

成人のプログラミング学習の ハードルと その越え方

環境の準備、学習範囲の限定、スモールステップ、「失敗」の学習

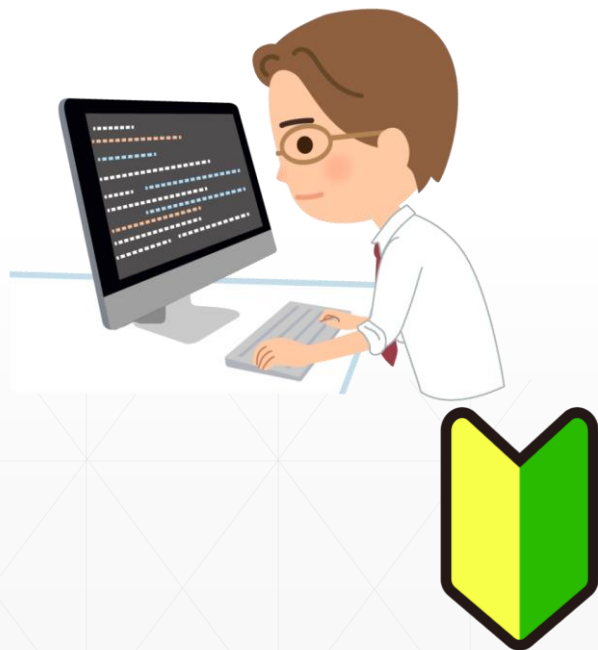
『情報Ⅰ』では
プログラミングを教えることが必須に

『探究』など、他科でも
プログラミングの知識・技能が
役に立つ

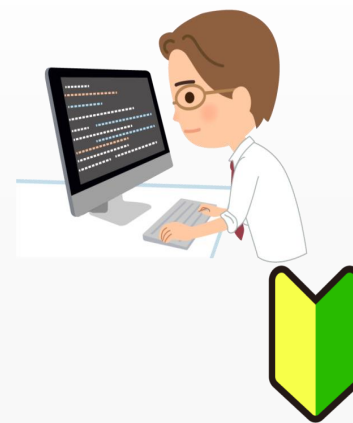


生徒に教える前に…

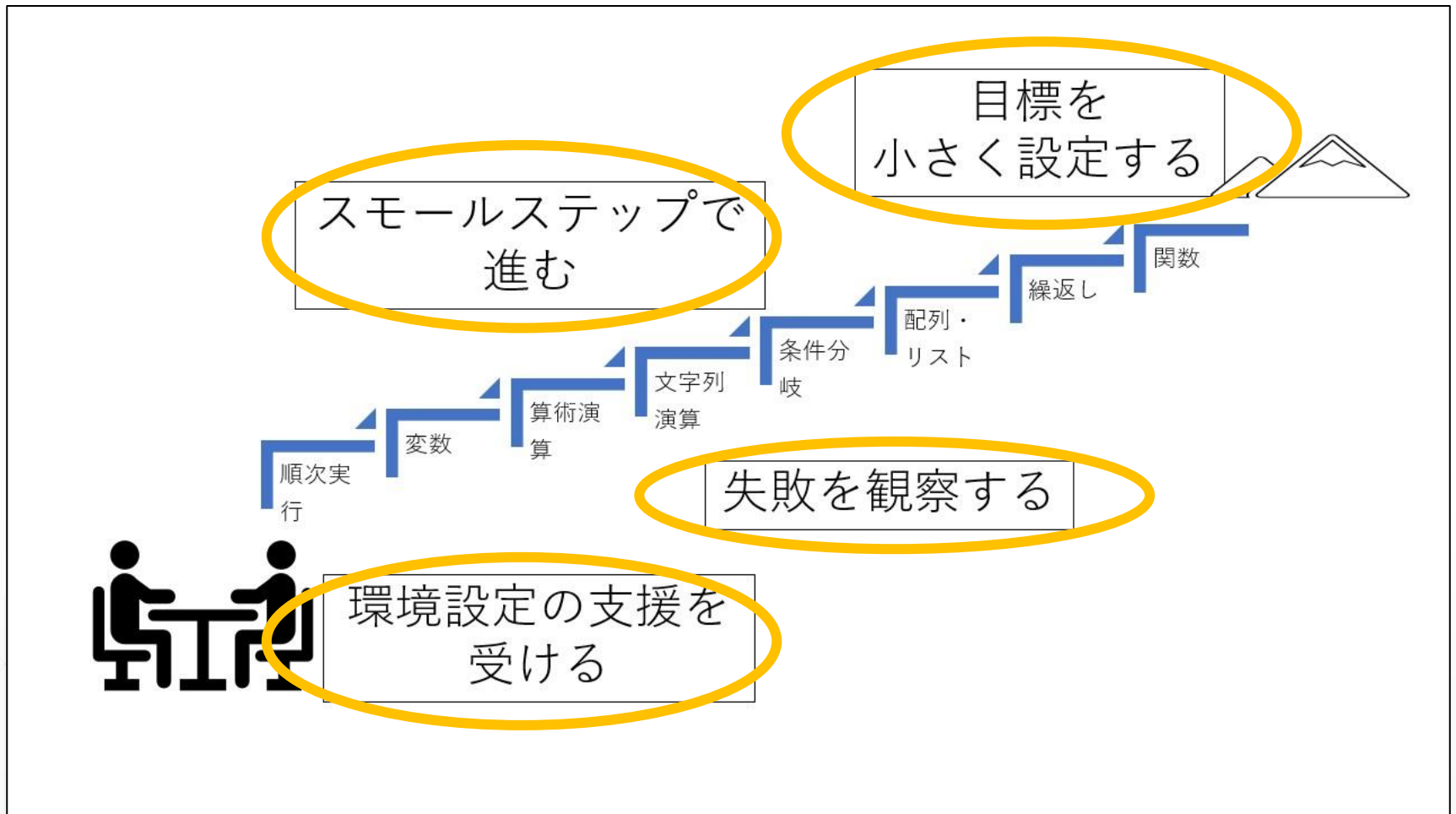
- プログラミング未経験の先生は自分で学ぶ



- プログラミング経験者の先生は、同僚の未経験者を教える・支援する



