

プログラム実習カリキュラム

本カリキュラムは、文部科学省の教育推進事業委託費による委託事業として、学校法人河原学園 河原電子ビジネス専門学校が実施した令和3年度「専修学校における先端技術利活用実証研究」の成果物です。

Python 実習授業（基礎編・応用編） カリキュラム案

番号	章主題	節	概要・記載事項
1	プログラミング・Pythonの基本知識	コンピュータとソフトウェア	<ul style="list-style-type: none"> ・概要 一般的なコンピュータとソフトウェアについて解説する。 ・記載事項 コンピュータについて、主要装置（CPU、メモリ、ハードディスク、キーボード、モニター、プリンター等）を図示して、各装置の機能および役割を解説する。 ソフトウェアについて、オペレーティングシステム（OS）・インタプリタ・アプリケーション・ライブラリなど、ソフトウェアを実行する際の階層構造を図示して、各階層の詳細を説明する。 さらに、OS やアプリケーション・Web アプリケーション（Web アプリ）・ミドルウェアなど、いくつかのソフトウェアの種類について図示しながら解説する。
		プログラム	<ul style="list-style-type: none"> ・概要 プログラムがコンピュータ上で動く仕組みについて解説を行う。 ・記載事項 ソースコード・機械言語・アセンブリ言語などプログラムの設計情報とコンパイラやインタプリタなどの処理系について解説した上で、コンピュータがプログラムを読み取って実行するまでの流れを図示しながら解説する。
		プログラミング言語の種類	<ul style="list-style-type: none"> ・概要 いくつかのソフトウェアについて、構成部品（Web アプリの場合は、クライアントサイドおよびサーバサイド）を解説した上で、各部品を開発するとき用いるプログラミング言語について解説する。 ・記載事項 構成部品、および、各構成部品の関係性を示した上で、各部品の開発に利用可能なプログラミング言語の特徴を解説する。 例えば、Web アプリの場合は、クライアントプログラムを記述する言語として、

番号	章主題	節	概要・記載事項
			<p>HTML・CSS・JavaScript について解説した上で、JavaScript にトランスパイル可能な言語および JavaScript フレームワーク (React・Vue.js・AngularJS) について解説する。また、サーバプログラムを記述する言語として、Java・PHP・Python・JavaScript について解説する。</p> <p>同様にパソコンやスマートフォン向けのアプリケーションやスクリプトを記述するための言語についても合わせて解説する。</p>
		Python 言語	<ul style="list-style-type: none"> ・概要 プログラミング言語の中における Python の特徴について解説する。 ・記載事項 一般的なアプリケーションの開発において Python を使用する際と、機械学習・データ分析において Python を使用する際の相違点について解説した上で、各場面で利用するライブラリおよびフレームワークについて解説する。 <p>また、Jupyter Notebook および Google Colab などの実行環境について解説する。</p>
2	変数・データ型・代入・数値計算	変数	<ul style="list-style-type: none"> ・概要 Python における変数の仕組みについて解説する。 ・記載事項 変数とは、プログラムで使用するデータに名前をつけて保持しておくための概念であることを解説する。また、変数の使い方について、変数を定義して値を読み書きするサンプルプログラムを提示しながら解説する。
		代入	<ul style="list-style-type: none"> ・概要 Python における変数への代入について解説を行う。 ・記載事項 代入とは、変数に新たなデータを格納する処理であることについて、サンプルプログラ

番号	章主題	節	概要・記載事項
			ムを提示しながら解説する。また、代入文は、「=」の記号を用いて「(変数) = (値)」と記述することを解説する。
		データ型	<ul style="list-style-type: none"> ・概要 Pythonにおけるデータ型について解説する。 ・記載事項 基本的なデータ型である int 型、float 型、str 型、bool 型について、サンプルプログラムを提示しながら解説した上で、コレクションのデータ型 (list 型、tuple 型、set 型、dict 型) について簡単に解説する。
		演算子と数値計算	<ul style="list-style-type: none"> ・概要 Pythonにおける演算子と数値計算について解説する。 ・記載事項 演算子の一覧、計算時の優先順位、int 型と float 型の数値計算について、サンプルプログラムを提示しながら解説する。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った変数・代入・データ型・演算子・数値計算に関するプログラムを作成する課題を出題する。 ・記載事項 変数・代入・データ型・演算子・数値計算それぞれに関連した課題について、2問ずつ計10問を出題する。各問で想定する学生の取組時間は10分程度、ソースコード量は10行程度とする。
3	制御フロー	リストの基礎	<ul style="list-style-type: none"> ・概要 Pythonにおけるリストについて解説する。 ・記載事項 リストの基礎について解説する。リストリテラルの表記方法、リスト内の各要素へのアクセス方法について、サンプルプログラムを提示しながら解説する。

