**技術課題**

和歌山大学大学院 システム工学研究科1年

鈴木 洸海

**問題1.** フィボナッチ数を返すAPIサービスの開発

　ソースコード一式とユニットテストの結果は資料の終わりに記載する。

　作成したサーバーのエンドポイントURL

http://localhost:5000/fib?n=<正の整数>

　エンドポイントの <正の整数> 部分を正の整数に置き換える必要がある。

　作成したプログラムはローカルサーバーなので、外部からアクセスできる仕

　様にはなっていない。また、ソースコードはPython言語を使用し .ipynb形式なので、Jypyter Notebook環境下で実行する必要がある。

　ソースコードの構成、概要を説明するドキュメント

ダイアグラム が含まれている画像

自動的に生成された説明

・エンドパラメータURL　**http://localhost:5000/fib?n=<正の整数>**

　　ローカルサーバー起動後、エンドパラメータURLを検索エンジンに入力する。

　・エンドポイント’/fib’におけるデコレータ　**@app.route(‘/fib’)**

　　Flaskフレームワークで使用されるデコレータである。Pythonの機能であり、関数やクラスの動作を修飾、拡張するために使用する。Flaskアプリケーション内の特定のエンドポイントに関連する関数を定義することができる。URLパス(‘/fib’)にアクセスされた場合、デコレータ内の関数が作動する。

　・リクエストパラメータを判別する関数　**get\_fibonacci()**

リクエストパラメータnに代入された値を判別する。

n の値が正の整数の場合は、フィボナッチ数を計算する関数に値を渡し、返されたフィボナッチ数と n の値をJSON形式で表示する。

nの値が正の整数以外の場合は、正の整数を入力するようにエラーが表示される。

リクエストパラメータが n 以外で宣言されている場合は、n を宣言するようにエラーが表示される。

　・フィボナッチ数を計算する関数　**fibonacci(n)**

n が正の整数の場合にn 列目のフィボナッチ数を返す。

　 フィボナッチ数の計算は次のように行う。

　　n 列目のフィボナッチ数は1列前の値と2列前の値を足した値である。

初項は1、2列目を1とし、n列目は再帰的に求める。求めたフィボナッチ

　　数を get\_fibonacci() に返す。

**問題2.** 文章問題

　自身の親に対して「WebブラウザがWebページを表示する仕組み」を700〜800文字で説明してください。Webブラウザのなかでも本質的に重要な挙動について、筋道を立て理解できるように文章を組み立ててみてください。

　まずは、WebブラウザがWebページを表示する仕組みについて説明した後に、各用語について説明します。Webブラウザはユーザーが指定したURLを基に、Webサーバーに対してHTTPリクエストを送信します。Webサーバーはリクエストを受け取り、要求されたWebページのデータを含むHTTPレスポンスを返し、最終的にWebブラウザはユーザーが見ることができるようにHTMLを解釈してWebページを表示します。

　Webブラウザとはインターネット上のWebページを閲覧するためのソフトウェアです。Webページはプログラミングによって作成されており、これをユーザーが見ても理解することはできません。それをユーザーが見ても分かるようにプログラムからWebページに変換することができます。

　URLとは見たいWebページがインターネット上のどこにあるのかを表しています。我々の住所が地球から特定できるようなものです。また、URLにはWebページに条件や情報を与えることができるクエリパラメータを追加することができます。

Webサーバーとはインターネット上でWebページを管理している場所で、あるWebページが見たいという要求があるとURLを確認してユーザーに返答することができます。

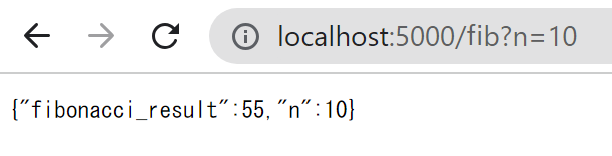
HTTPとはWeb上で情報をやり取りするためのプロトコルです。プロトコルとは通信間の約束事であり、電話をするときに「もしもし」から始めてお互いに通信が行えているか、適切な通信相手であるかを確認することと似ています。

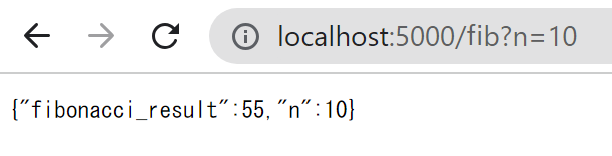
HTMLとはWebページを作成するための文章です。ユーザーが目にするWebページはデザインされたものですが、実際はマークアップ言語という文章で構成されています。この文章にはタグが組み込まれており、コンテンツの始まりと終わりの範囲を表しています。タグ内にはテキスト、リンク、画像、表など様々な属性と呼ばれる要素が含まれています。(736字)

