
教育レポート：遠隔授業でのプログラミング導入教育

総合情報学部教授 萩野正樹

はじめに

2020年度はコロナウイルス感染対策による影響のため、春学期は原則として遠隔授業、秋学期は原則対面による授業となるも受講生が大人数となる授業は遠隔による授業が義務づけられる状態となった。本稿では、筆者が担当しているプログラミングの導入に相当する授業において、遠隔授業の事例を紹介し、遠隔授業でのプログラミング教育の取り組みの現状と問題点、今後の展望について紹介していきたい。

「コンピュータの言語」

筆者が所属する総合情報学部では、情報処理の知識を基礎として、コンピューティング、社会情報システム、メディア情報について、分野横断的・融合的な分野の研究及び教育を行うことを目的としている。その基礎となる科目として1、2年生時には「情報処理」「コンピュータの言語」「コンピュータの物理」「コンピュータネットワークの基礎」「情報システムの基礎」「情報社会論」「情報と倫理」といった、全員が必修の基幹科目の授業が開講されている。これらの基幹科目を履修することで、学生は概ね情報処理分野の基礎を一通り学ぶことができるようになっている。

筆者は、この基幹科目群の中で「コンピュータの言語」という授業を担当している。本授業は1年生秋学期に開講されており、2020年度は金曜日に1、2限目に開講され、それぞれ300人ずつ、合わせて600人の受講者（総合情報学部は1学年約550人であり、毎年50人程度の再履修者がいるため）がいる。このため2020年度は遠隔授業での実施となった。

授業概要をシラバスから引用すると

コンピュータにおける情報表現の基礎ならびにコンピュータを構成するソフトウェアとその利用形態についての知識を習得する。特に、コンピュータによる情報処理の手順で

あるアルゴリズムの基礎、そしてアルゴリズムを図的に表現するフローチャートを用いてデータの探索や並び替えを行う処理手順を説明する。最後に、アルゴリズムをコンピュータ上で実行するソフトウェアである言語プロセッサについても概説する。また各トピックにおいてプログラム言語 Python または JavaScript を使ってプログラムの基本を学ぶ。

と、まさに総合情報学部におけるプログラミングの導入としての役割を担っている。授業の授業計画を図1に示す。

- (1) オリエンテーション
- (2) コンピュータと情報表現
- (3) 2進数と基数変換
- (4) 文字コードの体系
- (5) RGBによる色の表現
- (6) アルゴリズムとフローチャート
- (7) 順次構造と選択構造
- (8) 繰り返し構造
- (9) 逐次探索と二分探索
- (10) 整列アルゴリズム
- (11) 基数変換のアルゴリズム
- (12) 再帰的アルゴリズム
- (13) プログラムの実行方式
- (14) 言語プロセッサ
- (15) 総括

図1 「コンピュータの言語」の授業計画（関西大学シラバスより）

この授業は、基本的には座学の講義形式で行っているが、実際にプログラミングを通してフローチャートやアルゴリズムを理解すること、そして、近年データサイエンスやAI分野で非常によく使われるようになったプログラミング言語 Python の導入にも力をおいている。以下では、2020年度に遠隔授業で利用したプログラミング学習のための支援ツールについて紹介していきたい。