番号	章主題	節	概要・記載事項
		繰り返し (1)	<ul style="list-style-type: none"> ・ 概要 Python における for 文について解説する。 ・ 記載事項 for 文の構文と意味を説明した上で、for 文と range クラスを組み合わせた繰り返し方法について解説する。また、for 文はリストなどの指定した要素集合の各要素を列挙しながら、処理を繰り返すための構文であることについて、サンプルプログラムを提示しながら解説する。
		条件分岐	<ul style="list-style-type: none"> ・ 概要 Python における条件分岐について解説する。 ・ 記載事項 if 文および if 式の構文および意味を解説する。また、if 文および if 式を用いることで、指定した条件を満たしたときにだけ、特定の処理を実行できることについて、サンプルプログラムを提示しながら解説する。
		繰り返し (2)	<ul style="list-style-type: none"> ・ 概要 Python における while 文について解説する。 ・ 記載事項 while 文を用いた繰り返しについて解説する。while 文は指定した条件を満たす間、処理を繰り返すための構文であることについてサンプルプログラムを提示しながら解説する。
		繰り返しの制御	<ul style="list-style-type: none"> ・ 概要 Python における繰り返しの制御について解説する。 ・ 記載事項 無限ループについて解説した後に、break 文および continue 文を扱った繰り返し処理の制御方法について、サンプルプログラムを提示しながら解説する。

番号	章主題	節	概要・記載事項
		パターンマッチ	<ul style="list-style-type: none"> ・概要 Pythonにおけるパターンマッチについて解説する。 ・記載事項 match文について解説する。match文はPython3.10でサポートされた新機能であり、パターンごとに分岐を記述でき、マッチしたパターンの処理だけを実行できることについて、サンプルプログラムを提示しながら解説する。また、パターンマッチ特有の機能であるワイルドカードについても、サンプルプログラムを提示しながら解説する。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った条件分岐・繰り返し・match文に関するプログラムを作成する課題を出題する。 ・記載事項 if文・for文・while文・match文・break文・continue文について、それぞれ課題を1問ずつ、計6問を出題する。各問で想定する学生の取組時間は15分程度、ソースコード量は15行程度とする。
4	関数とメソッド	関数	<ul style="list-style-type: none"> ・概要 Pythonにおける関数について解説する。 ・記載事項 関数の定義、関数の呼び出し、引数、戻り値に関して解説する。特に引数については、多くのプログラミング言語でサポートされている位置引数について、サンプルプログラムを提示しながら解説する。 Python特有の機能であるキーワード引数、可変長引数、可変長キーワード引数について、サンプルプログラムを提示しながら解説する。また、呼び出し時の必須引数とオプション引数について、サンプルプログラムを提示しながら解説する。

番号	章主題	節	概要・記載事項
		オブジェクトとメソッド	<ul style="list-style-type: none"> ・概要 Pythonにおけるオブジェクトとメソッドについて解説する。 ・記載事項 オブジェクトの概要とオブジェクトの型について解説する。また、文字列やリスト型などの基本的な組み込み型のオブジェクトの持つ種々のメソッドとその使い方について、サンプルプログラムを提示しながら解説する。メソッドにより、各オブジェクトの内部の詳細を知らなくても、オブジェクトが保持するデータの操作が容易に行えることを示す。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った関数に関するプログラムを作成する課題を出題する。 ・記載事項 関数およびメソッドについて関連した課題4問を出題する。問題は条件分岐・繰り返しの内容も含むものとする。各問で想定する学生の取組時間は20分程度、ソースコード量は20行程度とする。
5	コレクション	コレクションの種類	<ul style="list-style-type: none"> ・概要 Pythonにおけるコレクションについて解説する。 ・記載事項 リスト、辞書、タプルの概要について解説する。いずれも複数のデータを保持するためのデータ構造であることは共通しているが、データの保持の仕方に違いがあり、置換、削除、挿入、検索、走査などの処理の性能に違いがあることについて、サンプルプログラムを提示しながら解説する。
		リスト	<ul style="list-style-type: none"> ・概要 Pythonにおけるリストについて解説する。 ・記載事項 リストとは、複数のデータに番号を付与して特定の順番で並べたデータ構造であることに

