



メディア制作演習 (データ分析)

担当：大用庫智

2018年春学期開始 2025年度改訂版

関西学院大学 総合政策学部

- ・ イントロダクション
- ・ プログラミングとは？
- ・ Pythonとは
- ・ Visual Studio Code のインストールと使い方の実践

- ・ 本講義の目的
 - ・ 履修者がデータ分析のスキルを身につけること
 - ・ データ分析のスキルを身につけながら、
 - ・ 履修者はVisual Studio Code の基本操作を身につけ、
 - ・ プログラミング言語Pythonの基本となる用語と概念を理解

- ・ データ分析のスキル獲得
 - ・ データ分析のスキルを獲得するために、
 - ・ VSCode の基本操作を習得する
 - ・ Python言語の基本的な用語と概念を習得する
- ・ データ分析のスキル獲得の想定レベル
 - ・ プログラミングを使い、ある程度ソフトウェア的に下記の分析を行えるようにする。
 - ・ 統計的検定： Basic statistical tests
 - ・ 回帰分析: Regression analysis
 - ・ 重回帰分析: Multiple Regression analysis
 - ・ ロジスティック回帰 Multiple Regression analysis
 - ・ 階層的クラスタリング手法: Ward's method
 - ・ 決定木分析: Decision Tree Analysis
- ・ ただし、高度な数学的な内容はブラックボックスとして進める。

本講義では扱わないが、これらを理解するためには高校数学ⅠからⅢ、統計学が不可欠
参考文献：松坂 和夫 の 数学読本1-6, 参考文献：久保川 達也の現代数理統計学の基礎

- ・ 大まかな授業の流れ
 - ・ 講義形式：30分から60分
 - ・ 演習形式：残り時間
 - ・ 課題を配布：授業時間外で実施
- ・ 評価の割合
 - ・ 平常レポート100%:
 - ・ 履修者が習得した知識の内容と活用を問うため、小課題とレポートを課し、その結果に基づいて評価する。
 - ・ シラバス転用：
 - ・ **注意点1**：本講義は演習形式であり、毎回前回の内容を積み上げる形式である。各講義の内容を理解したことを確かめるために、毎週演習の課題の提出を求める。演習時間内に課題提出が終わらない場合は、講義時間外での時間が必要である。
 - ・ **注意点2**：寝ていてなどの理由で作業をしないなどの場合は欠席とカウントする場合がある。

- ・ 担当教員
 - ・ メディア情報学科 大用庫智 (おおよう くらとも)
- ・ 簡単な質問の場合
 - ・ Slackのチャンネルで質問
- ・ 個人情報を含む限定的な連絡のみ
 - ・ e-mail: o.kura@kwansei.ac.jp は
- ・ 難しい質問の場合 (対面可能な場合のみ)
 - ・ 直接居室に訪問 (メールで連絡してくれるとスムーズ)
 - ・ パソコンやプログラムなどの資料を持ち込む
 - ・ オフィスアワー 水曜3時間目、部屋番号 3-311

履修者へのお願い2



- ・ 履修者同士で協力してください。
 - ・ 隣の人がわかっていなかったら、教えてあげてください。
- ・ 演習中わからないことは積極的に質問してください。
 - ・ **手をあげるか、呼び止めてください。**
- ・ 本演習を履修する学生は、初めて高度なコンピュータ操作を行うことを想定
 - ・ **（初めてなので知らなくて当たり前）質問は恥ずかしがらずに**
 - ・ **「こんなくだらないことを聞いてもいいのか？」と思うかもしれませんが、思ったとしても、聞いてください。**
- ・ **質問の言語化が難しい場合**
 - ・ **「今、〇〇資料を見ていて、ここで詰まりましたなど」と教えてください。**

受講の準備

- ・ パソコンの持参 (Bring Your Own Device: BYOD)
 - ・ **原則、自分のPCを持参**して授業に参加してください。
 - ・ 原則対面授業から、**完全オンライン授業に変更になった場合に、**自分のPCで作業できれば、**混乱なく学習を継続可能**
 - ・ 本講義の中で自分のパソコンを自分で管理できるようになります！
- ・ パソコンのスペック
 - ・ パソコンは、学内と家庭でインターネットに接続可能な **Windows 11** を搭載したパソコン（または購入3年以内の**Mac book**）
 - ・ **説明は基本Windows ベースです。**
- ・ パソコンが用意できない学生
 - ・ 対面授業の間は、**大学のPCを利用可能**。ただし、PC教室に余裕がなかったり、大学内で作業時間が取れないことを想定して、自宅PCでの予習、復習は必要