遠隔授業の支援ツール

全体構成

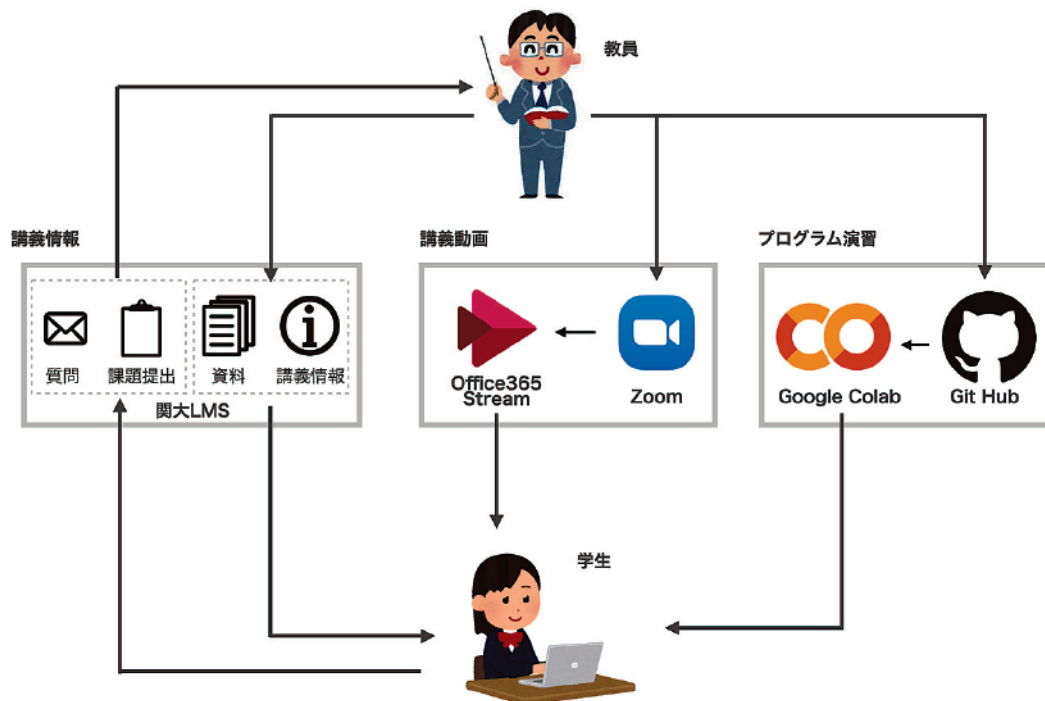


図2 教員と学生との間の情報の流れ

図2に教員と学生との間の情報の流れを示す。授業の情報としては、講義についてのお知らせや課題についての情報、資料などの「講義情報」、講義内容の解説の動画などの「講義動画」、そして課題を行うための「プログラム演習」からなる。

講義情報(教材配布と連絡)

講義情報の連絡に関しては関大LMSを利用した。関大LMS（Learning Management System）は2017年より関西大学に導入されたオンライン学習管理システムである。教材配布や課題レポートの提示・提出、出席確認等、非常に多くの授業支援機能を持っている。実は、授業「コンピュータの言語」では、2019年度までは教員が用意したサーバで講義情報の提示・教材配布・課題の提出などを行っていたが、2020年度は多くの授業が遠隔授業の形態で行われ関大LMSが活用されていたため、学生が多種のツールを利用すると授業の情報源について混乱が生じる恐れがあり、本授業においても関大LMSを利用することにした¹⁾。

1) 2020年度は、関西大学では授業の情報源として、インフォメーションシステム、関大LMS、関大webメールがあり、授業によって異なる手段で学生への授業連絡が行われたため一部で混乱が生じていた。2021年度からインフォメーションシステムが更新されることになり、講義情報の連絡手段について共通化が図られる予定である。

本授業では、関大 LMS の機能の中から、おもに「教材作成」と「お知らせ」の2つの機能を使い、「教材作成」機能によって、動画配信情報の提示、教材配布、課題提出の管理を行い、「お知らせ」機能によって学生からの質問対応を行なった。

また、教科書は指定せず、授業で利用する Microsoft PowerPoint のプレゼンテーションファイルを PDF にしたものを関大 LMS で配布した。

動画配信

関西大学の2020年度の秋学期の授業は、受講者人数が200人以下の場合には原則「対面授業」、200人以上の場合には「オンデマンド授業」と決められていた。このため300人程度の受講生がいる「コンピュータの言語」ではオンデマンド配信による授業を行うことになった。オンデマンド授業は、通常あらかじめ録画した授業の動画を受講生が好きな時間に見ることができるという授業形態である。しかし、できるだけ通常の授業と同様の形態に近づけられるように、本授業では時間割に応じた授業時間に Zoom によってリアルタイムに配信を行い、それを録画した動画をオンデマンドに配信を行った²⁾。オンデマンド配信にはマイクロソフトの Office365 の提供するサービスの一つである Microsoft Stream を利用した。Microsoft Stream は「組織内のユーザーがビデオを安全にアップロード、視聴、共有できるエンタープライズ ビデオ サービス」である³⁾。

