

Python 実習授業（基礎編・応用編） カリキュラム

基礎科目

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。

番号	章主題	節	概要・記載事項
			<p>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、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、計 8 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文に関連する課題について、計8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。
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8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10分程度、ソースコード量は10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

番号	章主題	節	概要・記載事項
			<p>について、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
			連した課題について、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	名前空間とスコープ	<ul style="list-style-type: none"> ・概要 Python における名前空間とスコープについて解説をする。 ・記載事項 名前空間およびそれと関連するスコープについて解説する。クラスは名前空間の仕組みを活用して実現されることを踏まえ、サンプルコードを提示しながら解説する。
		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。
		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。

番号	章主題	節	概要・記載事項
		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、<code>isinstance</code> 関数と <code>issubclass</code> 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。
		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 クラスに関連した問題および継承に関連した問題について、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。

番号	章主題	節	概要・記載事項
			<ul style="list-style-type: none"> ・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 <code>__init__.py</code> 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、<code>import</code> 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。<code>*</code>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。
		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、パッケージの呼び出し、実行に関する課題について、計 6 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。

番号	章主題	節	概要・記載事項
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。
		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。

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 本章で扱った内容のうち、標準入出力・標準エラー出力 s についての課題を出題する。 ・記載事項 標準入出力・標準エラー出力に関する課題について、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。
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。

番号	章主題	節	概要・記載事項
		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。
		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、例外の送出の定義についての課題を出題する。 ・ 記載事項 例外の処理、例外の送出に関する課題について、計 5 問を出題する。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 20 行程度とする。

番号	章主題	節	概要・記載事項
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問出題する。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10行程度とする。
11	探索アルゴリズム	時間計算量・空間計算量	<ul style="list-style-type: none"> ・概要 時間計算量と空間計算量について解説する。 ・記載事項 時間計算量と空間計算量を紹介し、プログラムの動的な品質についての考え方を解説す

番号	章主題	節	概要・記載事項
			<p>る。</p> <p>プログラムの入出力が等しくても計算量が異なる場合があることを解説するとともに、演習を通じて理解させる。</p>
		線形探索	<ul style="list-style-type: none"> ・概要 探索アルゴリズムの一つとして線形探索について解説を行う。 ・記載事項 線形探索の動作や特性、時間計算量・空間計算量について解説する。アルゴリズムの動作を抽象化した手順やサンプルプログラムを提示しながら解説する。
		二分探索	<ul style="list-style-type: none"> ・概要 探索アルゴリズムの一つとして二分探索について解説を行う。 ・記載事項 二分探索の動作や特性、時間計算量・空間計算量について解説する。アルゴリズムの動作を抽象化した手順や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 本章で扱った線形探索、二分探索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 線形探索、二分探索に関連した課題を計4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20行前後とする。
12	ソートアルゴリズム	バブルソート	<ul style="list-style-type: none"> ・概要 バブルソートについて解説を行う。 ・記載事項 簡単なソートアルゴリズムの一つであるバブルソートを解説する。擬似言語で記述されたアルゴリズムを読んで理解させ、演習として、ある具体的な入力に対する実行の様子をトレースさせる。

番号	章主題	節	概要・記載事項
		選択ソート	<ul style="list-style-type: none"> ・概要 選択ソートについて解説を行う。 ・記載事項 前提となるソート（並べ替え）アルゴリズムの要件について確認する。昇順・降順などの言葉の意味を確認する。 簡単なソートアルゴリズムの一つである選択ソートを解説する。擬似言語で記述されたアルゴリズムを読んで理解させ、演習として、ある具体的な入力に対する実行の様子をトレースさせる。
		挿入ソート	<ul style="list-style-type: none"> ・概要 挿入ソートについて解説を行う。 ・記載事項 簡単なソートアルゴリズムの一つである挿入ソートを解説する。擬似言語で記述されたアルゴリズムを読んで理解させ、演習として、ある具体的な入力に対する実行の様子をトレースさせる。
		マージソート	<ul style="list-style-type: none"> ・概要 マージソートについて解説を行う。 ・記載事項 簡単なソートアルゴリズムの一つであるマージソートを解説する。擬似言語で記述されたアルゴリズムを読んで理解させ、演習として、ある具体的な入力に対する実行の様子をトレースさせる。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱ったバブルソート・選択ソート・挿入ソート・マージソート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 バブルソート・選択ソート・挿入ソート・マージソートそれぞれに関連した課題について、計 8 問作成する。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 行前後とする。