大人がプログラミングを学ぶときのハードルとその越え方



自己紹介

■ 井澤哲也（いざわ てつや）



■ 所属

- 放送大学大学院 芝崎ゼミ
 - 研究テーマ:「成人のプログラミング教育のためのツール、サポートサイトの開発」
- アシアル株式会社 エデュケーション事業部

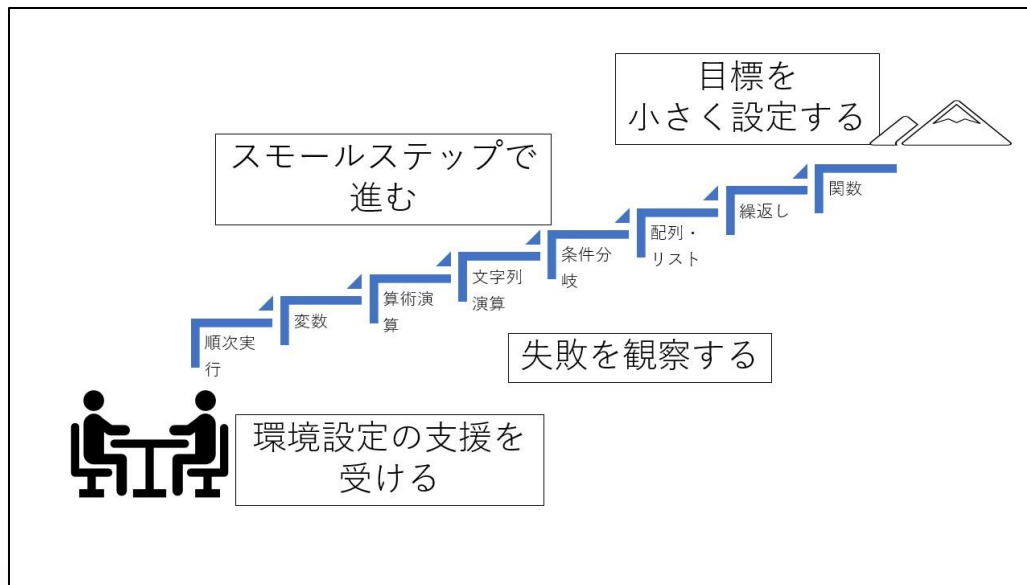
■ 経歴

- 大学でプログラミングを初めて学ぶ
- IT企業でエンジニアとして勤務する
 - その間、複数のプログラミング言語を習得
- **ITトレーニング会社**で、プログラミングなどの**講師**をつとめる
- ITサービス会社で、自社製品の講師をつとめる



