

1. はじめに

PyPLAS (Python Programming Learning Assistant System) は、Python プログラミング学習を支援するためのアプリケーションです。本システムは以下の特徴を持っています：

- ・ **自動採点機能**：選択式やコード記述式の問題に対応
- ・ **対話型実行環境**：コードを直接実行しながら学習可能
- ・ **進捗管理**：問題解答の進捗を確認し、結果を記録する

2. 学習の進め方

PyPLAS の学習の進め方は以下の通りです。

① カテゴリの選択

PyPLAS では学習者に解いてほしい問題を **カテゴリ**という単位でまとめています。トップページから興味のあるカテゴリを選択してください。

② 問題を選ぶ

カテゴリをクリックすると学習可能な問題のリストが表示されます。リストには各問題の進捗状況も表示されます。（図 1）

Problems		
#	title	progress
0	pandas基礎構文1	未学習
1	pandas基礎構文2	学習中
2	機械学習プログラミングの流れ	完了
3	scikit-learn基礎構文	学習中
4	モデルの検証とハイパーパラメータの決定	完了
5	【Extra】手書き文字の認識	未学習

図 1 問題(進捗)一覧

③ 問題を解く

問題の解き方については **4. 問題の解き方**で説明しています。

④ 解答ログを取得

カテゴリ内のすべての問題を解き終えたら、問題リスト画面に戻ります。（図 2）

学生番号と名前を入力して「**Download**」を押すと、解答ログファイルをダウンロードすることができます。

Problems

#	title	progress
0	pandas基礎構文1	完了
1	pandas基礎構文2	完了
2	機械学習プログラミングの流れ	完了
3	scikit-learn基礎構文	完了
4	モデルの検証とハイパーパラメータの決定	完了
5	【Extra】手書き文字の認識	未学習

Log

Student ID

50M*****

① 学生番号を記入

Input your student ID.

Name

山田太郎

② 名前を記入

Input your Name.

Download

③ ボタンを押す

図 2 ログファイルのダウンロード

⑤ 解答ログを提出

ダウンロードしたログファイルを指示された場所に提出してください。

3. UI の使い方

PyPLAS Python Programming Learning Assistant System

Home My Category

Operations

Execute All

Restart Kernel

Interrupt Kernel

Save Answer

User's Guide

Home

How to Use PyPLAS (JP)

How to Use PyPLAS (EN)

The Use of OSS in PyPLAS

Source Code (Github)

pandas基礎構文1

Summary

Data Source

Environment

pandasについて以下のことを学ぶ。

- pandasとは
- データ構造
- 基礎構文

The Source Code

pandasとは？

pandasは、機械学習プログラミングでよく使われるライブラリです。機械学習においてpandasがよく使われる理由は、データの処理や解析を効率的に行えるためです。pandasは、表形式のデータを扱うのに非常に適したライブラリであり、データの読み込みから加工、分析までを簡単かつ直感的に行うことができます。たとえば、データセットに欠損値が含まれている場合、それを見つけて補完したり削除したりする操作を容易に行えます。

ここでは、pandasの特徴的なデータ構造や基礎的な構文について学びます。

```

1 # pandasをインポート
2 import pandas as pd
3

```

pandasは慣習的にpdとしてインポートされます。

Category:

【12月】技術英語

Questions

Q. 1

Complete

Q. 2

Complete

Q. 3

Complete

▼ Inner Link

pandasとは？

1. pandasのデータ構造

1.1. pd.Series

1.2. pd.DataFrame

1.3. SeriesとDataFrameの違い

2. pandas基礎構文

2.1. データの読み込み

2.2. DataFrameの情報を取得

2.3. 列・行の選択

2.3. 欠損値の処理

図 3 問題ページ全体図

① 操作メニュー

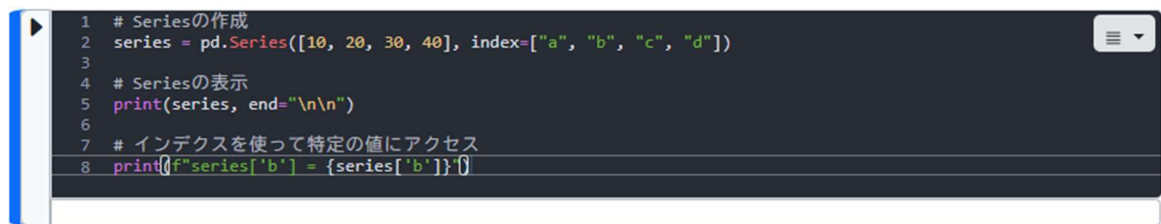
- *Execute All* : すべてのコードブロックを実行
- *Restart Kernel* : 実行環境をリセット
- *Interrupt Kernel* : 実行中の処理を中断
- *Save Answer* : 解答を保存

