2019年3月15日

加藤研究室

プログラミング学習マニュアル

~入門編~

川内敬介

北海道大学大学院 医理工学院 修士課程 2年

目次

はじめに	2
第1章 ~なぜ、プログラミングを学ぶのか~	3
1. そもそも、プログラミングとは何なのか	3
2. なぜ、加藤研究室ではプログラミングを学ぶのか	3
第2章 ~どのプログラミング言語を学ぶべきか~	4
① どのプログラミング言語を習得するべきか	4
② おすすめのプログラミング言語	4
第3章 ~プログラミングの学び方~	8
1. プログラミング学習の進め方	
2. プログラミング入門におすすめのホームページ	9
第4章 ~プログラミング環境構築~	10
1. python のインストール	
2. 仮想環境のすすめ(不要な場合、よくわからないという場合は3へ)	21
3. Visual Studio Code のインストールと設定	
第5章 ~困ったときの解決法~	41
1. 一番大事なことは「検索力」	
2. 検索のコツ	
3. 本当に困ったら素直に「質問」する	
第6章 ~今後のステップ~	43

はじめに

近年では人工知能に関する研究の発展により、コンピュータが囲碁の世界王者に 勝利、自動車の自動運転が現実味を帯びてくる等、様々な分野で成果を上げています。 この傾向は医療分野でも同様で、胸部レントゲン写真から肺結核の病変を自動検出したり、 脳 MR 画像から脳腫瘍を自動抽出したりというような形での応用研究が進められています。

本研究室でも時代の潮流に乗り、人工知能に関する研究を進めてきました。このよ うな研究を行うためにはまず、プログラミング言語を修得する必要があります。しかし、 本研究室に配属されたばかりの学生はそのほとんどがプログラミング言語に関する知識を 持ち合わせていないと思います(私もそうでした)。

そこで、本マニュアルでは「プログラミング言語を学ぶ理由」から「どのように 学んでいけばよいのか」までというプログラミング言語学習の導入部分をまとめました。 実際の研究はこのマニュアルを読み終えた後から始まります。サクッと読み終えてプログラ ミング言語学習を進めていただければと思います。なお、詳細な学習方法に関してはすべて 「おすすめのホームページ」に丸投げしています。

※本マニュアルの内容は1日で書き上げたため、内容の薄い部分が多分に存在します。
ご自由に加筆・修正し、より良いものに仕上げて頂けると幸いです。

2019年3月15日 川内敬介

2

第1章 ~なぜ、プログラミングを学ぶのか~

1. そもそも、プログラミングとは何なのか

加藤研究室に進学してきた学生のうちのほとんどが「そもそもプログラミングって 何?」とか「何となくイメージはできるけど、結局何ができるの?」とか感じていると 思います(かく言う私もそうでした)。詳しいことは Wikipedia でも参照してほしいと 思いますが、簡単にまとめると「人間がコンピュータにやってほしい事をコンピュータ にわかる言葉で説明する」ために行うのがプログラミングです。

2. なぜ、加藤研究室ではプログラミングを学ぶのか

本研究室は「**医用画像解析学研究室**」です。画像解析の際には既存のソフトウェアを 使用する方法もありますが、これではなかなか新しい研究が生まれません。そこで、本 研究室では自分たちでソフトウェアを開発し、画像解析を行っています。このソフトウ ェア開発にはコンピュータへの指示出し、つまりプログラミングの技術が必要です。

このようにして本研究室では、最新の研究テーマに取り組んでいます。

また、プログラミング言語を用いたソフトウェア開発の技術を生かし、企業へ就職 する事が可能です。実際、本研究室の医理工1期生は全員企業へ就職しています

(1期生就職実績:北海道電力、Panasonic、東芝、その他内定企業多数)。

以上を簡単にまとめると、「インターンシップ参加や企業就職の際にメチャメチャ 有利になるから勉強した方がいいよ」ということです(全然まとめてない…)。

第2章 ~どのプログラミング言語を学ぶべきか~

どのプログラミング言語を習得するべきか

プログラミング言語は様々ありますが、言語によって得手不得手があります。単純 に先輩が使っているからではなく、自分が将来どのようになりたいのかを考えた上で 学習する言語を決めるのが良いと思います。

次項に私のおすすめ言語をまとめておきましたので、参考にしてください。

ちなみに、プログラマとして働くことを考えている場合にはコンパイラ言語(C、 C++、C#、Java など)とインタープリタ言語(python、JavaScript、Ruby など)を一つ ずつ習得しておくと良いらしいです。