目次

1. プログラミング学習のハードル
2. 最初のハードルの越え方
3. 次のハードルはどこにある？
4. まとめ



参考：プログラミング学習ツールBEIの紹介

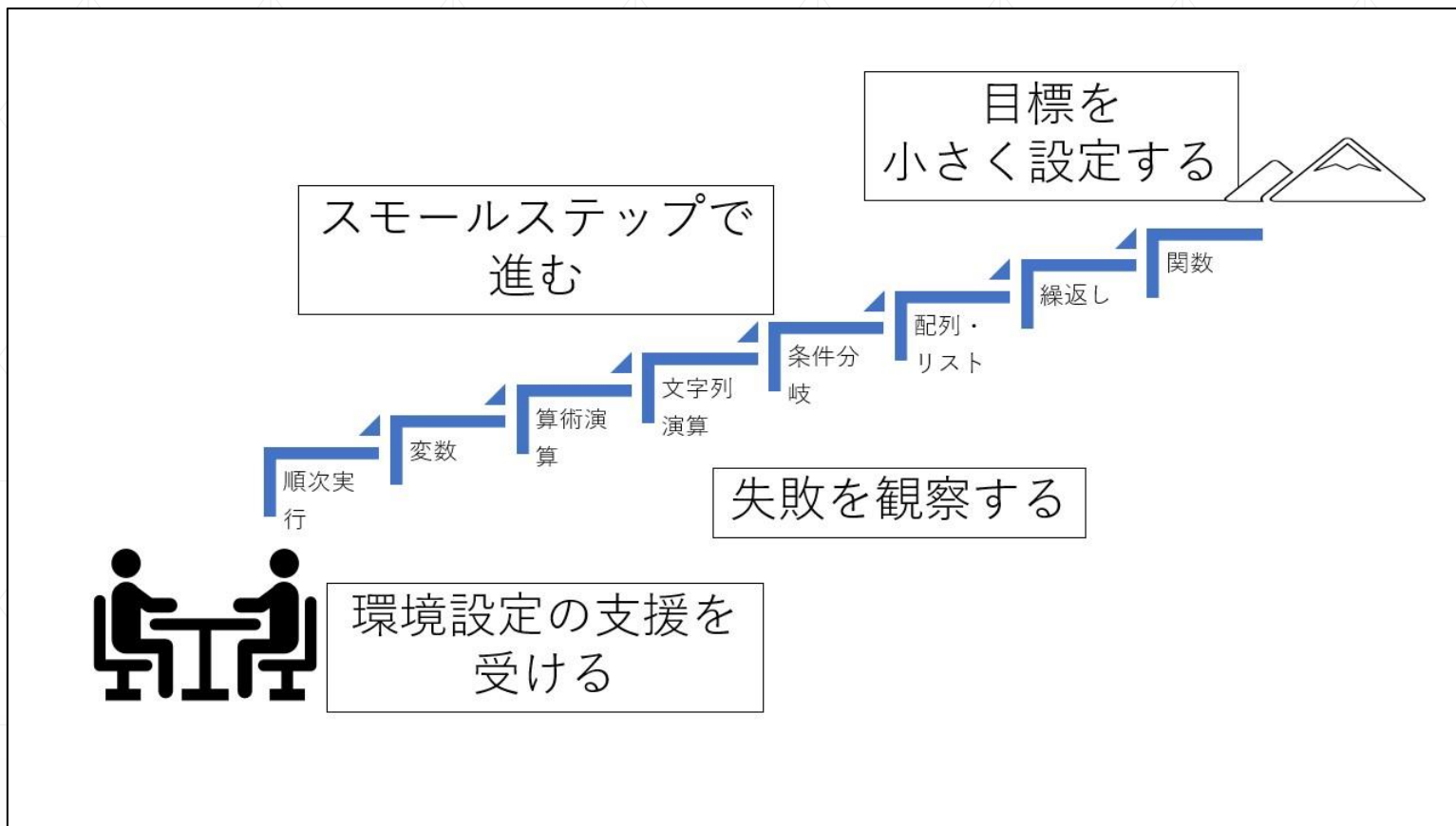


プログラミング学習のハードル

プログラミング学習のハードル

- **プログラミング環境をどう準備したら**よいか分からない
 - 何を使う？
 - どうインストールする？
 - アカウントが必要？
- **どこまで学べばよいか**分からない
 - プロのエンジニアのようにならなければいけない？
- 学習を**どう進めたらよいか**分からない
- 「失敗」が怖い
 - 「プログラミングに**失敗したら、どうしよう？**」

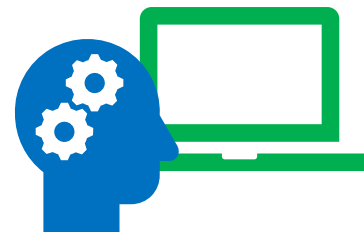




最初のハードルの越え方

PCの管理者を探す

- まず、PCに、プログラムを作成するためのツール、ソフトウェアを**インストール**する
 - プログラム実行環境
 - プログラム開発環境
 - 開発支援ツール（エディタ、デバッガなど・・・統合開発環境）
- 管理者がいるPCだと、**管理/制限**がかけられていて、一般のユーザーには、そのインストールができない/難しい場合がある
 - Microsoft Active Directoryによる管理
 - Google for Educationによる管理
 - 部外者には、その制限を超える手段は分からない（不正な方法で制限を超えると、違法になる可能性もある）
 - 管理者と相談の上、**正当な手段**でツール類をインストールする
- そもそも、インストールしたいツールやソフトウェアが、そのコンピュータでは**サポート**されていない場合もある



自分のPCを買って、「PCの管理者」になろう

- **事情が許せば**、「プログラミングを学習するための自分のPC」を買い、そのPCの管理を自分で行うとよい
- メリット
 - 自分のPCなので、好きなようにソフトウェアをインストールすることができる