Zoom では動画は mp4 ファイルとして録画され、Zoom のクラウドサーバまたはクライアント側のコンピュータに保存される。授業時間は通常の90分とほぼ同じ時間行ったが、動画ファイルの容量250-300MB 程度であった。Microsoft Stream の機能によって、閲覧者数がわかるが、興味深いことに、閲覧者数の増減は例年の対面授業と同様に、動画の閲覧者は授業回数によって徐々に減少し、定期試験が近づくとつれて増加していく傾向を示した。

演習環境

本授業では、実際にプログラム言語を使いながらやプログラムやアルゴリズムを理解してもらうために、プログラミング課題を受講生に課している。そのためのプログラム言語として、本授業では Python を採用している。

2) Zoom は Zoom ビデオコミュニケーションズが提供するクラウド型 Web 会議サービスである。2020年度に関西大学は学生・教員についての契約を結び、遠隔授業や会議を行う標準アプリケーションとして Zoom が使われた。

3) Microsoft Stream では組織内のユーザーを検索することが可能な機能があるが、この機能が学内の個人情報取扱に関する規定に反することから、Microsoft Stream は2021年1月に学内の提供サービスから削除されることになった。

Python

Python は Guido van Rossum 氏によって1990年代初めに開発されたプログラム言語である。可読性を重視した設計となっており、その高い拡張性から非常に広い応用分野を持っており、システム管理・ツール・アプリケーション開発・科学技術計算・Web システムなど様々な分野で利用されている。特に、最近のデータサイエンス・人工知能分野では Python が使われることが多く、高い人気を得ている (Guo, 2021)。

本授業では2014年から「コンピュータの言語」で教育用のプログラム言語として Python を導入している。ちなみに、2014年の時点で、アメリカの大学のプログラミング教育では、Python は最もよく採用されているプログラミング言語であったが、同じころ、日本ではまだ知名度が少なく、書店でも Python に関連した書籍はほとんどなかった。2021年現在では、人工知能ブームの高まりとともに Python に関連した本は非常に多くの書籍が出版されている。また、総合情報学部においても Python を扱う授業は、経済、経営、ロボット、データサイエンスを扱う授業や実習で扱われるようになってきており、2021年度からは、Python でプログラミング実習も追加される予定となっている。

Jupyter Notebook / Google Colab

プログラムの課題を与える場合に、最初に問題になるのは実行環境である。通常であれば、大学で行う場合には実習用の教室があり、必要なソフトウェアやライブラリのインストールを事前に行って環境を整えることができるが、2020年度は遠隔授業であるため、課題においても遠隔で行うことを想定しなければならない。学生の環境は OS やそのバージョンが多様であり、スマートフォンやタブレットしか持たない学生もいるかもしれない。そこで、本授業では Web ブラウザで Python が実行できる環境である Google Colaboratory を授業用の課題を行うプラットフォームとして選択した。