② おすすめのプログラミング言語

とりあえず、目標がない場合には python を学ぶ事をお勧めしています。 理由は、簡単な言語でありながら幅広い問題に対応できるからです。 詳細は以下で確認してください。

▶ インタープリタ言語編

コンパイラ言語と比較して、簡単に記述できる事を意識した言語が多く、やりたい事 を容易に実現できます。一方で、GUIが作りにくい、細かな調整がしにくい等の デメリットも存在しますが、初学者が取り組むには最適な言語と考えます。 2..1. Python

現在世界で最も**需要が高い言語**。非常に簡単で読みやすい上、守備範囲が広いため 人気の言語です。本研究室でも**機械学習**や IoT 開発を行う際に使用しています。 また、Web アプリの開発や統計分析(データサイエンス)など、<u>様々な分野に活躍の</u> <u>場があります</u>。世界的な実例としては、「YouTube」や「Evernote」、「Instagram」等 のホームページ開発にも用いられているらしいです。個人的には、<u>初心者が最初に</u> **学ぶ言語には最適**だと考えます。

2...2. Ruby

日本人が開発した**日本で人気の高い言語**。読みやすく、書きやすく、学びやすい言語 を目指して開発されているだけあって、非常に扱いやすい言語です。Python よりも 更に簡単に記述できる言語で、現在の主な守備範囲は<u>Web 関連</u>です。「クックパッド」 や「Hulu」のホームページは Ruby で開発されています。もちろん、Web 以外にも 応用が利く言語ではありますが、現在の主流から考えると、Web 関連の<u>IT 企業への</u> 就職を考える場合などには良い言語だと思います。 2...3. JavaScript

主に Web ページに動きを付けるために使用されている言語。C 言語に似た文法構造 を持ちながら、容易に書くことが可能な言語です。Ruby と同様に Web 関連に強い という特徴があります。現状では Ruby 以上に Web 特化の言語として扱われている ため、他の言語を身に着けた上で、Web 開発に興味を持った際に取り組むと良い かもしれません。

▶ コンパイラ言語編

細々とした設定が必要で面倒ですが、処理速度が格段に速く、あらゆる作業ができる 万能な言語が多いです。初学者には難易度が高めですが、メーカーで開発職を望む 場合には必須スキルなのでそのうち勉強する必要が出てくると思います。

②..1. C / C++

あらゆる言語のベースとなっている言語。**難度が高い**代わりに、すべてが得意分野と 言えるほど応用のしやすさに優れた言語です。企業のインターンシップでは「C 言語 もしくは C++言語が使用できること」という条件が設定されたテーマが非常に多く あり、<u>大企業のインターンシップに有利</u>になります。日頃から <u>C++を学んでおく事が</u> 工学部の学生と対等に渡り合うための唯一の方法と言えます。

大手電機メーカーに就職する際には習得が必須の言語だと考えます。

2...2. Java / kotlin / Scala

世界中で人気の高い言語。C++よりも難易度が低いにもかかわらず、同等の仕事が こなせる優秀な言語です。現在では特に<u>スマートフォン向けアプリの開発</u>で大きな 需要があります。また互換言語である kotlin はスマホアプリで、Scala は科学解析や サーバー関連等で非常に需要が高まっており、応用の幅が広がってきています。 IT 企業の開発への就職やフリーランスのエンジニアとして働いきたいと考えている場 合には C++よりも学習する価値のある言語だと考えます。

2...3. C#

ゲーム開発に特化した言語。Microsoft 社が開発した C++の改良版です。C++と比較 して動作が遅く企業での使用率は低いですが、<u>ゲーム開発に関して言えば、現在最強</u> <u>の言語</u>です。Unity という世界中のあらゆるゲームで使用されている開発エンジンは この C#で動作します。また、GUI が容易に作成できるという特徴もあるため、一応、 GUI 開発の需要も存在します。<u>ソフトウェアデザインやゲーム開発に興味があれば</u> 学んでも良いかもしれません。

ちなみに、私が3年間メインで使用した言語でもあります。

第3章 ~プログラミングの学び方~

1. プログラミング学習の進め方

英語の学習をする際には、バランスが重要だと思います。ひたすら単語の暗記を しても英文を読めるようにはなりませんし、文法だけを理解しても単語を知らなければ やはり英文は読めません。また、英文が読めても、声に出す練習をしなければ英会話が できるようにはならないと思います。