番号	章主題	節	概要・記載事項
			<p>ついて、サンプルプログラムを提示しながら解説する。</p>
		タプル	<ul style="list-style-type: none"> ・概要 Python におけるタプルについて解説する。 ・記載事項 タプルとは、ひとかたまりのデータを組みとして保持するデータ構造であることを解説する。 タプルはリストとよく似ているが、リストは複数のデータの繰り返しに用いることが通常であり、タプルは全体が一つのデータであって、各要素をデータの属性を表すために用いることについて、サンプルプログラムを提示しながら解説する。
		集合	<ul style="list-style-type: none"> ・概要 Python における集合について解説する。 ・記載事項 集合とは、重複を許さず、また、順序を保持せずに複数の値の集合を記録するデータ構造であることについて、サンプルプログラムを提示しながら解説する。
		辞書	<ul style="list-style-type: none"> ・概要 Python における辞書について解説する。 ・記載事項 辞書とは、複数のデータをキー（データ検索の ID）とバリュー（実際のデータ）の対応関係で保持するデータ構造であることについて、サンプルプログラムを提示しながら解説する。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱ったリスト・タプル・集合・辞書に関するプログラムを作成する課題を出題する。 ・記載事項 リスト・タプル・集合・辞書それぞれに関

番号	章主題	節	概要・記載事項
			連した課題を1問ずつ、また、複合的な課題2問の計6問を出題する。各問で想定する学生の取組時間は15分程度、ソースコード量は15行程度とする。
6	クラス	オブジェクト指向プログラミング	<ul style="list-style-type: none"> ・概要 オブジェクト指向プログラミングについて解説をする。 ・記載事項 旧来の手続き型プログラミングとモジュール化の方針について比較しながら、オブジェクト指向プログラミングについて解説をする。また、Pythonはオブジェクト指向プログラミング言語の一種であることを解説する。
		クラス	<ul style="list-style-type: none"> ・概要 Pythonにおけるクラスについて解説をする。 ・記載事項 クラスの定義（コンストラクタ・インスタンス変数・インスタンスメソッド）と定義したクラスのインスタンスの生成方法を解説する。また、Pythonの組み込み型であるリスト・タプル・集合・辞書にもクラスがあり、各組み込み型のリテラルはインスタンス生成の簡略表記に相当することを解説する。
		継承	<ul style="list-style-type: none"> ・概要 Pythonにおけるクラスの継承について解説をする。 ・記載事項 クラスの継承の方法および意味論とともに、継承によるモジュール化のメリットを解説する。さらに、オブジェクト間の継承関係を確認する方法について、isinstance関数とissubclass関数を扱ったサンプルコードを提示しながら解説する。また、多重継承について、基本的な概念のみを解説する。

番号	章主題	節	概要・記載事項
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱ったクラスおよび継承に関するプログラムを作成する課題を出題する。 ・記載事項 クラスに関連した課題を4問、継承に関連した問題を2問の計6問を出題する。各問で想定する学生の取組時間は15分程度、ソースコード量は15行程度とする。
7	モジュール	モジュール	<ul style="list-style-type: none"> ・概要 Pythonにおけるモジュールについて解説する。 ・記載事項 モジュール・パッケージ・ライブラリの特徴および相違点について解説する。 モジュールとは、1つのPythonファイルのことであることを解説する。パッケージとはPythonファイルを集めてディレクトリとし、外部に公開したものであることを解説する。ライブラリはパッケージをWeb上に公開し、誰でも利用可能なものとして保守・運用されているものであることを解説する。 モジュールの作り方、呼び出し方について、サンプルプログラムを提示しながら解説する
		パッケージ	<ul style="list-style-type: none"> ・概要 Pythonにおけるパッケージについて解説する。 ・記載事項 パッケージの作り方、呼び出し方について解説する。 通常のディレクトリをPythonパッケージとして認識させるための__init__.pyについて、サンプルプログラムを提示しながら解説する。さらに、import文の構文と意味を解説してから、絶対インポートと相対インポートを解説する。*を用いた全インポートについて、サンプルプログラムを提示しながら解説する。

番号	章主題	節	概要・記載事項
		標準ライブラリと外部ライブラリ	<ul style="list-style-type: none"> ・概要 Pythonにおける標準ライブラリと外部ライブラリについて解説する。 ・記載事項 標準ライブラリと外部ライブラリの違いと、具体例について解説する。 標準ライブラリとは、Pythonが公式に定めた、処理系が必ずサポートしなければならないライブラリのことであると解説する。一方、外部ライブラリとは、Pythonコミュニティのプログラマが独自に作成しているライブラリのことであると解説する。 Pythonは外部ライブラリが非常に充実しており、複雑な処理を初心者でも簡単に試用できることが大きな特徴であると述べる。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱ったモジュール、パッケージについての課題を出題する。 ・記載事項 モジュール、パッケージについて、実際に自作モジュール、自作パッケージを作って、呼び出し、実行するところまでを課題としてモジュール・パッケージを1問ずつ、複合的な問題を1問の計3問を出題する。各問で想定する学生の取組時間は30分程度、ソースコード量は30行程度とする。
8	入出力	標準入出力	<ul style="list-style-type: none"> ・概要 Pythonにおける標準入出力および標準エラー出力の方法を解説する。 ・記載事項 標準入出力および標準エラー出力の概念を解説した上で、Pythonにおける標準入出力および標準エラー出力の方法について、サンプルプログラムを提示しながら解説する。