番号	章主題	節	概要・記載事項
13	アプリケーション開発 (1)	コンソールアプリケーションの説明	<ul style="list-style-type: none"> ・概要 コンソールアプリケーションの概要について解説を行う。 ・記載事項 アプリケーションと、開発を行う上で最低限必要になるプログラムの雛形について解説する。雛形については、while 文を用いて無限ループを発生させ、input 関数、break 文などを用いて、特定の文字列を入力したら、無限ループを終了するといった簡素なプログラムを用意する。
		コンソールアプリケーションの実装	<ul style="list-style-type: none"> ・概要 開発するコンソールアプリケーションの仕様と実装方法について解説を行う。 ・記載事項 ゲームの内容は、フィールド上に置かれた敵オブジェクトを、同じフィールド上に駒を配置することで取るといったものである。ゲームの仕様について解説し、雛形を元に制御構文や関数、コレクションなど、クラス以外でこれまで扱った構文を使って実装方法について説明する。
		コンソールアプリケーションの追加実装	<ul style="list-style-type: none"> ・概要 コンソールアプリケーションの追加の仕様と実装方法について解説を行う。 ・記載事項 追加のゲームの仕様は、敵オブジェクトや駒をランダムで配置し、駒を動かして敵オブジェクトを取っていくという内容である。駒は1つだけ配置するものとする。ゲームの追加の仕様について解説し、これまで開発したゲームを元に制御構文や関数、コレクションなど、クラス以外でこれまで扱った構文を使って実装する方法を説明する。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱ったコンソールアプリケーション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

番号	章主題	節	概要・記載事項
			<p>本章で扱ったコンソールアプリケーションに修正を加える問題および本章で使用した特殊な記法に関する問題について、計 5 問を出題する。各問で想定する学生の取組時間は 25 分程度、ソースコード量は 30 行から 50 行程度とする。</p>
14	アプリケーション開発 (2)	クラスを使ったコンソールアプリケーションの実装	<ul style="list-style-type: none"> ・概要 クラスを使いコンソールアプリケーションを実装する方法について解説を行う。 ・記載事項 前章で作成したプログラムに、追加でクラスを用いた実装をするゲーム開発の解説を行う。フィールドに配置する敵オブジェクトや駒、フィールドなど複数のクラスの定義し、それぞれのインスタンスを生成するような実装方法の解説を行う。敵オブジェクトや駒の座標、フィールドの初期状態など、インスタンス生成時に各インスタンスのデータ属性を定義したいため、各クラスで__init__関数を用いた実装方法の解説を行う。 クラスを用いたアプリケーション開発を通してオブジェクト指向プログラミングの理解を促す。
		クラスを使ったコンソールアプリケーションの追加実装	<ul style="list-style-type: none"> ・概要 クラスを使ったコンソールアプリケーションの追加の仕様と実装方法について解説を行う。 ・記載事項 追加のゲームの仕様である配置する駒を複数フィールドに配置するという内容の実装方法について解説する。 追加の仕様の実装から、インスタンスの複雑な操作を行い、オブジェクト指向プログラミングへのより深い理解を促す。
		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を使ったコンソールアプリケーション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 本章で扱ったクラスを使ったコンソールア

番号	章主題	節	概要・記載事項
			アプリケーションに修正を加える問題について計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80行から110行程度とする。
15	アプリケーション開発 (3)	継承を使ったコンソールアプリケーションの実装	<ul style="list-style-type: none"> ・概要 継承を使いコンソールアプリケーションを実装する方法について解説を行う。 ・記載事項 前章までに作成したプログラムを元に、追加の仕様を実現したゲームを実装する方法を解説する。実装には、継承を用いる。追加の仕様として、異なる動きをする複数の種類の駒を実現する。この仕様を満たすために、各駒の共通するメソッドを持たせるような親クラスを定義し、異なる動きを実現するようなメソッドを持つ子クラスを定義するような実装方法を解説する。 継承を用いたアプリケーションを通してオブジェクト指向プログラミングの理解を促す。
		リファクタリング	<ul style="list-style-type: none"> ・概要 リファクタリングについて解説を行う。 ・記載事項 リファクタリングとは、外から見たプログラムの振る舞いを変えずに、容易な理解や修正を実現するようにプログラムの内部構造を改善することである。具体的には、子クラスで親クラスのメソッドをオーバーライドする箇所について、オーバーライドするクラスのメソッドの処理を再度記述しなくても済むように、<code>super</code> 関数を用いた実装に修正を行う解説をする。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った継承を使ったコンソールアプリケーション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 本章で扱ったコンソールアプリケーションに修正を加える問題および本章で使用した特