プログラミング言語も「言語」と呼ばれているだけあって、学習の基本は英語の 勉強法と同じだと考えています。練習を繰り返して、単語や文法を学習し、自力で作文 できるようになっていきます。

その練習方法としておすすめなのが<u>ホームページを参照しながら写経</u>(ページに 記載されているプログラムをひたすら書き写す事)する事です。これは英語学習で言う ところの音読のような効果があるので、プログラミング言語特有の単語や文法も頭に 入れつつ、プログラミングの流れというものが学べます。写経を通してプログラムを 書くことに慣れてきたら次は<u>参考ホームページに載っている練習問題や自分の欲しい</u> <u>ソフトウェアを実装</u>(実際に作成すること)してみるのが良いと思います。この実装は 英語学習で言うところの英作文のようなもので、自力で記述する能力を磨けます。

言語の学習はとにかく練習あるのみだと思います。毎日小さな目標を立てて、 コツコツと鍛錬を重ねてください。

- 2. プログラミング入門におすすめのホームページ
- ▶ python の入門
 - ① Paiza ラーニング https://paiza.jp/works

会員登録が必要ですが、非常にわかりやすく動画で様々な言語の解説をして

くれるサイトです。有料の講座も存在しますが、python は無料です。

② Let's プログラミング Python 入門 https://www.javadrive.jp/python/

①と違って分かりにくいですが、網羅的にまとまっているため、①だけでは

よくわからなかったところを復習する際などに役立つと思います。

③ サンプル集 https://algorithm.joho.info/programming/python/sample-code-py/

Python のサンプルや様々な応用方法がまとめられているため、研究の際に

「画像処理ってどうやるの?」とか「機械学習って…(略)」とかを調べるの

に役立ちます。

▶ C++の入門

①ロベールの C++教室 http://www7b.biglobe.ne.jp/~robe/cpphtml/

C++の基本から応用まですべて網羅されたホームページです。内容は非常 に良いのですが、ボリュームがありすぎます。全部真面目に取り組もうとすると たぶん食中りになってしまうので「飽きたら途中で投げ出しても良い」くらいの 気持ちで取り組むことをお勧めします。

第4章 ~プログラミング環境構築~

1. python のインストール

「Anaconda」という Python 言語をインストールするためのツールがあります。

この Anaconda を利用すると簡単に python 言語をインストールしてくれます。

また、軽量版の「Miniconda」もお勧めです。自分の PC のスペックに応じて好きな方

を利用してください。(基本的には Anaconda で大丈夫です)

<Anaconda による python のインストール手順>

- ① まず、Anaconda 公式 HP を開きます。
- ② 次に、画面中段の Windows というボタンをクリックします。

		$\square \backslash \square$	
spyder Nu DASK	umPy Ss Sokeh	iciPy Numba Numba	er
econ H	20.ai Tensor	Flow	
	Linux	Linux	Linux

③ 「Anaconda 2018.12 for Windows Installer」という画面に切り替わったら、

Python 3.7 version の下の Download をクリックします。

🕊 Windows 🛛 🗯 n	nacOS 🛛 🔬 Linux
Anaconda 2018.12 for	Windows Installer
Python 3.7 version	Python 2.7 version
Download	Download
64-Bit Graphical Installer (614.3 MB)	64-Bit Graphical Installer (560.6 MB)
32-Bit Graphical Installer (509.7 MB)	32-Bit Graphical Installer (458.6 MB)

④ ダウンロードした exe ファイル (Anaconda3-2018.12-Windows-x86_64.exe) を

ダブルクリックで起動します。

⑤ 以下のような「Welcome to~」画面が表示されたら Next をクリックして次へ。



⑥ I Agree をクリックして次へ。

O Anacond	la3 2018.12 (64-bi	t) Setup				8
		License Agree	nent			
	IACONDA	Please review the 2018.12 (64-bit)	ne license terms l).	before installing	Anaconda3	
Press Pag	e Down to see the	rest of the agree	ment.			
Anacond	a End User License	Agreement				^
Copyrigh	t 2015, Anaconda,	Inc.				
All rights	reserved under th	e 3-dause BSD Lic	ense:			
Redistrib permitter	ution and use in so d provided that the	urce and binary for following condition	orms, with or wit	hout modificatio	n, are	
I						¥
If you acc agreemer	cept the terms of terms of the terms of terms of the terms of terms o	he agreement, die da3 2018.12 (64-	:k I Agree to con bit).	itinue. You must	accept the	
Anaconda, II	nc,					
			< Back	I Agree	Canc	el