回数	内容
第 1 回	ガイダンスとPythonと Visual Studio Code の導入
第 2 回	プログラミング基礎：変数
第 3 回	プログラミング基礎：データフレームと基本統計量
第 4 回	データ分析: 統計的検定
第 5 回	データ分析: 統計的検定
第 6 回	データ分析: 回帰分析
第 7 回	データ分析: 重回帰分析
第 8 回	データ分析: ロジスティック回帰
第 9 回	階層的クラスタリング手法: Ward's method
第 1 0 回	階層的クラスタリング手法: Ward's method
第 1 1 回	画像の取り扱い
第 1 2 回	画像の取り扱い
第 1 3 回	画像の取り扱い
第 1 4 回	決定木分析

回数	内容
第1回	ガイダンスとPythonと Visual Studio Code の導入
第2回	プログラミング基礎：変数
第3回	プログラミング基礎：データフレームと基本統計量
第4回	データ分析: 統計的検定
第5回	データ分析: 統計的検定
第6回	<div>6回目以降は要相談</div> <div>Pythonでは画像処理や自然言語処理など多種多様のデータ分析ができるため、新しい機材等を用いた発展的なデータの扱い方についても検討</div>
第7回	
第8回	
第9回	
第10回	
第11回	
第12回	
第13回	
第14回	

プログラミングってなに？



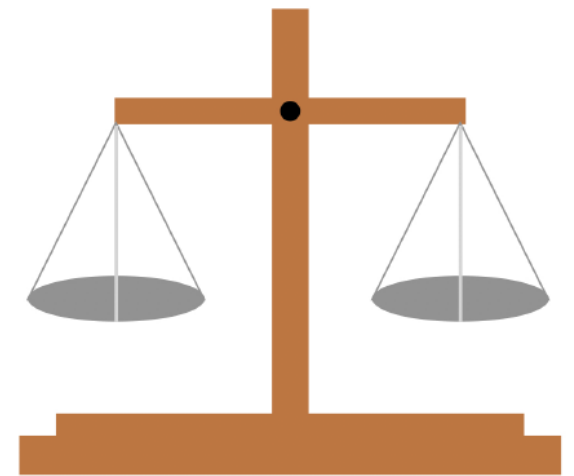
- 私たちはコンピュータに仕事(処理)を指示するために、コンピュータにやらせたい仕事(処理)を専用の命令で順番に記述したもの、即ちプログラムを作成します。このプログラムを作成する作業を、プログラミングといいます。



- コンピュータは人間と異なり、自ら考えることができないため、一つ一つの作業手順を記述する必要があります。この作業手順を記述するために、私たちが無意識のうちに一瞬で判断している様々な処理を、**一つずつ分解した命令**として順番に書き下す必要があります。初学者に一番わかりにくいのが、**この処理の分解**です。

処理の分解

- 簡単な例で考えてみましょう。
 - ここに見た目は全く同じ3枚の金貨と秤があります。
 - 金貨のうち、1枚は他の2枚より重いのですが、見た目からはわかりません。
 - 秤を使って重い1枚を見つけるにはどういう作業手順が必要でしょうか。
- 私たち人間の場合を考えてみましょう。
 - 私たちであれば、金貨を持てば一番重い金貨を直感的に見つけるでしょう。また、私たちは次のような作業手順を意識して考えずとも秤を利用できます。ところが、コンピュータの場合には処理を分解し、次のような作業手順を明確にする必要があります。
 - 手順1: 金貨や秤とは何か。
 - 手順2: 金貨3枚から2枚を適当に選ぶ。
 - 手順3: 選んだ金貨を秤に一枚ずつ乗せる。
 - 手順4: もし秤が釣り合ったら、皿に乗っていない金貨が重い金貨だとわかる。
 - 手順5: 秤が傾けば、下がった方の皿に乗った金貨が重いとわかる。