番号	章主題	節	概要・記載事項
			<p>殊な記法に関する問題について、計 5 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 150 行前後とする。</p>
16	リストと配列	データ構造	<ul style="list-style-type: none"> ・概要 データ構造に関する基本的な概念を説明し、データ構造が時間計算量および空間計算量に与える影響を解説する。 ・記載事項 主要なデータ構造であるリスト・スタック・キュー・ハッシュテーブル・木・グラフの概要を解説する。さらに、データ構造の違いによって計算量が変わることを解説する。
		単方向リスト	<ul style="list-style-type: none"> ・概要 連結リストと単方向リストの概要を説明し、Python における実装方法を解説する。 ・記載事項 連結リストの特徴と、主な連結リストである単方向リスト・双方向リスト・循環リストの概要を解説する。単方向リストの概要を説明して、単方向リストにおける値の探索・挿入・削除方法について、サンプルプログラムを提示しながら解説する。
		双方向リスト	<ul style="list-style-type: none"> ・概要 双方向リストの概要を説明し、Python における実装方法を解説する。 ・記載事項 双方向リストの概要と、単方向リストとの相違点を解説する。さらに、双方向リストにおける値の探索・挿入・削除方法について、サンプルプログラムを提示しながら解説する。
		循環リスト	<ul style="list-style-type: none"> ・概要 循環リストの概要を説明し、Python における実装方法を解説する。 ・記載事項 循環リストの概要と、単方向リスト・双方向リストとの相違点を解説する。さらに、循

番号	章主題	節	概要・記載事項
			環リストにおける値の探索・挿入・削除方法について、サンプルプログラムを提示しながら解説する。
		Python における配列	<ul style="list-style-type: none"> ・概要 Python の array モジュールの使い方を説明して、リストと array の違いを理解する。 ・記載事項 Python の array モジュールの概要と、配列とリストの相違点を解説する。さらに、配列における値の探索・挿入・削除方法について、サンプルプログラムを提示しながら解説する。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った単方向リスト・双方向リスト・循環リスト・array モジュールを使用するようなプログラムの実装を求める課題を出題する。 ・記載事項 単方向リスト・双方向リスト・循環リスト・配列それぞれに関連した問題について計 6 問を出題する。各問で想定する学生の取組時間は 25 分程度、ソースコード量は 30 行前後とする。
17	スタックとキュー	スタック	<ul style="list-style-type: none"> ・概要 スタックの概要を説明し、データの追加および取り出し方法を解説する。 ・記載事項 スタックは後入れ先出し (LIFO: Last In First Out) という性質を持つデータ構造であり、要素が追加される際には一番上に積み重ねられ、取り出される際には一番上の要素から取り出されることを説明する。スタックの振る舞いやその使用例について、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
		キュー	<ul style="list-style-type: none"> ・概要 キューの概要を説明し、データの追加および取り出し方法を解説する。 ・記載事項 キューは先入れ先出し（FIFO: First In First Out）という性質を持つデータ構造であり、新しく挿入される要素はキューの最後に追加され、取り出される要素は最初に入れた要素から順に取り出されることを説明する。キューの振る舞いやその使用例について、サンプルプログラムを提示しながら解説する。
		優先度付きキュー	<ul style="list-style-type: none"> ・概要 優先度付きキューの概要を説明し、データの追加および取り出し方法を解説する。 ・記載事項 優先度付きキューは、各要素が一定の優先度を持ち、優先度が高い要素が優先的に取り出されるようなデータ構造であることを説明する。優先度付きキューの振る舞いやその使用例について、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 本章で扱ったスタック・キュー・優先度付きキューを使用するようなプログラムの実装を求める課題を出題する。 ・記載事項 スタック・キュー・優先度付きキューそれぞれに関連した問題について計7問を出題する。各問で想定する学生の取組時間は25分程度、ソースコード量は50行前後とする。
18	ハッシュテーブル	ハッシュテーブル	<ul style="list-style-type: none"> ・概要 ハッシュテーブルについて解説する。 ・記載事項 ハッシュテーブルは、キーと値をマッピングするデータ構造の一つであり、高速なデータ検索を可能とする特性から、頻繁に利用されるデータ構造であることを説明する。さらに、キーを元にハッシュ関数を計算し、その