⑦ そのまま Next をクリックして次へ。

O Anaconda3 2018.12 (64-b	it) Setup	
	Select Installation Type	
	Please select the type of installation you Anaconda3 2018.12 (64-bit).	would like to perform for
 Just Me (recommended) 		
All Osers (requires admir	(privileges)	
Anaconda, Inc. ————		
	< Back Nex	t > Cancel

⑧ そのまま Next をクリックして次へ。

O Anaconda3 2018.	12 (64-bit) \$	Setup				
	Ch	oose Install	Location			
	NDA c	hoose the fold	der in which	to install /	Anaconda	a3 2018.12 (64-bit).
Setup will install Ar folder, click Browse	naconda3 20 e and select a	18. 12 (64-bit) another folder	in the follo . Click Next	wing folde t to continu	r. To inst ue.	all in a different
Destination Fold	er					Prowee
Space required: 2. Space available: 6	8GB 2.2GB					biowsciii
Anaconda, Inc. ——			< Ba	ick	Next >	Cancel

⑨ 「Add Anaconda to~」にもチェックを付けてから Install をクリック。

O Anaconda3 2018.12 (64-	bit) Setup	_		\times
	Advanced Installation Options			
	Customize how Anaconda integrates with W	indows		
Advanced Options				
Add Anaconda to	my PATH environment variable			
Not recommended. I menu and select "An Anaconda get found cause problems requ	nstead, open Anaconda with the Windows Star aconda (64-bit)". This "add to PATH" option mal before previously installed software, but may iring you to uninstall and reinstall Anaconda.	t kes		
Register Anacono	a as my default Python 3.7			
This will allow other p PyCharm, Wing IDE, detect Anaconda as	programs, such as Python Tools for Visual Studie PyDev, and MSI binary packages, to automatic the primary Python 3.7 on the system.	o cally		
Anaconda, Inc. —	< Back Install		Cano	el

⑩ インストールが完了したら Next をクリックして次へ。

O Anaconda3 2018.12 (64-1	pit) Setup	
	Installation Complete Setup was completed successfully.	
Completed		
Show details		
Anaconda, Inc. ————		
	< Back	Next > Cancel

① 以下の画面が出たら Skip をクリックして次へ。

O Anaconda3 2018.12 (64-1	pit) Setup	_		×
	Anaconda3 2018.12 (64-bit) Microsoft Visual Studio Code Installati	on		
Anaconda has partnered Code is a free, open sou support for Python code more. To install Visual Studio C connectivity.	d with Microsoft to bring you Visual Studi irce, streamlined cross-platform code ed editing, IntelliSense, debugging, linting ode, you will need Administrator Privileg	io Code. Visua ditor with exco , version con les and Interr	al Studio ellent trol, and net	
Visual Studio Code Licen	se Install Microsoft VSCode			
Anaconda, Inc. ————	< <u>B</u> ack	Skip	Can	cel

12 2つのチェックを外して Finish をクリックで完了。

O Anaconda3 2018.12 (64-bit)	Setup	—		\times
	Thanks for installing Ar	nacond	a3!	
O ANACONDA.	Anaconda is the most popular Python data science platform. Share your notebooks, packages, projects and environments on Anaconda Cloud!			
	Learn how to get started with An	<u>aconda;</u>		
	< <u>B</u> ack	inish	Cano	el

13 続いてインストールに成功しているかを確認するために、スタートボタンを押して

スタートメニューを開き、Windows システムツールからコマンドプロンプトを起動

し、以下を1行コピペして Enter を押します。

conda list

🔤 בער ארב דע

Microsoft Windows [Ver (c) 2018 Microsoft Cor	ion 10.0 oration.	.1718 All	84.590] rights	reserved.	
C:¥Users¥K>conda list					

その後、文字がずらずらっと表示されたらインストール成功です。

確認後はこの画面を閉じてください。

<Miniconda による python のインストール手順>

① まず、Miniconda 公式 HP からインストーラーをダウンロードします。

Miniconda

	Windows	Mac OS X	Linux
Duther 2.7	64-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer)
Python 3.7	32-bit (exe installer)	64-bit (.pkg installer)	32-bit (bash installer)
Duthon 2.7	64-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer)
	32-bit (exe installer)	64-bit (.pkg installer)	32-bit (bash installer)