 - 気に入らなければ、設定をやり直すこともできる
 - もちろん、プログラミング以外にも使える
- デメリット
 - 学ぶことが増える（OSの設定・管理）
 - データのバックアップ、定期的なアップデート、セキュリティの配慮など、継続的な取り組みも必要



Windows / Mac / Chromebook どれにする？

- 一長一短
- 昔は「ビジネスならWindows、デザインならMac」などと言われていたが、今はプログラマーもMacを選んでいる



学習指導要領/教科書/指導計画を見直す

- 学ぶ範囲は、自分で決めてよい
 - プロのエンジニアでも、「とりあえず自分が今必要としている範囲（+アルファ）」を目標とし、必要に応じて学び足していくことがほとんど

▪ 学校の場合

- 学習指導要領の「プログラミング」
 - 小学校・中学校・高校
- 教科書
- 教科書会社・教材会社の示す指導計画
- 入試の範囲



高校『情報Ⅰ』の教科書の
“プログラミングの基礎”の範囲

- 制御構造
 - 順次
 - 分岐
 - 繰返し
- データ構造
 - 変数
 - 配列/リスト
- 関数



スモールステップで少しずつ取り組む

- いきなりゴールにたどり着く必要はない

- 一つずつ取り組む



- **前に戻るのを気にしない**

- 後から学んだ概念によって、前に学んだ概念が分かる場合は多い

- 例) 順次実行：「プログラムに書いてある順に処理を実行する構造」

- 当たり前に見える
- 分岐（条件によって処理の流れを変える）や、繰り返し（条件に応じて処理を繰り返す）など、

「プログラムに書いてある順ではない順番で処理を実行する構造を学ぶと、その意義がはっきり分かる



「エラーメッセージ」と付き合う

- 失敗は、どんなベテランプログラマーもする

- いちいちガッカリする必要はない

```
print("Hello, Python!)  
      ^  
SyntaxError: unterminated string literal (detected  
at line 1)
```

