科目名	プログラミング基礎							年度	2026
英語科目名	Programming Basics							学期	前期
学科・学年	ＡＩシステム科　１年次	必／選	必	時間数	60	単位数	4	種別※	演習
担当教員	竹内 大吾	教員の実務経験		有	実務経験の職種		研究者		

【科目の目的】

Pythonを用いたプログラミングの基礎を習得し、他者のコードや生成AIのコードを理解し、実行できる能力を養う。仮想環境とGitによるバージョン管理を習慣化し、次世代AIエンジニアとしての基盤スキルを確立する。

【科目の概要】

Pythonの基本的な文法、データ構造、制御構文を学習し、実際に簡単なプログラムを作成する。開発環境の構築、Python仮想環境の利用、Gitを用いたバージョン管理の基礎を演習を通じて実践的に学ぶことで、機械学習やデータサイエンス科目で活かせるプログラミングスキルを身につける。

【到達目標】

Pythonの基本的な文法とデータ構造を理解し、簡単なプログラムを作成できる。Pythonの仮想環境を構築し、Gitを用いてコードのバージョン管理を行える。他者のコードを読み解き、自身の環境で実行できるスキルを習得し、生成AIが生成したコードも理解し修正できる。

【授業の注意点】

講義と演習を組み合わせた形式で進めます。積極的に手を動かし、疑問点は積極的に質問してください。基礎的な内容から着実に学ぶことが重要です。Pythonの環境構築は各自のPCで行い、スムーズな学習を心がけてください。

評価基準＝ルーブリック

ルーブリック 評価	レベル５ 優れている	レベル４ よい	レベル３ ふつう	レベル2 あと少し	レベル１ 要努力
到達目標 A	Pythonの主要な文法とデータ構造を自在に活用できる	Pythonの主要な文法とデータ構造を正確に利用できる	Pythonの基本的な文法とデータ構造を理解し利用できる	Pythonの基本的な文法の理解があと一歩必要である	Pythonの基本的な文法を理解するのに努力が必要である
到達目標 B	複数の環境構築を比較し最適解を導き実行できる	Pythonの実行環境を自力で構築し適切に実行できる	Pythonの実行環境を構築しプログラムを実行できる	環境構築の手順を理解するのにあと一歩必要である	環境構築やプログラム実行に努力が必要である
到達目標 C	Gitの高度な機能も活用し効率的に管理できる	Gitの基本操作とワークフローを正確に実践できる	Gitを用いたバージョン管理の基本操作を行える	Gitの基本操作の理解があと一歩必要である	Gitによるバージョン管理に努力が必要である
到達目標 D	他者の複雑なコード構造や意図を完全に理解できる	他者の書いたコードの構造と機能を正確に理解できる	他者の書いたコードの内容を基本的なレベルで理解できる	他者のコード理解に一部不明な点があと少しある	他者のコードを読み解くのに努力が必要である
到達目標 E	生成AIのコードを最適化し品質向上も可能である	生成AIのコードを理解し目的に合わせて修正できる	生成AIのコードを読んでその内容を理解できる	生成AIのコード理解にあと少し支援が必要である	生成AIのコードを理解するのに努力が必要である

【教科書】

配布資料または指定教科書

【参考資料】

独立行政法人情報処理推進機構（IPA）デジタルスキル標準（DX推進スキル標準）

【成績の評価方法・評価基準】

演習課題への取り組み状況と提出物、授業内でのパフォーマンスを総合的に評価します。到達目標の達成度に基づいて評価を行い、

※種別は講義、実習、演習のいずれかを記入。

科目名	プログラミング基礎	年度	2026
英語表記	Programming Basics	学期	前期

回数	授業テーマ	各授業の目的		授業内容	到達目標＝修得するスキル	評価方法	自己評価
1	プログラミングの概要	プログラミングの概念を理解	1	プログラミング	プログラミングの目的を説明できる	1	
			2	Pythonの概要	Pythonの特徴と利用例を説明できる		
			3	開発環境導入	開発環境の準備と設定ができる		
2	Python基礎１	データ型と変数を学ぶ	1	データ型	基本的なデータ型を識別できる	2	
			2	変数宣言	変数を宣言し値を代入できる		
			3	演算子	算術演算子を使用できる		
3	Python基礎２	条件分岐を使いこなす	1	if文の基本	if文の基本構造を理解できる	1	
			2	比較演算子	比較演算子を正しく使用できる		
			3	論理演算子	論理演算子を組み合わせて使える		
4	Python基礎３	繰り返し処理を学ぶ	1	for文	for文で繰り返し処理を記述できる	2	
			2	while文	while文で条件に応じた繰り返しができる		
			3	制御文	breakやcontinueを活用できる		
5	データ構造１	リストとタプルを使う	1	リストの操作	リストを作成し要素を操作できる	1	
			2	タプルの操作	タプルを作成し要素にアクセスできる		
			3	データの順序	リストとタプルの違いを説明できる		
6	データ構造２	辞書と集合を使う	1	辞書の操作	辞書を作成しキーで値にアクセスできる	2	
			2	集合の操作	集合を作成し要素を操作できる		
			3	データの関連	辞書と集合の使い分けを理解できる		
7	関数	関数を作成し活用する	1	関数の定義	関数を定義し呼び出すことができる	1	
			2	引数と戻り値	引数と戻り値を活用して関数を作る		
			3	スコープ	変数のスコープを理解し利用できる		
8	モジュールとPKG	コードを整理し再利用	1	モジュールの作成	Pythonモジュールを作成できる	2	
			2	インポート	モジュールをインポートし利用できる		
			3	標準ライブラリ	標準ライブラリの活用方法を理解できる		
9	仮想環境構築	仮想環境を活用する	1	利点と必要性	仮想環境の必要性を説明できる	2	
			2	venv/conda	仮想環境を構築し有効化できる		
			3	パッケージ管理	仮想環境でのパッケージ管理ができる		
10	Git基礎１	バージョン管理の導入	1	バージョン管理	バージョン管理の概念を説明できる	1	
			2	Gitの基本操作	Gitの初期化とファイル追加ができる		
			3	コミットと履歴	コミットで履歴を管理できる		
11	Git基礎２	GitHub連携を学ぶ	1	リモートリポジトリ	リモートリポジトリの概念を理解できる	2	
			2	プッシュとプル	変更をプッシュし更新をプルできる		
			3	リポジトリ管理	GitHubでリポジトリを管理できる		
12	ファイルの入出力	ファイル操作の基本	1	ファイルの読み込み	ファイルからデータを読み込める	1	
			2	ファイルへの書き込み	ファイルにデータを書き込める		
			3	エラー処理	ファイル操作のエラーを処理できる		
13	他者コード読解	他者コードの理解と実行	1	コードリーディング	他者の書いたコードを読み解ける	2	
			2	デバッグ技術	簡単なデバッグ手法を適用できる		
			3	ドキュメンテーション	コードの意図を把握し説明できる		
14	生成AIコード活用	AI生成コードの活用	1	AIコード確認	生成AIのコードを検証できる	2	
			2	コード修正	生成AIコードを修正し実行できる		
			3	効率的な利用	生成AIをプログラミングに活用できる		
15	総合演習	知識の統合と応用	1	複数機能開発	複数の機能を組み合わせた開発ができる	2	
			2	問題解決実践	実践的な課題解決に取り組める		
			3	コードレビュー	自身のコードを見直し改善できる		

評価方法：1.小テスト、2.パフォーマンス評価、3.その他

自己評価：S：とてもよくできた、A：よくできた、B：できた、C：少しできなかった、D：まったくできなかった

備考 等