② ダウンロードした exe ファイル (Miniconda3-latest-Windows-x86_64.exe) をダブル

クリックで起動します。

③ 以下のような「Welcome to~」画面が表示されたら Next をクリックして次へ。



④ I Agree をクリックして次へ。

O Miniconda3 4.5.12 (64-bi	it) Setup	8
	License Agreement	
	Please review the license terms before installing Miniconda3 4.5.12 (64-bit).	
Press Page Down to see th	e rest of the agreement.	
Miniconda End User Licens	e Agreement	^
Copyright 2015, Anaconda	a, Inc.	
All rights reserved under t	he 3-dause BSD License:	
Redistribution and use in s permitted provided that th	ource and binary forms, with or without modification, are e following conditions are met:	
		¥
If you accept the terms of agreement to install Minico	the agreement, click I Agree to continue. You must accept the nda3 4.5.12 (64-bit).	
Anaconda, Inc		
	< Back I Agree Cano	cel

⑤ そのまま Next をクリックして次へ。

O Miniconda3 4.5.12 (64-b	it) Setup	
	Select Installation Type	
	Please select the type of installation you Miniconda3 4.5.12 (64-bit).	would like to perform for
Just Me (recommended))	
◯ All Users (requires adm	in privileges)	
Anaconda, Inc. ———		
	< Back Nex	t > Cancel

⑥ そのまま Next をクリックして次へ。

O Miniconda3 4.5.12 (64-b	it) Setup	
	Choose Install Location	
	Choose the folder in which to install	Miniconda3 4.5.12 (64-bit).
Setup will install Miniconda folder, dick Browse and se	3 4.5.12 (64-bit) in the following folder elect another folder. Click Next to contir	. To install in a different nue.
Destination Folder	-3	Browse
Space required: 250.6MB Space available: 61.6GB		bioliscin

⑦ 「Add Anaconda to~」にもチェックを付けてから Install をクリック。

O Miniconda3 4.5.12 (64-bi	t) Setup		_		\times
	Advanced Installati	ion Option	5		
	Customize how Anac	conda integr	rates with Windows		
Advanced Options					
Add Anaconda to	my PATH environment	variable			
Not recommended. Ir menu and select "Ana Anaconda get found cause problems requi	nstead, open Anacond aconda (64-bit)". This " before previously insta ring you to uninstall an	a with the W add to PATH alled softwar ad reinstall A	/indows Start H" option makes re, but may maconda.		
Register Anacond	a as my default Pythor	n 3.7			
This will allow other p PyCharm, Wing IDE, detect Anaconda as	rograms, such as Pyth PyDev, and MSI binary the primary Python 3.7	on Tools for / packages, ⁷ on the sys	Visual Studio to automatically tem.		
Anaconda, Inc					
		< <u>B</u> ack	Install	Cance	el

⑧ インストールが完了したら Next をクリックして次へ。

O Miniconda3 4.5.12 (64-b	it) Setup	
	Installation Complete Setup was completed successfully.	
Completed		
Show details		
Anaconda, Inc. ————	< Back	Next > Cancel

⑨ 2つのチェックを外して Finish をクリックで完了。



 ⑩ 続いてインストールに成功しているかを確認するために、スタートボタンを押して スタートメニューを開き、Windows システムツールからコマンドプロンプトを起動 し、以下を1行コピペして Enter を押します。

conda install pip

■ コマンド プロンプト	_	
Microsoft Windows [Version 10.0.17134.11: (c) 2018 Microsoft Corporation. All righ	2] ts	reserved.
C:¥Users¥katoh7700k>conda install pip		

Proceed ([y]/n)? というような 1 文が表示されたら y キーを押した後に Enter を押

してください。さらに、動作が完了したら同様に以下をコピペして Enter を押します。

pip install virtualenv

C:¥Users¥katoh7700k>pip install virtualenv

この段階でエラーが出なければインストール完了です。

2. 仮想環境のすすめ(不要な場合、よくわからないという場合は3へ)

(参考ホームページ: http://www.bokupy.com/detail/4)

仮想環境とは、コンピュータ上に仮想的に作成するプログラミング環境のことです。 Python は現在積極的に開発が進められているため、頻繁にバージョンアップされます。 バージョンが新しくなることは良いことですが、それまで動作していたプログラムが 動作しなくなるというようなエラーが起こることもあります。つまり、いつでも最新の バージョンが良いというわけではありません。また、環境構築に際して何かミスが発覚 した場合、仮想環境を削除するだけで過去のミスを無かった事にできます。

(仮想環境を使用しない場合はインストールした Anaconda または Miniconda 本体を アンインストールして、python 言語を利用する環境を再構築する必要があります。) また、自分で python 言語を実行する環境を作成することによって、どのフォルダに何 があるかというような構造を確実に把握できるようになるというメリットもあります。