番号	章主題	節	概要・記載事項
			結果を使ってデータへのアクセスすることを説明する。ハッシュテーブルの動作原理と構造について、サンプルプログラムを提示しながら解説する。
		オープンアドレス法	<ul style="list-style-type: none"> ・概要 オープンアドレス法によるハッシュテーブルにおける衝突解決策について説明する。 ・記載事項 衝突が起こった際に、他の位置へデータを再ハッシュすることで衝突を解決できる、また、ハッシュテーブル内の格納領域を効率的に利用することが可能となることを説明する。オープンアドレス法の具体的な動作とその効果について、サンプルプログラムを提示しながら解説する。
		チェイン法	<ul style="list-style-type: none"> ・概要 チェイン法によるハッシュテーブルにおける衝突解決策について説明する。 ・記載事項 衝突が起こった際、同じ格納位置にチェイン（連結リスト）を形成し、その中にデータを格納することで、衝突が生じた場合でもデータの追加や取得が可能になることを説明する。チェイン法の具体的な動作とその効果について、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 本章で扱ったハッシュテーブル・オープンアドレス法・チェイン法を使用するようなプログラムの実装を求める課題を出題する。 ・記載事項 ハッシュテーブル・オープンアドレス法・チェイン法それぞれに関連した問題について計5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行から80行前後とする。

番号	章主題	節	概要・記載事項
19	木構造	木構造	<ul style="list-style-type: none"> ・概要 木構造の概念と木を構成する各要素について説明する。 ・記載事項 木構造は情報科学における重要なデータ構造の一部であり、親子関係を持つデータを視覚化し、データ間の関係を簡単に理解することが可能であることを説明する。木構造の概念や性質、利点について解説し、簡単なサンプルプログラムを提示しながら解説する。
		二分木	<ul style="list-style-type: none"> ・概要 二分木の概念について説明する。 ・記載事項 二分木は木構造の一種であり、各ノードが最大で2つの子ノードを持つ性質を持つことを説明する。さらに、二分木の特性や操作方法について、簡単なサンプルプログラムを提示しながら解説する。
		完全二分木	<ul style="list-style-type: none"> ・概要 完全二分木の概念を説明する。 ・記載事項 完全二分木は二分木の特殊なケースであり、すべての葉が同じ深さを持つか、あるいは最下層だけが右側からいくつかの葉が欠けている特性を持つことを説明する。この特性により、データのバランスが保たれ、効率的な処理が可能となることを説明する。完全二分木の特性や利点について、簡単な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 本章で扱った木構造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 木構造・二分木・完全二分木それぞれに関連した問題について計5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50行前後とする。

番号	章主題	節	概要・記載事項
20	二分探索木	深さ優先探索	<ul style="list-style-type: none"> ・概要 木構造における深さ優先探索と探索順序について説明する。 ・記載事項 深さ優先探索は一つの経路を根から葉まで探索し、次に別の経路を同じように探索するアルゴリズムであることを説明する。また、3つの探索順序があり、先行順、中間順、後行順であることを説明する。深さ優先探索における各探索順序について、サンプルプログラムを提示しながら解説する。
		幅優先探索	<ul style="list-style-type: none"> ・概要 木構造およびグラフにおける幅優先探索について説明する。 ・記載事項 幅優先探索は木構造やグラフを探索する際のアルゴリズムで、深さ優先探索とは異なり、まず近いノードから探索を行うことを説明する。また、最初に見つけた解が最短経路となる問題等に適していることを説明する。幅優先探索の動きやその適用例について、サンプルプログラムを用いて詳細に説明する。
		二分探索木	<ul style="list-style-type: none"> ・概要 二分探索木概念と二分探索木に対する各種操作について説明する。 ・記載事項 二分探索木は探索木の種類であり、各ノードが最大で2つの子ノードを持つ特性があること、また、特定の情報を効率的に探索することが可能であることを説明する。二分探索木の作成方法やその動作について、サンプルプログラムを用いて詳細に説明する。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った深さ優先探索・幅優先探索・二分探索木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 深さ優先探索・幅優先探索・二分探索木それぞれに関連した問題について計5問を出題