- プログラマーの対処法

- 失敗を早く見つけるように取り組む

→ いちどに沢山のプログラムを書かない。**少しずつ書いて、試す**

- 失敗が起きたら、起きている**現象をよく観察する**

→ 出来事を順番に書き出す

→ プログラムを一行ずつ実行する

- まず、エラーメッセージをよく読む**



次のハードルはどこにある？

高度な機能を持つプログラムを作成するには

次のハードル

- 「高度で、複雑なプログラム」は必須ではない
 1. 要件（何を作るか、何を実現するか）を分析・整理する
 2. 核になる部分をしっかり調べて、学び、作る
 - 全部自分で作る必要はない
 - ライブラリを探す
 - お手本や先例を探す
- 1. 2. を繰り返して、**出来ることを増やしていく**



まとめ： 成人のプログラミング学習の ハードルとその越え方

環境の準備、学習範囲の限定、スモールステップ、「失敗」の学習

まとめ

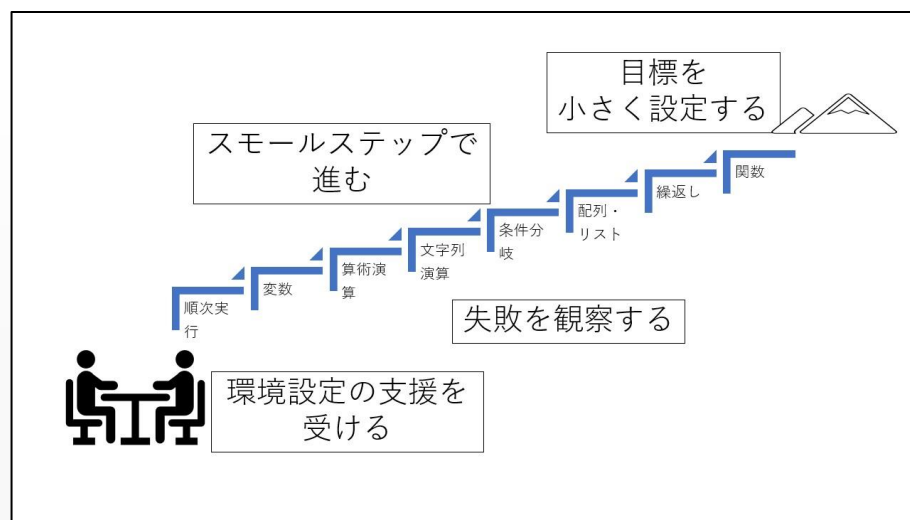
1. プログラミング学習のハードル

- プログラミング環境をどう準備したらよいか分からない
- どこまで学べばよいか分からない
- 学習をどう進めたらよいか分からない
- 「失敗」が怖い

2. 最初のハードルの越え方

- 「PCの管理者」
- 学習指導要領/教科書/指導計画を見直す
- スモールステップで少しずつ取り組む
- エラーメッセージと付き合う

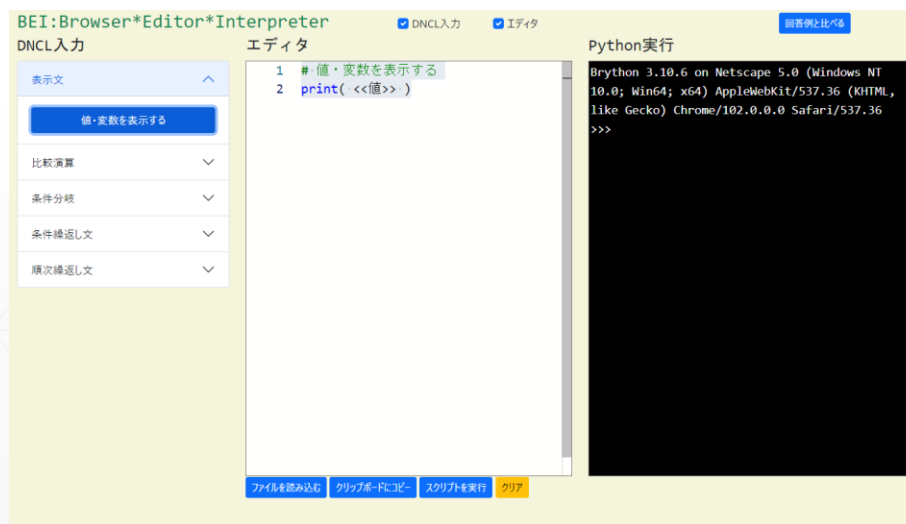
3. 次のハードルはどこにある？



参考：
プログラミング学習ツール
BEIの紹介

BEI (バイ: Browser Editor Interpreter) とは

- プログラミング言語 Python によるプログラミング学習ができるツール
- WebブラウザからBEIのURLにアクセスするだけで起動
- DNCL入力機能で、大学入試で使う語彙に慣れる



※BEIは、井澤が研究で作成したツールです



BEIの画面

The screenshot shows the BEI interface with the following components and callouts:

- パネルの表示/非表示**: A callout pointing to the top navigation bar with buttons for "DNCL入力", "エディタ", and "回答例と比べる". Below the bar, text reads "DNCL入力、エディタの表示/非表示を切り替える".
- 回答例と比べる**: A callout pointing to the "回答例と比べる" button, with text: "作成したプログラムと、回答例を比べる画面を呼び出す".