というわけで、仮想環境を構築するという面倒くささもあると思いますが、一度 構築してしまえば便利なこともあるので、ぜひ挑戦してみて下さい。 <仮想環境の構築手順>

① 仮想構築するために以下のコマンドを実行します。

conda create -n v_env numpy scipy pandas pillow matplotlib opencv

※conda create -n v_env で仮想環境の作成を行っています。この「v_env」は仮想環境の

名称で、自分の好きな名称に任意に変更が可能です。また、それ以降に続く内容は仮想

環境にインストールしたいライブラリです。後で追加することも可能ですが、ここで

指定しておくと自動でインストールしてくれるので楽になります



Proceed ([y]/n)?と表示されるので、y と入力してから Enter を押すとインストールが

開始されます。下のようになったら、インストール完了です。



② 仮想環境の作成が成功したかを確認するために以下のコマンドを実行します。

activate v_env



- ③ 以下、よく使うコマンドです。
 - ・仮想環境の起動: activate 仮想環境名
 - ・インストール済みライブラリの確認:conda list
 - ・ライブラリのインストール: conda install ライブラリ名
 - ・ライブラリのアンインストール: conda uninstall ライブラリ名
 - ・ライブラリのアップデート: conda update ライブラリ名
 - ・仮想環境の終了:deactivate
- ④ 必要に応じて以下で使用方法を確認してください。

https://qiita.com/ozaki_physics/items/985188feb92570e5b82d

3. Visual Studio Code のインストールと設定

Anaconda または Miniconda で python を導入しただけでもプログラミングは可能 ですが、操作性が悪いので、Visual Studio Code というテキストエディタ(超高性能な メモ帳のようなソフトウェア)を導入します。

<Visual Studio Code のインストール>

① とりあえず Anaconda (または、Miniconda) をインストールしてください。

(この作業は1で完了しているはずです。)

- ② 今後の作業のために、事前にCドライブに「pytest」のような作業用のフォルダを作成 します(フォルダの名称は好きに決めて構いません)。
- ③ Visual Studio Code (以下、VScode)の公式ホームページ開き、Windows 版の VScode

のインストールファイルをダウンロードします。

Download Visual Studio Code

Free and open source. Integrated Git, debugging and extensions.



④ ダウンロードした exe ファイル (VSCodeUserSetup-x64-1.32.1.exe) をダブルクリック

で起動します。

⑤ セットアップウィザードの開始画面が開いたら、次へ(N)。

🗙 Visual Studio Code セットアッ	
	Visual Studio Code セットアップウィザード の開始
	このプログラムはご使用のコンピューターへ Microsoft Visual Studio Code (User) をインストールします。
	続行する前に他のアプリケーションをすべて終了してください。
X	続行する(こは「〉次へ」、セットアップを終了する(こは「キャンセル」をク リックしてください。
	次へ(N) > キャンセル

⑥ 使用許諾契約書の同意画面が開いたら「同意する」にチェックして次へ(N)。

🗙 Visual Studio Code セットアップ 📃	
使用許諾契約書の同意 続行する前に以下の重要な情報をお読みください。	N
以下の使用許諾契約書をお読みください。インストールを続行するにはこの契約書 する必要があります。 	:(こ同意
本ライセンスは Visual Studio Code 製品に適用されます。Visua Studio Code のソース コード	al î
は、 <u>https://github.com/Microsoft/vscode/blob/master/license.</u> に記載された MIT ライセンス契約に基づ き、 <u>https://github.com/Microsoft/vscode</u> で閲覧することができ	<u>txt</u> ≢
す。その他のライセンス情報 は、 <u>https://code.visualstudio.com/docs/supporting/faq</u> のFA	Q J
 同意する(A) ○ 同意しない(D) 	
< 戻る(B) 次へ(N) >	キャンセル

⑦ 次へ(N)



⑧ 次へ(N)

🗙 Visual Studio Code セットアップ	
プログラムグループの指定 プログラムアイコンを作成する場所を指定してください。	×
ゼットアップはスタートメニューにプログラムのショートカットを作成しま	ŧす。
続けるには「次へ」をクリックしてください。違うディレクトリを選択するには「参照 ください。	照」をクリックして
Visual Studio Code	参照(R)
□ プロガラムガル~づを作成しな(1/D)	
< 戻る(B) 次へ(N) >	キャンセル

⑨ 次へ(N)