番号	章主題	節	概要・記載事項
			<p>する。各問で想定する学生の取り組み時間は30分程度、ソースコード量は40行前後とする。</p>
21	複雑な木	平衡木	<ul style="list-style-type: none"> ・概要 平衡木の概念と平衡木の一種である AVL 木に対する各種操作について説明する。 ・記載事項 平衡木は、データの追加や削除といった操作を行っても常に木の高さがバランスを保つように設計されたデータ構造であり、バランスを保つことにより、検索や挿入、削除の操作を高速に行うことが可能となることを説明する。平衡木の性質や種類、その中でも AVL 木の操作方法について、サンプルプログラムを提示しながら解説する。
		多分木	<ul style="list-style-type: none"> ・概要 多分木の概念と多分木の一種である B 木に対する各種操作について説明する。 ・記載事項 多分木は、各ノードが2つ以上の子ノードを持つことが可能なデータ構造であり、階層的なデータの表現に利用され、木構造の中でも非常に柔軟性のあるデータ構造であることを説明する。多分木の特性、多分木の一種である B 木の操作方法について、サンプルプログラムを提示しながら解説する。
		ヒープ	<ul style="list-style-type: none"> ・概要 ヒープの概念とヒープに対する各種操作について説明する。 ・記載事項 ヒープは、親子関係についてのみ順序が保証されている木構造である。ヒープの特性やその操作方法について、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 本章で扱った平衡木・多分木・ヒープ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 平衡木・多分木・ヒープそれぞれに関連した問題について計3問を出題する。各問で想定する学生の取り組み時間は25分程度、ソースコード量は25行前後とする。
22	ビット演算	N進数	<ul style="list-style-type: none"> ・概要 進数の概念と異なる進数間の変換方法を説明する。 ・記載事項 N進数とは、数を表現する際に用いられる基数のことであり、最も一般的なものが10進数で、コンピュータの内部で2進数が主に用いられていることを説明する。サンプルプログラムを提示しながら、10進数から2進数や16進数への変換方法やその逆の変換方法を解説する。
		シフト演算	<ul style="list-style-type: none"> ・概要 シフト演算について解説する。 ・記載事項 シフトとは、ビット列を左右どちらかに動かす演算であり、ビットを左にシフトすると倍の値になり、右にシフトすると半分の値になることを説明する。サンプルプログラムを提示しながら、左シフトと右シフトの操作を具体的に解説する。
		論理演算	<ul style="list-style-type: none"> ・概要 論理演算について解説する。 ・記載事項 論理演算は、ビット単位でのAND、OR、XORなどの操作を行うものであることを説明する。さらに、論理演算の基本的な動作と、それを利用したプログラムの例について、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
		ビットマスク	<ul style="list-style-type: none"> ・概要 ビットマスクの概要とビットマスクによる計算量の削減方法を解説する。 ・記載事項 ビットマスクとは、特定のビットだけを選択的に操作するための方法であり、ビット演算の中でも特に利用頻度が高いことを説明する。また、ビットマスクを活用することで、時間計算量・空間計算量を削減できることを説明する。サンプルプログラムを提示しながら、ビットマスクの作成方法や利用方法を解説する。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱ったN進数・シフト演算・論理演算・ビットマスク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 N進数・シフト演算・論理演算・ビットマスクそれぞれに関連した問題について計7問を出題する。各問で想定する学生の取り組み時間は20分程度、ソースコード量は20行前後とする。
23	文字列処理	Pythonの文字列処理	<ul style="list-style-type: none"> ・概要 文字列処理のために提供されている標準ライブラリの機能を説明する。 ・記載事項 Pythonには、文字列を処理するための多くのメソッドが用意されており、文字列の置換、分割、結合等の操作だけでなく、大文字・小文字変換、前後の余白除去などの機能も含まれていることを説明する。提供されている文字列処理の各機能について、サンプルプログラムを提示しながら解説する。
		文字列アルゴリズム	<ul style="list-style-type: none"> ・概要 代表的な文字列アルゴリズムの概念と実装方法について説明する。 ・記載事項 文字列の検索、比較等の操作を行うための代表的な文字列アルゴリズムの概念と実装方