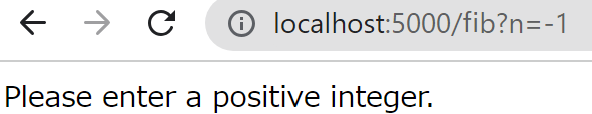
問題1.で作成したソースコード一式

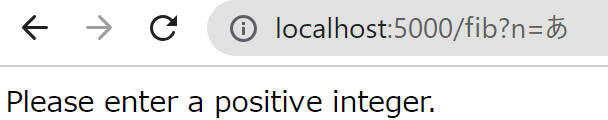
|  |
| --- |
| *#作成したソースコード*  *#サーバーのエンドポイントURL　http://localhost:5000/fib?n=<正の整数>*  *#フレームワークはFlaskを使用する。*  **from** **flask** **import** Flask, request, jsonify  app = Flask(\_\_name\_\_)  *#フィボナッチ数を計算する関数を定義*  **def** fibonacci(n):  **if** n == 1:  **return** 1 *#nの値が1のときは1を返す*  **elif** n == 2:  **return** 1 *#nの値が2のときは1を返す*  **else**:  *#nの値が3以上のときはフォボナッチ数を返す*  **return**(fibonacci(n-1) + fibonacci(n-2))  *#エンドポイント'/fib'におけるデコレータ*  @app.route('/fib')  **def** get\_fibonacci():  n = request.args.get('n') *# リクエストパラメータ 'n' の値を取得*  **if** n **is** **None**:  *#クエリパラメータ'n'が指定されていないときにエラーを表示*  **return** 'Request parameter **\'**n**\'** is not specified.'  **if** **not** n.isdigit():  *#nの値が0以上の整数ではないときにエラーを表示*  *#nに数値以外が入力されたときもここでエラーを表示*  **return** 'Please enter a positive integer.'  **if** int(n) < 1:  *#nの値が1未満のときにエラーを表示*  **return** 'Please enter a positive integer.'  *#nに1以上の正の整数が入力されたときにフィボナッチ数を返す*  result = fibonacci(int(n))  *#フィボナッチ数が返されたときに、フィボナッチ数と入力されたnの値を表示*  print('fibonacci\_result:',result , ', ' , 'n:', int(n))  *#JSON形式でフィボナッチ数と入力されたnの値を表示*  **return** jsonify({  'n': int(n),  'fibonacci\_result': result  })  *#スクリプトが直接実行された場合にFlaskが開発用Webサーバーを起動し、*  *#アプリケーションを実行する。*  **if** \_\_name\_\_ == '\_\_main\_\_':  app.run() |

実行結果

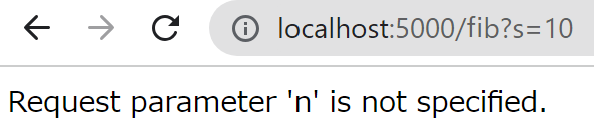
http://localhost:5000/fib?n=10　を入力した結果



http://localhost:5000/fib?n=-1　を入力した結果

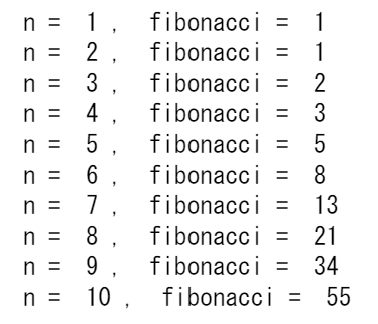
http://localhost:5000/fib?n=あ　を入力した結果

|  |
| --- |
| *#ユニットテスト1*  *#フィボナッチ数を正しく返しているかのテスト*  **def** fibonacci(n):  **if** n == 1:  **return** 1  **elif** n == 2:  **return** 1  **else**:  **return**(fibonacci(n-1) + fibonacci(n-2))  *#nの値が1から10までのフィボナッチ数を確認*  **for** i **in** range(1,11):  print('n = ',i, ', ', 'fibonacci = ', fibonacci(i)) |

http://localhost:5000/fib?s=10　を入力した結果

ユニットテスト

実行結果



|  |
| --- |
| *#ユニットテスト2*  *#リクエストパラメータ'n'の値が文字列であることを確認*  *#サーバーのエンドポイントURL　http://localhost:5000/fib?n=<正の整数>*  **from** **flask** **import** Flask, request, jsonify  app = Flask(\_\_name\_\_)  @app.route('/fib')  **def** get\_fibonacci():  n = request.args.get('n')  **if** isinstance(n, str):  *#'n'が文字列ならば表示*  print('**\'**n**\'**は文字列です。', n)  **else**:  *#'n'が文字列でなければ表示*  print('**\'**n**\'**は文字列ではありません。', n)  **if** \_\_name\_\_ == '\_\_main\_\_':  app.run() |

実行結果

http://localhost:5000/fib?n=10 を入力した結果