プログラミングとは？

- ・ プログラミングとは？
 - ・ 私たち人間が理解している作業手順を、一つずつ処理（命令）として書く作業です。一つ一つの命令は単純ですが、それらを組み合わせることで複雑なプログラムを作成することができます。実際にプログラムを書く際には、英文法の命令系のようコンピュータ用の命令を書きます。これまで述べた一つずつの作業をコンピュータに命令として与えるため使う言語がプログラミング言語です。私たちは日本語でコミュニケーションするとき、日本語の文法に従って話します。それと同じようにプログラミング言語にも言語ごとの文法（ルール）があります。

よく使う用語

- 実行
 - コンピュータがプログラムに従い仕事をする事
 - コンピュータは、プログラムの命令のとおりになし、動きません
- 実行結果
 - コンピュータがプログラムに従い仕事をした結果
- 出力
 - コンピュータが画面に何らかの文字や図表を表示すること

プログラムの作成と実行

①コンピュータに指示する仕事(処理)を分解して作業手順を明確化

手順1: 金貨とは何かを定義する。
手順2: 金貨3枚から2枚を適当に選ぶ。
手順3: 選んだ金貨を秤に一枚ずつ乗せる。
手順4: もし秤が釣り合ったら、皿に乗っていない金貨が重い金貨だとわかる。
手順5: 秤が傾けば、下がった方の皿にのった金貨が重いとわかる。

Pythonの構文により変換

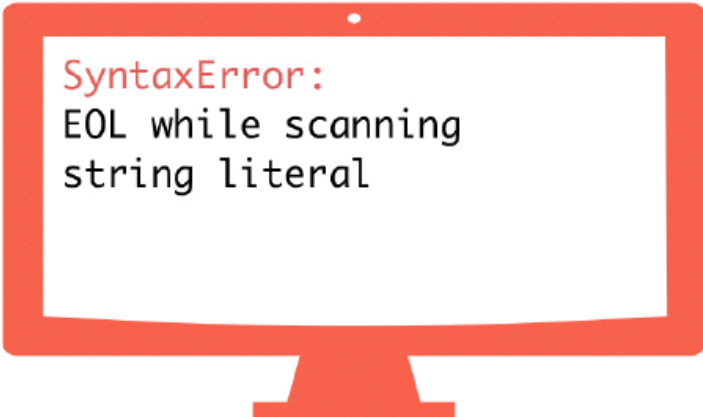
②プログラミング言語を用いてソースコードを記述しプログラムを作成(ソースコード例)

```
Gold_coin = {'kinka1':10,'kinka2':10,'kinka3':11}  
weight1 = Gold_coin[input('一枚目の金貨を選んでください:')]  
weight2 = Gold_coin[input('二枚目の金貨を選んでください:')]  
if weight1 == weight2:  
    print('選ばなかった金貨が重い金貨です')  
elif weight1 >= weight2:  
    print('重い金貨は一枚目です')  
elif weight1 <= weight2:  
    print('重い金貨は二枚目です')
```

③プログラムの実行とデバッグ

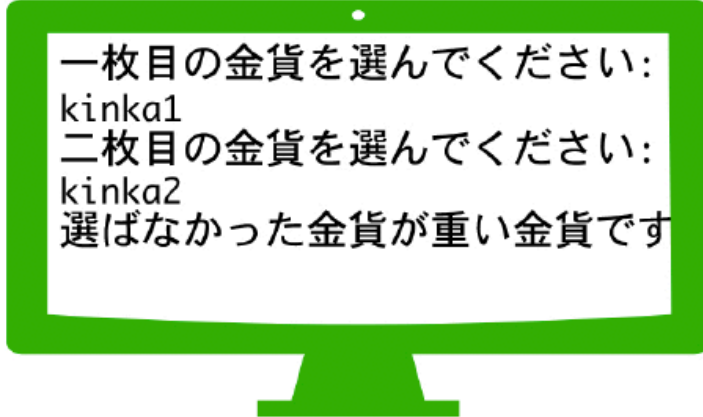
作業手順やソースコードを修正(デバッグ)

入力した命令を即座に実行(インタプリタ型)



```
SyntaxError:  
EOL while scanning  
string literal
```

プログラムに問題がある際の例



```
一枚目の金貨を選んでください:  
kinka1  
二枚目の金貨を選んでください:  
kinka2  
選ばなかった金貨が重い金貨です
```