番号	章主題	節	概要・記載事項
			法を説明する。さらに、具体的として文字検索のアルゴリズムでは、線形探索法、KMP 法、BM 法を、文字列の比較としてレーベンシュタイン距離について紹介する。また、各種文字列アルゴリズムについて、サンプルプログラムを提示しながら解説する。
		データ圧縮	<ul style="list-style-type: none"> ・概要 データ圧縮アルゴリズムの概念と実装方法を説明する。 ・記載事項 ハフマン符号・連長圧縮・辞書式圧縮など、代表的なデータ圧縮アルゴリズムを紹介する。さらに、連長圧縮などの実装が容易なアルゴリズムについて、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 本章で扱った文字列処理・文字列アルゴリズム・データ圧縮アルゴリズム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 文字列処理・文字列アルゴリズム・データ圧縮アルゴリズムそれぞれに関連した問題について計 5 問を出題する。各問で想定する学生の取組時間は 25 分程度、ソースコード量は 25 行前後とする。
24	デザインパターン 1	デザインパターン	<ul style="list-style-type: none"> ・概要 ソフトウェアパターンおよびデザインパターンの基本概念を説明する。 ・記載事項 デザインパターンとは、ソフトウェア設計において頻出する一般的な問題を予め解決済みのパターンとして抽象化したものであり、開発者があらゆる状況における最適な設計方法を毎回見つけ出す必要性を減少させることで、生産性を向上させることを説明する。代表的なソフトウェアパターンおよびデザインパターンについて、具体的なパターンを提示しながら概要を説明する。

番号	章主題	節	概要・記載事項
		Singleton パターン	<ul style="list-style-type: none"> ・概要 Singleton パターンの概念・構造・実装例を解説する。 ・記載事項 Singleton パターンは、特定のクラスのインスタンスが1つしか存在しないことを保証するデザインパターンであることを説明する。Singleton パターンの使い所と実装する方法、メリットとデメリットについて、サンプルプログラムを提示しながら詳しく解説する。
		Decorator パターン	<ul style="list-style-type: none"> ・概要 Decorator パターンの概念・構造・実装例を解説する。 ・記載事項 Decorator パターンは、既存のオブジェクトに新たな機能を動的に付加することができるデザインパターンであり、クラスを新たに作成することなく、オブジェクトの動作を変更する手段として利用されることを説明する。Decorator パターンの使い所と実装する方法、メリットとデメリットについて、サンプルプログラムを提示しながら詳しく解説する。
		Strategy パターン	<ul style="list-style-type: none"> ・概要 Strategy パターンの概念・構造・実装例を解説する。 ・記載事項 Strategy パターンは、同一の問題を異なるアルゴリズムで解決することに焦点を当てたデザインパターンであり、アルゴリズムを切り替えることによってオブジェクトの振る舞いを動的に変更できることを説明する。Strategy パターンの使い所と実装する方法、メリットとデメリットについて、サンプルプログラムを提示しながら詳しく解説する。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った Singleton パターン・Decorator パターン・Strategy パターン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

番号	章主題	節	概要・記載事項
			Singleton パターン・Decorator パターン・Strategy パターンそれぞれに関連した問題について計 5 問を出題する。各問で想定する学生の取組時間は 25 分程度、ソースコード量は 30 行前後とする。
25	デザイン パターン 2	Adapter パター ン	<ul style="list-style-type: none"> 概要 Adapter パターンの概念・構造・実装例を解説する。 記載事項 Adapter パターンは、既存のクラスやインターフェースが新たに求められる要件や規格に合わない場合にその中間に介在する形で既存のクラスやインターフェースと新規の規格とを結びつける役割を果たすことを説明する。Adapter パターンの使い所と実装する方法、メリットとデメリットについて、サンプルプログラムを提示しながら詳しく解説する。
		State パターン	<ul style="list-style-type: none"> 概要 State パターンの概念・構造・実装例を解説する。 記載事項 State パターンは、オブジェクトが内部状態に応じて振る舞いを変更することを可能にするデザインパターンであり、状態の変化に応じて振る舞いを変える場合に有効であることを説明する。State パターンの概念や構造について説明し、使い所と実装する方法、メリットとデメリットについて、サンプルプログラムを提示しながら詳しく解説する。
		Factory Method パターン	<ul style="list-style-type: none"> 概要 Factory Method パターンの概念・構造・実装例を解説する。 記載事項 Factory Method パターンは、オブジェクトの生成をサブクラスに委譲することで、クラスの再利用性を高めることを説明する。Factory Method パターンの概念や構造について説明し、使い所と実装する方法、メリットとデメリットについて、サンプルプログラムを提示しながら詳しく解説する。