- Python実行**: A callout pointing to the "Python実行" button, with text: "プログラムを実行し、結果を表示する".
- メニューボタン**: A callout pointing to a row of buttons: "ファイルを読み込む", "クリップボードにコピー", "スクリプトを実行", and "クリア". The text below lists: "作成したプログラムを実行する" and "クリップボードにコピーする".
- エディタ**: A callout pointing to the code editor area, with text: "Pythonプログラムを書きます。Pythonの予約語等は強調表示されません。".
- DNCL入力**: A callout pointing to the "DNCL入力" panel, with text: "ボタンを使ってプログラムのキーワードを入力できます。".

The interface itself is titled "BEI:Browser*Editor*Interpreter" and contains three main sections:

- DNCL入力**: A panel on the left with a "表示文" dropdown and a "値・変数を表示する" button. Below are sections for "比較演算", "条件分岐", "条件繰返し文", and "順次繰返し文".
- エディタ**: A central text area containing Python code:

```
1 print("Hello, Python!")
2 for i in range(1,11):
3     if i%2==0:
4         print(i,"は偶数")
```
- Python実行**: A terminal window on the right showing the output:

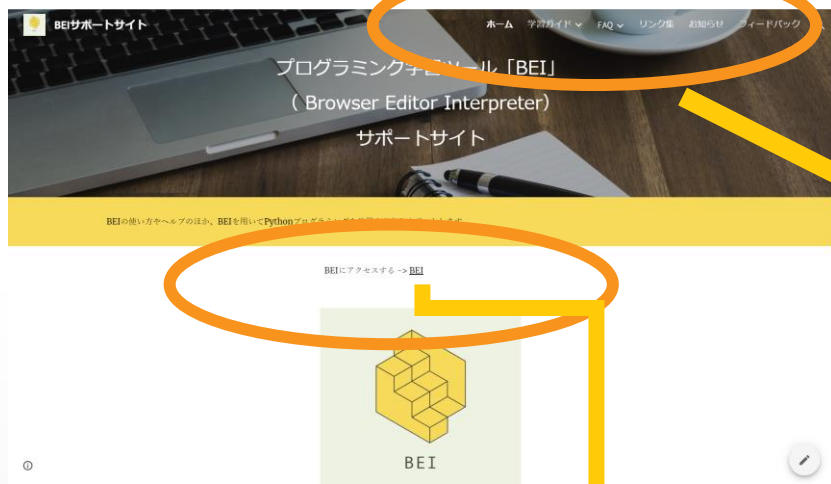
```
Brython 3.10.5 on Netscape 5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/101.0.4951.67 Safari/537.36
>>>
```



サポートサイト

(<https://sites.google.com/view/beioujizawa>)

- BEIを使ってPythonプログラミングを学習する人に、資料・教材・情報を提供します
- BEIをにアクセスするためのリンクがあります
- 画面右上の「学習ガイド」メニューに、「課題・回答例・サンプル」というメニューがあります



ブラウザの別のタブでBEIが開く



BEI