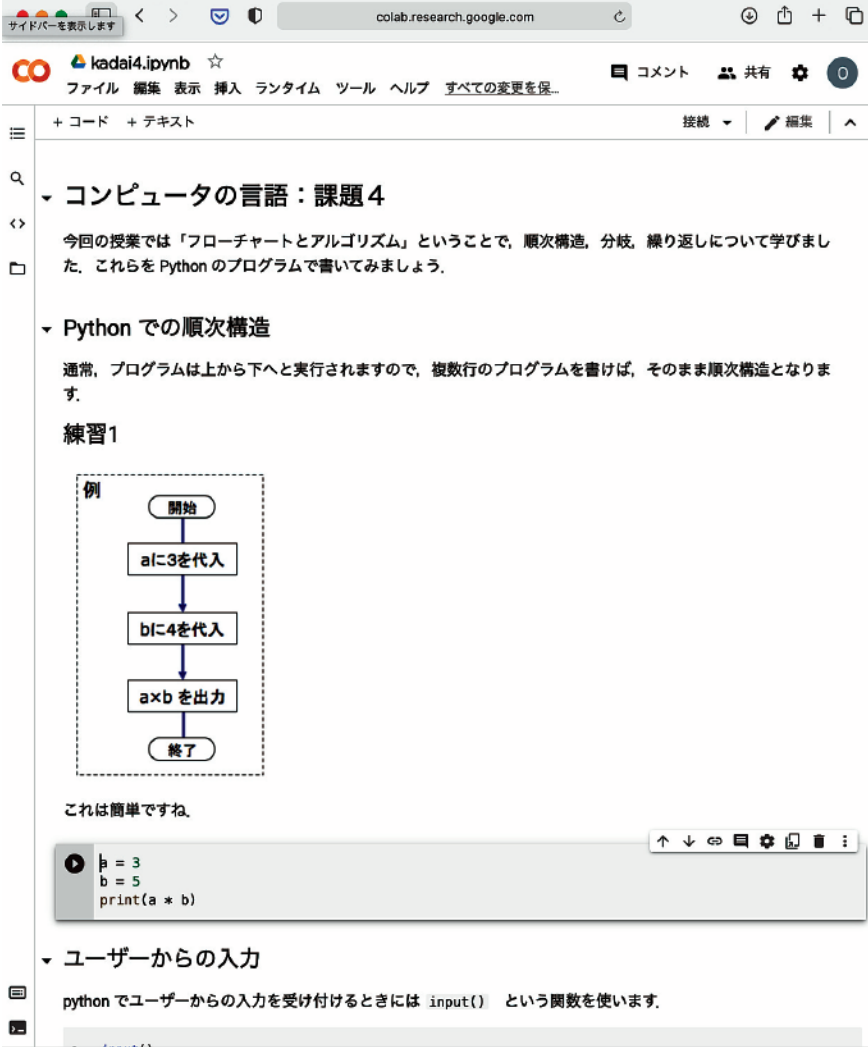
Google Colaboratory (以下では Google Colab と呼ぶ) とはブラウザ上で Python のプログラムを記述、実行できる Google が提供するクラウドサービスである。Google Colab では、ブラウザ環境さえあればオペレーティングシステムによらずに実行環境を利用することができるため、学生のような様々なコンピューター環境を考慮しなければならない遠隔授業では理想的な実行環境であるといえる。

Google Colab は、もともと「数十のプログラミング言語にまたがるインタラクティブ・コンピューティングのためのオープンソース・ソフトウェア、オープンスタンダード、サービスを開発する」ことを目的としたプロジェクトおよびコミュニティである Project Jupyter から生まれた Jupyter Notebook と呼ばれるインタフェースを Google が自社のクラウドで動

作するようにして公開したものである。ユーザーは Google アカウントを取得していれば無料で利用することができる。

Google Colab のもう一つの特徴は Jupyter Notebook と呼ばれるプログラムと説明記述を混在させることができるインターフェースである。これによって学生は画面を切り替えることなく、説明を読みながらプログラムを記述し、その結果をすぐに確認することが可能である。図 3 に教材の例を示す。

この Jupyter Notebook のファイルは、ipynb ファイルと呼ばれるフォーマットで記述されている。作成した課題ファイルを、あらかじめ Github 上に置いておくことで、クリックするだけで指定したノートブックファイルを Google Colab で開くことのできる URL リンクを作成可能である (Google Colaboratory, 2021)。学生は関大 LMS 上に掲示された課題のための URL リンクをクリックすることで、すぐに課題に取り掛かることができる。



The screenshot shows a Google Colab notebook titled 'kadai4.ipynb'. The main content is a lesson titled 'コンピュータの言語：課題4'. It includes a section 'Python での順次構造' with an example flowchart and a code cell. The flowchart shows a sequence: '開始' (Start) -> 'aに3を代入' (Assign 3 to a) -> 'bに4を代入' (Assign 4 to b) -> 'a×b を出力' (Output a*b) -> '終了' (End). The code cell contains:

```
a = 3
b = 5
print(a * b)
```

図 3 JupyterNotebook を使った教材の例

課題の提出と管理

課題の提出については、課題を実行した ipynb ファイルを自身のコンピュータにダウンロードし、そのファイルを関大 LMS の提出ページにアップロードしてもらうことによって課題の管理を行なった。

課題の実行ではほとんど問題はなかったが、課題の提出では、いくつか問題があった。

まず、「iPad で提出できない」という問い合わせがあった。これは iPad で行なった場合、Google Colab から ipynb ファイルをダウンロードすることができない仕様になっていたことが原因であった。これは一旦ファイルを Google Drive に保存してもらい、それを iPad にダウンロードしてから関大 LMS にアップロードするように指示をした。