🗙 Visual Studio Code セットアップ	
追加タスクの選択 実行する追加タスクを選択してください。	×
Visual Studio Code インストール時に実行する追加タスクを選択して、「次へ」 ださい。	」をクリックしてく
アイコンを追加する:	
□ デスクトップ上にアイコンを作成する(D)	
その他:	
🔲 エクスプローラーのファイル コンテキスト メニュー(こ [Code で開く] アクション	/を追加する
ロ エクスプローラーのディレクトリ コンテキスト メニューに [Code で開く] アクシ る	ヨンを追加す
🗌 サポートされているファイルの種類のエディターとして、 Code を登録する	
✓ PATH への追加(再起動後に使用可能)	
< 戻る(B) 次へ(N) >	キャンセル

⑩ インストール(I)を押すと、インストールが開始されます。

X Visual Studio Code セットアップ	
インストール準備完了 ご使用のコンピュータへ Visual Studio Code をインストールする準備ができました。	X
インストールを続行するには「インストール」を、設定の確認や変更を行うには「戻る」 クしてください。	をクリッ
インストール先: C:¥Users¥K¥AppData¥Local¥Programs¥Microsoft VS Code	^
プログラムグループ: Visual Studio Code	
追加タスクー覧: その他: PATH への追加(再起動後に使用可能)	
<	>
< 戻る(B) インストール(I) :	キャンセル

⑪ この画面が表示されたらインストール完了です。

🗙 Visual Studio Code セットアッ	
	Visual Studio Code セットアップウィザード の完了
	ご使用のコンピューターに Visual Studio Code がセットアップされま した。 アプリケーションを実行するにはインストールされたアイコンを 選択してください。
	セットアップを終了するには「完了」をクリックしてください。
X	☑ Visual Studio Code を実行する
	完了(F)

⑫ 完了ボタンを押し、VScodeの起動を確認したら、一旦 PC を再起動してください。

<Visual Studio Code の設定>

- ① プログラムメニューから Visual Studio Code を選択し、起動します。
- ② 現状の VScode ではまだ python を使用できないので、以下では python を使用できる

ように設定を変更していきます。

③ まず、画面左側にある四角いボタン(拡張機能タブボタン)を押します。



④ 拡張機能タブが開かれると、次のような画面が表示されるので、上部の検索バーに

python と入力します。



⑤ すると python に関係する拡張機能がたくさん表示されますが、今回は一番上の

☆マークがついている「Python」という部分をクリックしてください。

⑥ すると、次のような画面に切り替わります。画面が切り替わったら、Install ボタンを

クリックします。



↓のように、Install ボタンが ✓ Installed に変わったらインストール完了です。

ᆀ <u>F</u> i	le <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>D</u> ebug <u>T</u> ermin	nal <u>H</u> elp	Extension: Python - Visual Studio Code	>	\times
D	EXTENSIONS: MARKETPLACE	\equiv Extension: Python \times			
0	python		Python ms-python.python		
Ŷ	Python 2019.2.5558 Linting, Debugging (multi-threaded, remote), Microsoft √ Installed ☆		Microsoft $ \phi$ 33,976,412 $ \star \star \star \star \star $ Rep Linting, Debugging (multi-threaded, remote), Intellisense, co	ository License	2
8	AREPL for python 1.0.11 real-time python scratchpad Almenon Install		✓ Installed Uninstall This extension is recommended based on the files you recently op	ened.	
¢	Python Preview 0.0.4 Provide Preview for Python Execution. dongli Install	<u>Details</u> Contributio	Ignore Recommendation		
	Python Extension Pack 1.4.0 Popular Visual Studio Code extensions for Pyt				
\$	Don Jayamanne Install Python for VSCode 0.23	Python exte	nsion for Visual Studio Code		
⊗04	0			🙂 u	

⑦ 続いて、VS Code の上部メニューの File をクリックし、 Open Folder...を選択、

先ほど作成した作業フォルダの「C:¥pytest」を開きます。すると、以下のような画面に

- I Eile Ed Selection View Go Debug Terminal Help Welcome - pytest - Visual Studio Code EXP ORER ···· 🖽 ᆀ Welcome 🛛 🗙 D OP N EDITORS ¥ 41 Welcome Q ▲ PYTEST Customize Y Start Tools and languages 8 Install support for JavaScript, TypeScript, Pytho ... Ē Settings and keybindings Install the settings and keyboard shortcuts of V... Recent Color theme ch03 c:\deep-learning-from-scratch-master Make the editor and your code look the way yo ... ch02 c\deep-learning-from-scratch-master Learn Find and run all commands Help Rapidly access and search commands from the ... Interface overview Get a visual overlay highlighting the major com... Interactive playground Try essential editor features out in a short walkt... Show welcome page on startup æ • OUTLINE 80▲0 0 🌲
- なり、ココを見ると pytest フォルダが開かれていると確認できます。

