

授業科目区分		科目名		単 位	科目コード	開講時期	履 修 方 法		
(全課程からの提供) リベラルアーツ系科目 文理横断		A I プログラミング入門(夏期集中講義)		1	G243-01	1 期 (前学期)	修学規程第 4 条を参照		
担当教員名		研究室	内線電話番号	電子メール I D			オフィスアワー		
授 業 科 目 の 学 習 ・ 教 育 目 標									
キーワード		学習・教育目標							
1	プログラミング	本科目では、以下が行えるようになることを目標とする。 ・ Python言語を用いて基本的なプログラムを作成でき、不具合の修正ができる ・ Pythonの構文、API、関連コマンドをインターネット、書籍から情報収集・活用できる ・ 他者が書いた基本的なPythonプログラムの処理を理解でき、説明できる ・ 処理概要から関数や制御構造を考え、プログラムを作成できる							
2	Python								
3	データ構造								
4	制御構文								
5	関数								
授業の概要および学習上の助言									
<p>本科目では、AIに関連する技術を扱う際に必須となっているプログラミング言語Pythonの基本的な内容について学ぶ。具合的には、変数、関数、制御文、繰り返し文などの基本的な内容に加えて、リスト、コンテナの処理、イテレータ、モジュールの利用などについて学ぶ。</p> <p>処理概要が与えられたら、入力、処理、出力を整理して、Python言語のプログラムを作成し、動作に問題があれば修正できる能力と、他者が書いたプログラムの処理内容を理解し説明できる能力を養う。</p> <p>以下のPythonプログラミングが行え、処理を説明でき、要求に応じて変更できることが学習の目標である。</p> <ul style="list-style-type: none"> ・プログラミング未経験者： 判断(if)、繰り返し(for)、入出力、式、関数 ・Python以外のプログラミング言語経験者： PythonのAPI・パッケージの利用、他プログラミング言語からの移植 ・Pythonプログラム経験者(情報工学科学生含む)： 辞書、リスト、集合などのデータ構造とパッケージの活用 									
【教科書および参考書・リザーブドブック】									
教科書：指定なし									
参考書：指定なし									
リザーブドブック：指定なし									
履修に必要な予備知識や技能									
本科目では、プログラミング未経験者でも対応できるよう基本的なことから説明しプログラミング演習を行う。履修者は、キーボードやマウスの操作ができること、インターネット情報検索ができることが必要である。既にPython言語の書籍を所有している履修者は持参して参照してもよい。									
No.	学科教育目標 (記号表記)	学生が達成すべき行動目標							
①	D,H	Pythonプログラムの作成方法、実行方法、不具合発生時の調査・修正方法を説明できる							
②	D,H	Python言語の基本的なデータ型や制御構造がわかり、サンプルプログラムの動作を説明できる							
③	D,H	簡単な処理の関数を作成でき、関数の機能、入力、出力を説明できる							
④	D,H	Pythonの言語仕様、API仕様を書籍やインターネットから調査することができる							
⑤	D,H	他者が書いた基本的なPythonプログラムを見て処理概要、入力、出力を説明できる							
⑥									
達 成 度 評 価									
評価方法		試 験	クイズ 小テスト	レポ-ト	成果発表 (口頭・実技)	作 品	ポ-トフォリオ	その他	合 計
指標と評価割合									
総合評価割合		0	0	0	30	40	0	30	100
総合力指標	知識を取り込む力	0	0	0	0	20	0	0	20
	思考・推論・創造する力	0	0	0	10	20	0	0	30
	コラボレーションとリーダーシップ	0	0	0	0	0	0	0	0
	発表・表現・伝達する力	0	0	0	20	0	0	0	20
	学習に取組む姿勢・意欲	0	0	0	0	0	0	30	30