番号	章主題	節	概要・記載事項
		Template Method パターン	<ul style="list-style-type: none"> ・ 概要 Template Method パターンの概念・構造・実装例を解説する。 ・ 記載事項 Template Method パターンは、アルゴリズムの骨組みを定義しながら、一部の処理をサブクラスに任せるデザインパターンであり、オブジェクト指向プログラミングにおける継承・多態性を活用したパターンであることを説明する。Template Method パターンの使い所と実装する方法、メリットとデメリットについて、サンプルプログラムを提示しながら詳しく解説する。
		プログラム作成課題	<ul style="list-style-type: none"> ・ 概要 本章で扱った Adapter パターン・State パターン・Factory Method パターン・Template Method パターン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 Adapter パターン・State パターン・Factory Method パターン・Template Method パターンそれぞれに関連した問題について計 8 問を出題する。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 20 行前後とする。

応用科目

番号	章主題	節	概要・記載事項
1	データ処理	CSV ファイル	<ul style="list-style-type: none"> ・概要 CSV ファイルについて解説する。 ・記載事項 CSV は様々なアプリケーションで扱うことができる最も一般的なデータフォーマットの1つである。そのため、実務上 CSV 形式でデータを取り扱うことが多い。CSV の概要やデータフォーマットについて解説することで、CSV ファイルを扱ったデータ処理の理解を促す。
		CSV ファイルの読み込み	<ul style="list-style-type: none"> ・概要 Python での CSV ファイルを読み込む方法について解説する。 ・記載事項 Python において csv モジュールの csv.writer を使って CSV ファイルを読み込む方法について解説する。
		CSV ファイルの書き込み	<ul style="list-style-type: none"> ・概要 Python での CSV ファイルにデータを書き込む方法について解説する。 ・記載事項 Python において csv モジュールの csv.reader を使って CSV ファイルにデータを書き込む方法について解説する。
		CSV ファイルの集計・可視化	<ul style="list-style-type: none"> ・概要 Python での CSV ファイルを集計・可視化する方法について解説する。 ・記載事項 CSV ファイル内のデータを取り出し、集計を行い、そして集計した結果をコンソール上で可視化する方法について解説する。
		NumPy による配列の処理	<ul style="list-style-type: none"> ・概要 NumPy ライブラリの概要について説明した上で、NumPy が提供する配列機能の使い方を解説する。 ・記載事項

			<p>NumPy の配列と Python のリストの相違点について説明した上で、NumPy の配列において、Python のリストと同様に使える機能を中心にして、サンプルプログラムを提示しながら解説する。また、NumPy の配列と Python のリストを相互に変換する方法について、サンプルプログラムを提示しながら解説する。</p>
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った CSV ファイルの操作および NumPy ライブラリを使用するようなプログラムについて、作成を指示する課題を出題する。 ・記載事項 CSV ファイルの読み込み・書き込み、CSV ファイルの集計・可視化、NumPy による配列の処理に関する問題について、計 7 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 行前後とする。
2	pandas (1)	pandas の概要	<ul style="list-style-type: none"> ・概要 pandas ライブラリの概要と、データフレームの概要を解説する。 ・記載事項 データ分析作業を支援するモジュールを提供しているライブラリである pandas の概要と、pandas が提供するデータフレームの概念について解説する。
		pandas によるファイル読み込み	<ul style="list-style-type: none"> ・概要 Pandas を用いてファイルからデータを読み込む方法を解説する。 ・記載事項 pandas を用いて CSV および JSON ファイルを読み込み、データフレームを生成する方法について、サンプルプログラムを提示しながら解説する。また、生成したデータフレームの内容を表示する方法について、サンプルプログラムを提示しながら解説する。
		pandas によるデータフレームの操作 (基礎)	<ul style="list-style-type: none"> ・概要 pandas を用いてデータフレーム内のデータを操作する方法を解説する。 ・記載事項

			<p>pandas を用いて行や列を指定することによりデータを選択する方法、特定の条件を満たす行をフィルタリングする方法、特定の列を昇順や降順で並び変える方法、行や列をデータフレームに追加する方法、各行や各列に対して関数を適用することで値を変更する方法について、サンプルプログラムを提示しながら解説する。</p>
		pandas による統計量の計算	<ul style="list-style-type: none"> ・概要 pandas によりデータフレームの統計量を計算する方法を解説する。 ・記載事項 データから算出できる平均値や中央値、最大値、最小値、標準偏差、標本サイズなどの各種統計量の意味を解説した上で、pandas を用いてデータフレームの各種統計量を計算する方法について、サンプルプログラムを提示しながら解説する。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った pandas ライブラリを使用した基礎的なデータ操作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 pandas を用いた、ファイルの読み込みとデータフレーム生成、データフレーム内のデータ操作、統計量に関する問題について、計 8 問を提示する。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 行前後とする。
3	pandas (2)	pandas によるデータフレームの操作 (応用)	<ul style="list-style-type: none"> ・概要 pandas を用いてデータフレームを操作する応用的な手法について解説する。 ・記載事項 データフレーム内のデータを特定の条件でグループ化し集計する方法や、複数のデータフレームを結合して新たなデータフレームを作成する方法、データフレームの行と列を入れ替える方法などについて、サンプルプログラムを提示しながら解説する。
		pandas による欠損値や異常値	<ul style="list-style-type: none"> ・概要 pandas を用いて欠損値や異常値を確認する

