

実習課題

どれでも好きな課題を選んで試してみてください。

プログラムを使わない編

1. 企画・業務効率化

図書館で実施する企画や業務効率化などについて、自分でやってみたいことを考えて生成 AI を活用して作成・実施してください

例)

- ・ 生成 AI を活用した館内案内の作成
- ・ 本の紹介文をもとにした Blind Date with a Book（中身が見えない図書のパッケージ貸出）の作成
- ・ 図書の購入希望をもとにした需要分析
- ・ 貸出データと予約待ち人数をもとにした重複図書必要冊数の算出

プログラムお試し編

2. html と JavaScript で動作する電卓

html と JavaScript で動作する電卓を作成してください。

(ヒント)

プログラム作成の条件（一部）

- 数字ボタン(0-9)
- 演算子(+,-,*,/)
- = ボタン
- C ボタン

3. Power Shell で動作する電卓

PowerShell 5.1 で動作する GUI 電卓を作成してください。

(ヒント)

プログラム作成の条件（一部）

- WinForms のみ使用
- 1 ファイルの.ps1 で完結
- 一般的な電卓のレイアウト

4. QR コード作成

テキストを入力すると QR コードを作成するプログラムを作成してください。

(ヒント)

言語 : html と JavaScript

5. ヒストグラム（度数分布）作成

データ列（複数の数値）をコピーするとヒストグラム（度数分布）を作成するプログラムを作成してください。

(ヒント)

言語 : html と JavaScript

プログラム作成の条件（一部）

- 分割数（ビン数）を入力
- 改行区切りの数値データを貼り付け

図書館業務のためのプログラム編

6. ISBN から書誌情報を取得

図書の ISBN リスト(10~20 冊)を書き込むと、openBDAPI へ問い合わせ、書名、著者名、出版社、紹介文のウェブページで表示し、csv ファイルを出力するシステムを作ってください。

(ヒント)

API の URL : <https://api.openbd.jp/v1/get> (最重要事項 : URL のドメインやパスを 1 文字でも省略・変更することを厳禁とする)

言語 : html と JavaScript

7. 入館データ分析

入退館時の記録データ（入館データ_サンプル.csv）を読み込んで、以下のデータを集計してウェブページでグラフ表示してください。

- ・ 年間入館者数
- ・ 月別入館者数
- ・ 曜日別入館者数
- ・ 時間帯別入館者数

- ・ 平均滞在時間
- ・ 滞在時間の分布

(ヒント)

言語 : html と JavaScript

8. 図書館統計調査票作成支援

貸出データ（貸出データ_サンプル.csv）と ILL データ（ILL データサンプル.csv）のファイルを読み込んで、以下のデータをウェブページで画面出力してください。また、表示結果を csv 形式でダウンロードしてください。

【貸出データ】

- ・ 利用者区分別貸出冊数（貸出データの行数を利用者区分ごとに集計）
- ・ 資料種別別貸出冊数（貸出データの行数を資料種別ごとに集計）

【ILL データ】

- ・ 利用者区分別 ILL 依頼件数（ILL データの行数を利用者区分ごとに集計）

(ヒント)

言語 : html と JavaScript

9. 返却実績日数の分布調査

貸出返却データ.csv ファイルを読み込んで、返却期限日を基準として、何冊の資料が、どのくらい早く（または遅く）返却されたかを確認できるウェブページを作成してください。

(ヒント)

表示項目 : 返却期限からの日数差、件数、分布（簡易グラフ）

言語 : html と JavaScript

10. MARC データを利用したメール文面の作成

以下の手順で、利用者に新着電子書籍を案内するメール文章を作成するプログラムを作成して下さい。

1. 電子書籍の MARC ファイル（MARC サンプルファイル.txt）を読み込む。
2. 読み込んだ情報をもとに、書名、著者、電子書籍のリンク先をウェブページに表示する。
3. その情報をもとにメールの案内文をウェブページに作成する。

(ヒント)

言語 : html と JavaScript

11. きまぐれ OPAC 検索

ランダムに検索語を取得できる API から検索語を取得して、その検索語を自館の OPAC の検索 (クエリー) にセットして検索を実行できるウェブページを作成してください。

(ヒント)

API : <https://random-words-api.vercel.app/word/japanese>

言語 : html と JavaScript

まず、自館の OPAC の URL にどのような形式でキーワードをセットできるかを調べましょう。OPAC 検索を URL から行う場合は、基本的に①ドメイン (どの Web サイトか) と②パス (そのサイトのどの機能を使うか)、③クエリー (その機能にどんなキーワード等を渡すか) の 3 つの要素が必要です。

例) 広島都市学園大学附属図書館の OPAC 検索

① DOMAIN = 'https://tosyo-hcu.opac.jp'

② PATH = '/opac/Advanced_search/search'

③ QUERY = '?q=' + [ここに取得したワード] +
'&q_m=1&q_c=2&jof1=1&mtl1=1&mtl4=1&mtl5=1'

詳細は森保さんの発表資料「8. 自館 OPAC の検索結果直リンクの方法」を参照してください。

プログラム応用編

12. フォルダ検索

指定したフォルダ内の Word 文書 (.docx) を一括検索し、指定した文字列を含む文書を一覧表示する GUI ツールを作ってください。動作環境は PowerShell 5.1 + Windows 11 + Microsoft Word だけで動くものとします。

(ヒント)

プログラム作成の条件 (一部)

- コードはできるだけ短く単純にする
- 一発で動くことを最優先する

- 完成した.ps1 コードを 1 本だけ提示する

13. 帳票作成

以下のプロンプトを基に動作するコードを完成させた後、プロンプトを少し変えるなどして、帳票のグレードアップをしてください。

---以下プロンプト---

PowerShell 5.1・Windows 11 標準環境だけで動く GUI ツール（フォーム使用）を作ってください。

- ・ CSV（UTF-8 BOM 付き）を読み込み、Word テンプレートに差し込み、教員ごとに PDF を作成する。
- ・ CSV 列：教員名,管理番号,区分,書誌情報
- ・ Word テンプレートは <日付>、<教員名>、<書籍リスト表> を含む
- ・ <日付> には今日の日付を置換
- ・ <教員名> には CSV の教員名を置換
- ・ <書籍リスト表> → その教員の CSV データから作成した Word 表に置換
- ・ 表の列は「管理番号」「区分」「書誌情報」
- ・ 表はシンプルなものにする
- ・ <書籍リスト表> の位置に表を挿入する（センタリング）
- ・ 罫線は COM 操作で直接設定
- ・ Word オブジェクトの『\$wordApp.Dialogs.Item(84)』（wdDialogFileSaveAs）を使用し、初期パス（PDF 名）と Format=17 を設定した上で『.Show0』メソッドで Word 標準の保存ダイアログを強制的に最前面に出してユーザーに保存させる
- ・ 同名のファイル名がある場合もダイアログは表示させる
- ・ 保存ダイアログが終了したらすぐ次に進む
- ・ Word は表示しながら処理する
- ・ 教員ごとに「教員名.pdf」を作成する
- ・ GUI には CSV、Word テンプレート、PDF 出力先の選択と実行、終了のボタンを配置する