また、「ipynb ファイルをダウンロードするときにエラーメッセージが表示される」というものもいくつかあった。これは ipynb ファイルはテキストファイルになっているが、一般的なアプリケーションのフォーマットではないため、利用するブラウザあるいはセキュリティソフトの設定によってダウンロードが実行できないことが原因であった。これについては、まず ipynb ファイルを「メモ帳」などのアプリケーションに紐付けを行ってダウンロードを可能にするように指示をした。

さらに、「ブラウザで実行したものが反映されているか不安である」という問い合わせも多くあった。これは ipynb ファイルは一般的なアプリケーションで表示することができないことが原因である。これについては、ipynb ファイルをドラッグ&ドロップで表示できるページ (kokes, 2021) で確認する、あるいは Google Colab のページへアップロードすることによって確認するように指示をした。これらのトラブルについては関大 LMS に「よくある質問」のページを設け、解決策を掲示した。しかし、このページを見る人は少なく、同じような質問を関大 LMS のメッセージ機能で送ってくる学生が多くいた。学生にとって最も手軽に尋ねることができる手段は関大 LMS のメッセージ機能となっている印象であった。

課題の採点

提出された ipynb の課題ファイルは、教員は関大 LMS から一括でダウンロードすることができる。これらは受講生約600名分あるため、自動採点プログラムを導入することによって採点を行なった。

前述したように ipynb ファイルはテキストファイルであり、その中は説明記述のマークダウン、プログラムの入力、出力などの種類ごとに JSON フォーマットで記述されている。こ

のため、Python プログラムによって簡単に読み込みを行うことができる。Jupyter Notebook のフォーマットと、それを読み込み、書き込みをするためのライブラリ nbformat の詳細についてはライブラリのドキュメントページ (The Jupyter Notebook Format, 2021) に詳しく説明されている。

上記で説明されているライブラリ nbformat を利用することにより、ファイルに記述されている課題のプログラムの実行結果を読み込み、文字列情報として確認することができる。その文字列の比較を行うプログラムを書くことによって自動採点を行うことができる。

採点結果は、関大 LMS でダウンロード時に生成される CSV 形式のファイルに合わせて、点数と講評を追加記入したファイルを生成する。このファイルを関大 LMS にアップロードすることによって関大 LMS の各課題の学生の成績に反映され、学生自身で結果を確認することができる。例年は提出のみで、結果については学生に十分にフィードバックできていなかったが、2020年度は自動採点プログラムを導入することにより、今年度は課題の点数が低いものについては再提出などを課すことができた。

最終試験の提出システム

例年、コンピュータの言語では定期試験によって成績評価を行なっているが、大人数の受講者に対して定期試験を行うことは感染のリスクを伴うため、2020年度は初めて遠隔試験を導入することにした。遠隔試験での不正をできるだけ少なくするため、一人一人に与える問題を異なったものとし、ダウンロードしてから一定時間以内に提出するというシステムを構築した。

Jupyter Notebook は基本的にはテキストファイルであることから、プログラムを使って簡単に生成することができる。そこで、出題分野を10分野として、それぞれの分野ごとに5題程度の問題候補を作成しておき、そこからランダムに3つの問題を選択して問題ファイルを作成するプログラムを作成した。生成されたファイルはあらかじめダウンロード用のサイトにおいておき、それぞれのファイルに対して乱数を使ったダウンロード用の URL を設定した。この URL は学生個人ごとに異なるため、学生の名簿ファイルから抽出した利用者 ID の情報を使って関大メールのメールアドレスに送付した。関大メールは Exchange Online の制限により「1分あたり30個のメッセージ」までしか送信できない仕様になっているため (Microsoft, 2021)、2秒ごとに送信した。600名の送信に20分程度を要した。

学生は、各自都合のよい時間にメール内に示されたダウンロード用の URL をクリックして試験ファイルをダウンロードし、そのファイルを Google Colab のページにアップロード

して、試験問題に取り組む（問題の指示に応じてプログラムを記入、実行する）。そして、試験問題のファイルはダウンロードしてから 1 時間以内に関大 LMS に提出するように指示をした。

おわりに

本稿では、2020年度に実施した遠隔授業「コンピュータの言語」の実施方法について概説した。Google Colab という Google から提供されている Python の実行環境を利用することで、学生のコンピュータ環境によらず、ブラウザさえあればプログラミング教育を行うことができることを示した。