プログラムを作成完了

プログラミング言語の種類1

表 1.1: 代表的なプログラミング言語

名前	補足説明
C 言語	Java や Python などの多くの プログラミング言語の基礎となる代表的なコンパイラ型の言語 。多くの教育機関で古くから教育用に利用されてきたため学習用の資料が多い。プログラムの処理速度が高速な言語です。C 言語は処理の手続きを一つ一つ記述する手続き型言語のため、Java や Python などよりも学習やプログラムを作成する負担が非常に大きくなります。
C++	C++ はオブジェクト (詳細は下巻で解説) を中心にプログラミングするオブジェクト指向を取り入れて C 言語を拡張したコンパイラ型の言語です。C++ はシーplusplusまたはシーplusplus、シーplusplusと読みます。C 言語のように、オペレーティングシステム (例えば、マウス操作やキーボード入力などの基本的な操作を可能とするソフトウェア) やコンパイラ等の開発、ロボットなどのハードウェアの制御等に使うことが多い言語。上記のような用途に利用するため、C 言語と同様に、メモリやハードウェアにアクセスするプログラムの作成に向いている言語です。
Java	オブジェクト指向型の代表的なコンパイラ型の言語 。多様なアプリケーション開発が可能のため、企業での利用頻度が高く、大規模なシステム開発に向いています。また、多種多様なコンピュータごとにソースコードを書き換える必要がない特徴があります。膨大な参考資料があるため、オブジェクト指向を学びやすい言語です。

プログラミング言語の種類2

名前	補足説明
C#	C # はシーシャープと読む、コンパイラ型の言語です。パソコン用のアプリケーション開発や、Web ページの作成、スマートフォン用のゲーム開発、VR(virtual reality) などを開発する環境 (Unity 等) で幅広く利用できるプログラミング言語。C 言語や C++ と名前が似ていますが、それらの言語と C # の互換性はありません。
Scala	Scala はスカラーと読む、コンパイラ型の言語です。冗長と言われる Java のコードを短縮しつつ、Java の機能をそのまま利用できる言語。Scala は関数型 (関数を主体にプログラミングする言語) とオブジェクト指向の両方の特徴を持つため、Java や関数型言語の利用経験などの知識が必要です。
JavaScript	JavaScript はジャバスクリプトと読む、インタプリタ型の言語です。Web ページを作成する際に Web ページの大枠を作る HTML や Web ページのデザインを記述する CSS と一緒に利用することができる言語。JavaScript は閲覧中の Web ページ上で動的なコンテンツを作成します。例えば、インタラクティブな可視化を可能にする D3.js などが有名です。Python の Django パッケージなどのように他の言語から呼び出して利用します。ただし、JavaScript は Node.js の利用により、Web ページの表側 (クライアント側) と裏側 (サーバー側) を同時に管理もできます。

プログラミング言語の種類3

- PHP PHP はピーエイチピーと読む、インタープリタ型の言語です。主に Web 開発に利用することがある言語。JavaScript とは異なり、Web ページの裏側（サーバー側）を管理（HTML の操作やデータベースと連携）するために PHP を利用します。
- LISP LISP はリスプと読みます。歴史が長く、古くから人工知能研究などで利用されてきた関数型言語。LISP には Scheme などの種類があります。
- VBA VBA は Visual Basic for Applications の略語であり、そのままブイビーエーと読みます。VBA は表計算ソフトと共に利用することができる言語。表計算ソフトのマス目（セル）を操作する命令と共に、コンピュータの操作手順を自動化した命令（マクロ）を作成するために利用します。
- R 言語 **統計処理に必要なパッケージが非常に充実したインタープリタ型の言語。**ソースコードの記述が簡単ですが、処理速度が遅い等の理由から大規模なプログラムの作成には不向きです。
- COBOL COBOL はコボルと読む、コンパイル型の言語です。金融業界等で最も古くからシステム開発で利用され続けている言語です。

プログラミング的思考

- ・ 定義

- ・ 自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力

- ・ 解釈（本演習に隣接する部分）

- ・ コンピュータにやってもらいたい仕事を与えるために、どのような仕事の組み合わせが必要かを考え、一つ一つの仕事をコンピュータにわかりやすい形で説明して与え、……

- ・ だから、どんな能力が必要なのか？

- ・ コンピューターが問題（仕事）を行えるように、人間が問題解決のための正しい視野視点をえることが不可欠