番号	章主題	節	概要・記載事項
		ファイル入出力	<ul style="list-style-type: none"> ・概要 Pythonにおけるファイル入出力の方法を解説する。 ・記載事項 ファイル入出力の概念を解説した上で、Pythonにおいてファイルを開き、文字列データおよびバイナリデータを読み書きする方法について、サンプルプログラムを提示しながら解説する。
		クリーンアップ処理	<ul style="list-style-type: none"> ・概要 Pythonにおける with 文を解説する。 ・記載事項 ファイルなどのリソースは使用後に開放する必要があることを解説する。また、with 文の構文と意味を解説して、リソースを解放する方法として、close メソッドなどを呼び出す方法と with 文を使う方法の両方について、サンプルプログラムを提示しながら解説する。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った標準入出力・標準エラー出力、ファイル入出力、クリーンアップ処理についての課題を出題する。 ・記載事項 標準入出力・標準エラー出力、ファイル入出力、クリーンアップ処理それぞれに関する課題を各 1 問、計 3 問を出題する。各問で想定する学生の取組時間は 30 分程度、ソースコード量は 30 行程度とする。
9	エラーと例外	エラーの種類	<ul style="list-style-type: none"> ・概要 Pythonにおけるエラーの種類を解説する。 ・記載事項 Pythonには構文エラーと例外の 2 種類のエラーがあることを解説して、各種類のエラーの具体例やエラーから問題を突き止める方法について、サンプルプログラムとエラーの例

番号	章主題	節	概要・記載事項
			を提示しながら解説する。
		例外の処理	<ul style="list-style-type: none"> ・ 概要 Python における例外の処理方法を解説する。 ・ 記載事項 try 文、try 節、except 節、catch 節、finally 節の構文および意味を解説して、例外が送出されたときの対応方法を解説する。また、いくつかの具体的な例外を取り上げ、各例外の意味について、例外を処理するサンプルプログラムを提示しながら解説する。
		例外の送出	<ul style="list-style-type: none"> ・ 概要 Python における例外の送出方法を解説する。 ・ 記載事項 raise 文の構文および意味を解説して、例外の送出方法を解説する。また、from を用いて例外を連鎖させる方法を解説する。さらに、いくつかの具体的な例外を取り上げ、各例外の使い方について、例外を創出するサンプルプログラムを提示しながら解説する。
		独自例外の定義	<ul style="list-style-type: none"> ・ 概要 Python における独自例外の定義方法を解説する。 ・ 記載事項 既存の例外クラスを継承したクラスを定義することで、独自の例外を定義する方法を解説する。親クラスとなるいくつかの具体的な例外を取り上げ、各例外を継承して独自の例外を定義する方法について、サンプルプログラムを提示しながら解説する。

番号	章主題	節	概要・記載事項
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った例外の処理、例外の創出、独自例外の定義についての課題を出題する。 ・記載事項 例外の処理、例外の創出、独自例外の定義それぞれに関する課題を各1問、計3問を出題する。各問で想定する学生の取組時間は30分程度、ソースコード量は30行程度とする。
10	ソフトウェア品質とコーディング規約	ソフトウェア品質	<ul style="list-style-type: none"> ・概要 ソフトウェアの品質を評価するための基準を解説する。 ・記載事項 ソフトウェアの品質を考えるための概念として、ISO9126 および ISO/IEC 25000 (SQuaRE) の品質特性および品質副特性について解説する。
		コーディング規約	<ul style="list-style-type: none"> ・概要 ソフトウェア品質におけるいくつかの品質特性を高める道具の一つとしてコーディング規約を解説する。 ・記載事項 Python Enhancement Proposal (PEP) について解説をして、PEP 8 においてコーディング規約の提案があることを紹介する。さらに、PEP8 中のいくつかの規約を取り上げ、規約を満たすサンプルコードと満たさないサンプルコードの両方を比較することで、規約を守ることとソフトウェア品質を高めることの関係性について解説する。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱ったコーディング規約についての課題を出題する。 ・記載事項 コーディング規約に違反したソースコードが規約を遵守できるように修正する課題を4

番号	章主題	節	概要・記載事項
			問出題する。各問で想定する学生の取組時間は 20 分程度、ソースコード量は 40 行程度とする。