※総合力指標で示す数値内訳は、授業運営上のおおよその目安を示したものです。

評価の要点

評価方法	行動目標	評価の実施方法と注意点	
試験	①		
	②		
	③		
	④		
	⑤		
	⑥		
クイズ 小テスト	①		
	②		
	③		
	④		
	⑤		
	⑥		
レポート	①		
	②		
	③		
	④		
	⑤		
	⑥		
成果発表 (口頭・実技)	①	レ	発表内容、プログラムのデモ、他者からの質問・アドバイスの応答を通じて、理解度を判断する。 他者に理解されやすい説明になっているか、発表態度についても評価する。
	②	レ	
	③	レ	
	④	レ	
	⑤	レ	
	⑥		
作品	①	レ	作成するプログラムについて、概要、入力、処理、出力が整理されているか、Python言語の機能を活用しているかを総合的に評価する。 可読性、メンテナンス性、他者からみた理解しやすさも評価する。
	②	レ	
	③	レ	
	④	レ	
	⑤	レ	
	⑥		
ポートフォリオ	①		
	②		
	③		
	④		
	⑤		
	⑥		
その他	①	レ	理解しようとする努力や演習の取り組みなどの学習態度について評価する。
	②	レ	
	③	レ	
	④	レ	
	⑤	レ	
	⑥		

具体的な達成の目安

理想的な達成レベルの目安	標準的な達成レベルの目安
標準的な達成レベルに加えて以下が行える 4) 作るべきプログラムの仕様を適切に説明できる 5) 授業で説明していないPython仕様を調査し実装できる 6) AI分野のPythonで書かれたサンプルプログラムでも不明な部分は単独で調査して理解を進めることができる	1) Python言語プログラミングの流れを理解しプログラム作成ができる 2) 関数、変数、制御構造、リストなどPython基本機能を理解し、入門書レベルの例題のプログラムを作ることができる 3) 言語仕様、API、パッケージなどの仕様調査を単独で行える

CLIP学習プロセスについて

一般に、授業あるいは課外での学習では：「知識などを取り込む」→「知識などをいろいろな角度から、場合によってはチーム活動として、考え、推論し、創造する」→「修得した内容を表現、発表、伝達する」→「総合的に評価を受ける、Good Work!」：のようなプロセス（一部あるいは全体）を繰り返し行いながら、応用力のある知識やスキルを身につけていくことが重要です。このような学習プロセスを大事に行動ください。※学習課題の時間欄には、指定された学習課題に要する標準的な時間を記載してあります。日々の自学自習時間全体としては、各授業に応じた時間（例えば2単位科目の場合、予習2時間・復習2時間/週）を取るよう努めてください。詳しくは教員の指導に従ってください。

回数 日付	学習内容	授業の運営方法	学習課題(予習・復習)	時間(分)※
1 /	ガイダンス Python言語の特徴 Pythonインストール演習（行わない場合もある） 対話型モード(REPL)の利用	講義・演習・質疑 実務家教員が担当する 資料は授業時に配布する	復習：配布資料でわからない部分の確認	15
2 /	簡単な計算 数学とプログラミングの違い 変数、定数、代入 データ型	講義・演習・質疑 実務家教員が担当する	復習：配布資料でわからない部分の確認	15
3 /	制御構造(if, for, while等) ブロック 関数	講義・演習・質疑 実務家教員が担当する	復習：配布資料でわからない部分の確認	60
4 /	テキストエディタ プログラム開発の流れ 構造的なデータ（リスト、辞書、タプル、集合）	講義・演習・質疑 実務家教員が担当する	復習：配布資料でわからない部分の確認、やり残した演習の検討	15
5 /	リスト内包 パッケージ、API、クラス パッケージ管理ツール	講義・演習・質疑 実務家教員が担当する	復習：配布資料でわからない部分の確認、やり残した演習の検討	15
6 /	例外処理 発表（または紹介）したいプログラムの作成・動作確認	講義・演習・質疑 実務家教員が担当する	予習：発表準備	100
7 /	作成したプログラムまたは紹介したいプログラムの発表と説明	発表・討議 実務家教員が担当する	復習：理解度の低い部分の復習	