		の確認	<p>方法を解説する。</p> <ul style="list-style-type: none"> ・ 記載事項 データの品質がデータ処理に与える影響の一例として、データに欠損値や異常値が含まれる場合に統計量やデータの傾向が大きく変化することを解説する。また、pandas を用いてデータフレーム内のデータの欠損値や異常値を確認する方法について、サンプルプログラムを提示しながら解説する。
		pandas によるデータクレンジング	<ul style="list-style-type: none"> ・ 概要 pandas を用いたデータクレンジングの方法について解説する。 ・ 記載事項 データの品質を上げるためのデータクレンジングについて、欠損値や異常値の置き換えや削除、表記ゆれの補正などの手法を解説する。また、pandas を用いてデータクレンジングを行う方法について、サンプルプログラムを提示しながら解説する。 さらに、データクレンジングを行った後のデータの統計量が実際に変化していることについて、サンプルプログラムとその実行結果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 pandas ライブラリによる応用的なデータ操作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 pandas を用いた、データフレーム内の応用的なデータ操作、欠損値や異常値の確認、データクレンジングに関する問題について、計 9 問出題する。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 行前後とする。
4	Web スクレイピング	Web スクレイピング	<ul style="list-style-type: none"> ・ 概要 Web スクレイピングについて解説を行う。 ・ 記載事項 分析対象のデータを収集する手法の一として Web スクレイピングを紹介する。HTML フォーマットを初めとするウェブ技術を解説

			し、Web スクレイピングに必要な前提知識を修得させる。
		HTTP リクエスト・レスポンス	<ul style="list-style-type: none"> ・概要 HTTP リクエスト・レスポンスについて解説を行う。 ・記載事項 HTTP リクエスト・レスポンスを解説したうえで、Python でリクエストを送信する方法、レスポンスを処理する方法について、HTTP リクエスト・レスポンスを扱うためのライブラリである requests を用いて解説する。
		HTML のパースおよびデータ抽出	<ul style="list-style-type: none"> ・概要 HTTP リクエストからデータを抽出する方法について解説を行う。 ・記載事項 Web スクレイピングの過程で取得できた HTML をパースし、目的のデータを抽出する方法を解説する。Web スクレイピングには、Python の Web スクレイピング向けライブラリである BeautifulSoup を用いる。
		プログラム作成課題	<ul style="list-style-type: none"> ・概要 本章で扱った Web スクレイピング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 Web スクレイピングに関連した問題について計 7 問作成する。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 30 行から 60 行程度とする。
5	Web スクレイピングしたデータの加工・分析	pandas を用いた Web スクレイピングデータの加工	<ul style="list-style-type: none"> ・概要 pandas を用いた Web スクレイピングデータの加工方法について解説を行う。 ・記載事項 前章で Web スクレイピングを用いて取得・抽出したデータをもとにした pandas ライブラリを使ったデータフレームの作成、分析が容易になるようにデータの加工を行うことで後に分析を行うための前処理についての解説をする。
		pandas を用い	<ul style="list-style-type: none"> ・概要

		<p>た Web スクレイピングデータの分析</p>	<p>pandas を用いた Web スクレイピングデータの集計方法について解説を行う。</p> <ul style="list-style-type: none"> ・ 記載事項 pandas ライブラリを用いて、欠損値の対処や表記ゆれの補正などのデータクレンジングを行ったうえで、データの集計や統計量の算出などの分析を行う方法について解説をする。
		<p>プログラム作成課題</p>	<ul style="list-style-type: none"> ・ 概要 本章で扱った pandas を用いた Web スクレイピングデータの加工と分析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 pandas を用いた Web スクレイピングデータの加工と分析それぞれに関連した問題について計 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 20 行から 110 行程度とする。