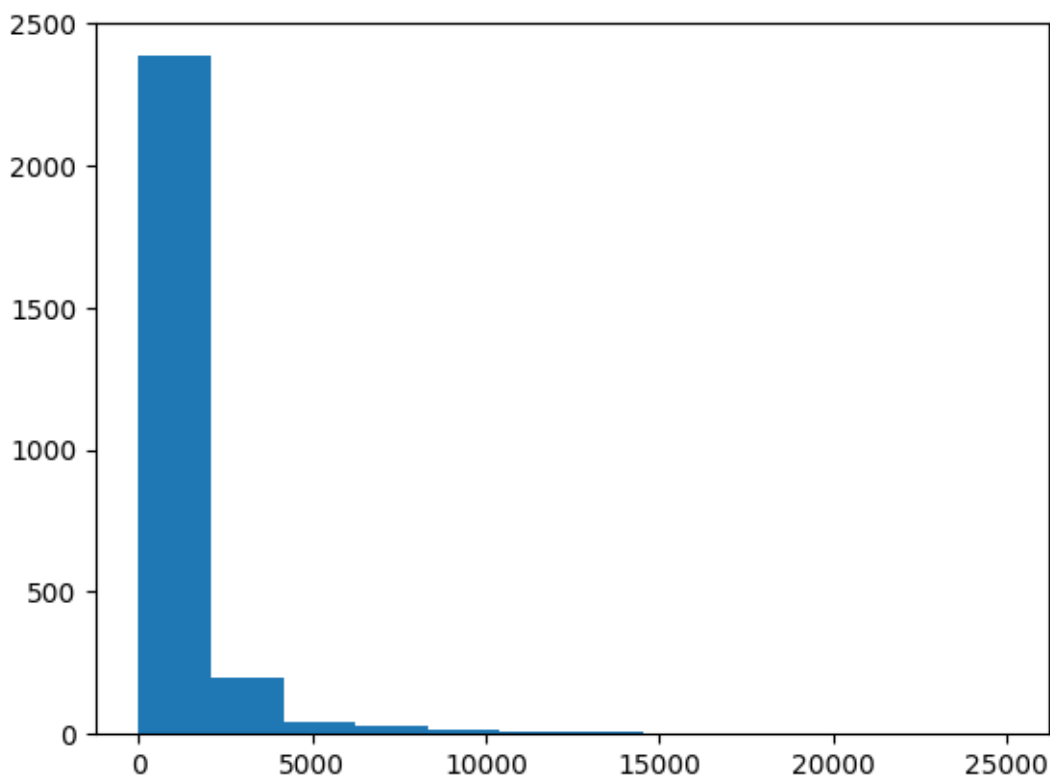
最小値：0

平均値：849.93

中央値：174

最頻値：0

標準偏差：1914.82



下から 1356 番目

下から 50.35%

偏差値：46.48

6-4：その他：「なろうユーザ検索 API」

「なろうユーザ検索 API」を使用して、投稿した作品数ごとのユーザー数を調べてみました。
ソースコードの大部分は既に説明しているものと変わらないので最低限の説明のみを行います。
なろうユーザ検索 API については、以下の URL から公式ドキュメントを確認してください。
なろうユーザ検索 API：<https://dev.syosetu.com/man/userapi/>

```
01:  #投稿した作品数ごとのユーザー数を調べる
02:
03:  import requests
04:  import json
05:  import gzip
06:
07:  #関数化パラメータ渡して辞書型を返す
08:  def NarouAPI(url, params):
09:      r = requests.get(url, params=params)
10:      data = r.content
11:      data = gzip.decompress(data)
12:      data = json.loads(data)
13:      return data
14:
15:  #本体
16:
17:  url = "https://api.syosetu.com/userapi/api/"
18:
19:  params = {}
20:  #出力 GET パラメータの設定
21:  params.update([("gzip",5),("out","json"),("lim",1)])
22:
23:  #条件抽出 GET パラメータの設定
24:  params.update([("minnovel",5),("maxnovel",5)])
25:
26:  #of パラメータの設定
27:  params.update([("of","sg")])
28:
```

```

29: data = NarouAPI(url,params)
30:
31: ZenCount = data[0]["allcount"]
32: print(ZenCount)

```

既に説明しているものについては省略します。

- ・17: API の URL を「なろうユーザ検索 API」の URL に変更します。
- ・24: 条件抽出 GET パラメータの作品投稿数を指定します。

- ・minnovel: 指定した数値以上の作品投稿数があるユーザーを検索します。
- ・maxnovel: 指定した数値以下の作品投稿数があるユーザーを検索します。

今回はどちらも 5 を指定しているため作品投稿数が 5 以上 5 以下 = 5 のユーザーを検索します。

- ・27: of パラメータを指定します。指定した sg は sum_global_point (総合評価ポイントの合計) を意味します。今回は allcount のみを使うため、of パラメータは通信量削減のために設定しています。

以上のソースコードで実行することで以下の出力を得ます (2025.03.30)。

8692

また、minnovel と maxnovel を変更して調べたところ、次のようなデータが得られました。

作品投稿数	ユーザー数	全ユーザーに対する割合	投稿者に対する割合
全ユーザー数	2,724,727	100%	
投稿作品が 00	2,398,750	88.04%	
投稿作品が 01 以上	325,977	11.96%	100%
投稿作品が 01	196,483	7.21%	60.28%
投稿作品が 02	51,663	1.90%	15.85%
投稿作品が 03	23,615	0.87%	7.24%
投稿作品が 04	13,226	0.49%	4.07%
投稿作品が 05	8,692	0.32%	2.67%
投稿作品が 06 以上	32,298	1.19%	9.91%

全ユーザーの 1 割が作品を投稿したことのあるユーザーであること、作品を投稿したことのあるユーザーの 6 割は投稿作品数が 1 であることが分かります。

6-5: これから

今までの説明で「なろう小説 API」を使って「小説家になろう」の生データを入手することができるようになりました。この後は、手に入れた生データをどのような目的でどうやって処理していくかが問題になります。しかしながら、これ以上は発展的なものであり、無数の目的に対応して様々な手法が存在します。入門講座であるこの同人誌ではデータの入手方法の説明までで終えたいと思います。この同人誌が皆さんの苦労を軽減できれば幸いです。