② Header

- *Summary*: そのページで学ぶ内容の要約
- *Data Source*: ページで使ったデータセットのリンクや参考にした資料
- *Environment*: 実行環境(Python や使用パッケージのバージョン)

③ The Source Code

- **Code ブロック**: プログラムの動作を確認するためのデモコード. (図 4)
 - ✓ 左の▶ボタンでコードを実行可能です



```
1 # Seriesの作成
2 series = pd.Series([10, 20, 30, 40], index=["a", "b", "c", "d"])
3
4 # Seriesの表示
5 print(series, end="\n\n")
6
7 # インデックスを使って特定の値にアクセス
8 print(f"series['b'] = {series['b']}")
```

図 4 Code ブロック

- **Question ブロック**: 学習者の理解を確認するためのテスト.

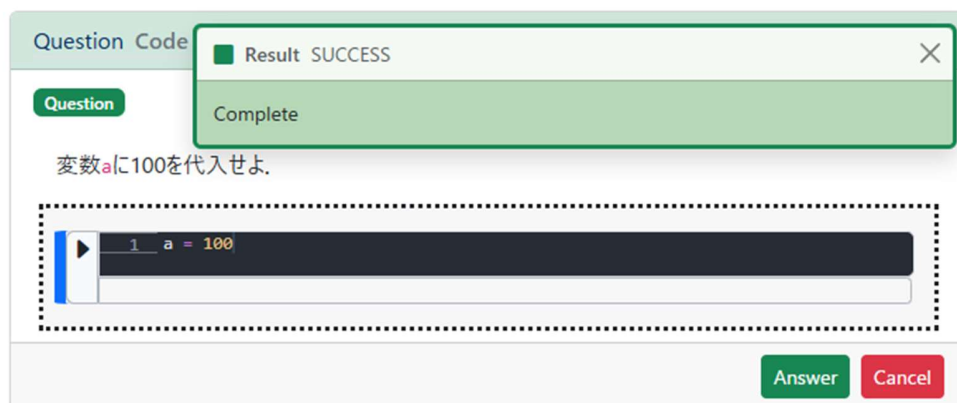


図 5 Code Test 用 Question ブロック(正解時)

- ✓ Question ブロックには 2 つの形式があります。
 - ・ *Word Test*: 選択問題または空白を埋める形式
 - ・ *Code Test*: 条件に従ったコードを記述し、実行する形式
- ✓ 「Answer」ボタンで採点を行います。
- ✓ 「Cancel」ボタンで採点処理を中断できます(Code Test のみ)

4. 問題の解き方

問題への解答は以下の手順で進めてください。

1. Header を確認する

Summary を読み、そのページで学ぶ内容を確認します。

2. The Source Code を実行する

説明文を読みながら Code ブロックを実行することで、プログラムの動作を確認します。

※コードの内容を変更しても、問題には影響しません。

3. Question に取り組む

- Code テストの場合、すでに一部のコードが書かれていることがありますが、その場合は足りない箇所を自分で埋めてください。 **コードを記述し終わったら、一度各コードブロックを実行し、エラーがないことを確認してから Answer ボタンを押すことをおすすめします。**
- 解答が条件に従っていなかった場合は、エラーコードが表示されます。

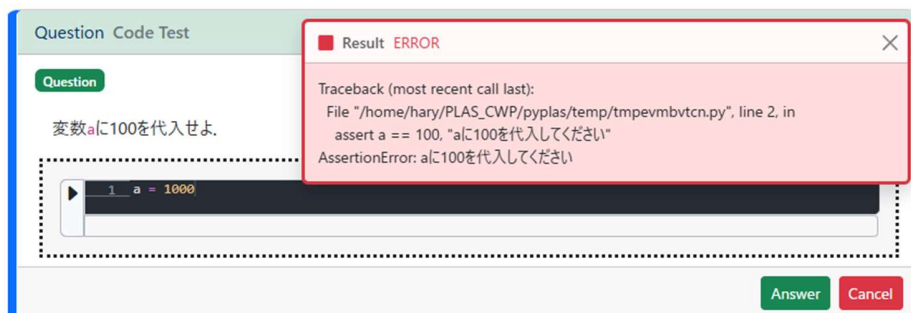


図 6 Question ブロック (不正解時)

注意事項

- 推奨ブラウザ: **Google Chrome, Microsoft Edge**
- 複数タブで同時に問題を解かないでください