最後に遠隔授業を実施して感じた幾つかの点について述べておきたい。

まずは、遠隔授業での教え合いが減ったことである。例年の対面授業では、学生同士が自然発生的に生じたグループ内や知り合い同士で互いに教え合うということが行われていた。しかし、4月の授業開始時から遠隔授業を受けざるを得なかった1年生では、このようなグループや知り合いは形成しづらかったようである。取り残された学生は、わからない場合の対処が難しく、授業についていけない学生も少なからずいたようである。そのような人間関係の形成に関するところまでは授業の範囲外と考えていたが、このような状況下においては、人間関係を形成していく仕掛けを遠隔教育で用意することも考える必要があるかもしれない。

次に、遠隔授業の利点についてである。特に総合情報学部は立地の問題もあり、アンケートにおいても通学における混雑は常に学生の不満にあげられるところである。また、下宿をせずに長い通学時間をかけて自宅から通学する学生も多く、本授業のように1限に開講されている授業は出席者が減りがちである。このような学生には通学の問題が解消されてよかったと感じる人も多かったようである。また、動画で配信される授業は、自分のペースで動画を見ることができ、また何度も見返すことができるため、理解しやすかったというアンケート結果もあった。大人数が大教室で受講する授業よりも教員を身近に感じやすいのか、関大 LMS 経由での質問も例年に比べて多くあった（その分教員にとっては対応がたいへんであるのだが）。このような利点もあるため、コロナ後の教育においても、遠隔での教育環境をどのように活用していけるのか、についての検討は必要であろう。

最後に、Google Colab のような外部システムを利用することについてである。これについては、無料とはいえ、学外の一企業によって提供されているサービスを利用することについての懸念があるかもしれない（このサービスの利用のためには、学生一人一人が Google アカ

ウントに登録する必要がある)。また、現在無料で提供されているサービスが有料化された場合にどのように対処するのか、といった問題もある。これについては、JupyterHub というオンプレミスのサーバーで同様の Python ノートブックの実行環境を提供することができるオープンソースのプロジェクト (Jupyterhub, 2021) があり、実際に、既に多くの大学で JupyterHub を使ったプログラミング教育も行われている。このシステムを大学内のコンピュータで構築することにより、同様のサービスを提供可能であるだろう。

一方で、Google Colab を積極的に教育で利用するという事も考えられる。最近データサイエンスや AI に関する教育の必要性が重要視されているが、Google Colab は時間制限があるとはいえ、計算能力の高い GPU を無料で利用することができる (最近、時間制限を緩めた有料サービスが日本でも開始された)。多くの受講生を一度に教育する大学としては、Web ブラウザ経由で手軽に高度な計算機環境が利用可能なサービスは今後ますます重要になると考えられる。

今回は初めての遠隔授業ということで、画面の共有設定を誤り、スライドが映し出されていないことに気づかずに遠隔授業を行うなどの基本的な失敗も経験した。また遠隔授業のためのシステムや自動採点システムなどの構築に時間がかかり、作成した演習問題などのチェックが疎かになり、誤りについて学生からの指摘されることも多くあった。遠隔授業時の不具合や演習問題の誤りを指摘してくれた受講生には感謝の意を表したい。また、これらの至らない点については大いに反省し、その経験を次年度の講義に活かしていきたい。

引用文献

Google Colaboratory. (2021年4月19日). Using Google Colab with GitHub. 参照先: <https://github.com/googlecolab/colabtools/blob/master/notebooks/colab-github-demo.ipynb>

GuoPhilip. (2021年4月19日). Python Is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. 参照先: BLOG@ACM: <https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-u-s-universities/fulltext>

Jupyterhub. (2021年4月19日). 参照先: JupyterHub: <https://jupyterhub.readthedocs.io/en/stable/>

kokes. (2021年4月19日). 参照先: nbviewer.js: <https://kokes.github.io/nbviewer.js/viewer.html>

Microsoft. (2021年4月19日). Exchange Online の制限. 参照先: Microsoft ドキュメント: <https://docs.microsoft.com/ja-jp/office365/servicedescriptions/exchange-online-service-description/exchange-online-limits>

The Jupyter Notebook Format. (2021年4月19日). 参照先: <https://nbformat.readthedocs.io/en/latest/index.html>