⑧ 下図の赤丸部分をクリックして test.py と入力します。



⑨ 以下のように画面が変化します。右下に何か表示されていると思いますので、

とりあえず、Install をクリックしてください。

🧐 I	Eile <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>D</u> ebug <u>T</u> e	rminal <u>H</u> elp	test.p	ay - pytest - Visual Studio Code			
n	EXPLORER	ᆀ Welcome	🔹 test.py	×			
A courted	OPEN EDITORS	1					
Ω	ᆀ Welcome						
	🗙 🚽 test.py						
88	▲ PYTEST						
8	🔮 test.py						
œ							
LE1				The 'Python' extension is recommended for this file	tvpe.	t	×
					-olber		
*				Install Sho	w Recomme	endation	ns
**	♦ OUTLINE						
⊗04	0			Ln 1, Col 1 Spaces: 4 UTF-8 CRLF	Python	•	1

⑩ Install クリック後、この画面が表示されたら、ユコをクリックして元の画面に戻ります。



① (仮想環境を使用しない場合は⑬へ)

以前「仮想環境のすすめ」で仮想環境を作成した場合はここで、左下のココをクリック

して自分で作成した仮想環境の名称を選択してください(最初は、~('base': conda)と なっているので、これを~('v env': conda)などに切り替えます)。 12 仮想環境を選択した場合には、このあたりで、pylint というライブラリをインストール

しろと言われるので、素直に Install をクリックする。

1	ile <u>E</u> dit <u>S</u> election ⊻iew <u>G</u> o <u>D</u> ebug <u>T</u> ermi	inal <u>H</u> elp	test.p	y - pytest - Visual Studio Code			
A	EXPLORER	Welcome	🔹 test.py	×			
Absend	▲ OPEN EDITORS						
Ω	ᆀ Welcome						
~	🗙 🍖 test.py						
88	▲ PYTEST						
s	▶ .vscode						
6	🍨 test.py						
S							
63							
Ľ				🙁 Linter pylint is not installed.		ф	×
					-		
*				Source: Python (Extension) Install Select Linter	Do not s	how agai	n
	▶ OUTLINE						-
Pytho	n 3.7.2 64-bit ('v_env': conda) 🛛 😣 0 🛕 0			Ln 1, Col 1 Spaces: 4 UTF-8 CRLF	Python	0 1	11

続いて、以下のような表示が出てくるので、Install using Conda をクリックする。

🥑 E	ile <u>E</u> dit <u>S</u> election <u>V</u> iew	<u>G</u> o <u>D</u> ebug <u>T</u> ermir	al <u>H</u> elp	test.py - pytest - Visual Studio Code			— C	ב	×
രീ	EXPLORER	Select an option to ins	all pylint						••••
	▲ OPEN EDITORS	Install using Conda							
Ω	ᆀ Welcome	install using Pip							
-	🗙 🍨 test.py								
83	▲ PYTEST								
8	.vscode								
A	🍦 test.py								
(S)									
-									
*									
*	▶ OUTLINE								
Pytho	n 3.7.2 64-bit ('v_env': conda)	⊗ 0 ∧ 0		Ln 1, Col 1	Spaces: 4 U	JTF-8 CRLF	Python	•	4

すると、自動的にインストールが始まり、Proceed ([y]/n)?と表示されるので、y と入力

してから Enter を押すとインストールできます。

<u>ৰ</u> ৷	ile <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>D</u> ebug <u>T</u> ermir	al <u>H</u> elp		• test.py -	pytest - Visual Stud	dio Code						
r h	EXPLORER	ᆀ Welcome	🌵 tes	t.py 🔹							m	
	OPEN EDITORS 1 UNSAVED											
Ω	刘 Welcome											
/-	🛛 🚭 test.py											
89	▲ PYTEST	PROBLEMS	OUTPUT TE	RMINAL		1: Pythor	۱		+	m 🛍	^	×
8	Ivscode											
	🗇 test.py	astroi	d:	2.1.0-py	37 0							
S		colora	ma:	0.4.1-py	37_0							
		isort:	bjact provu	4.3.8-py	37_0							
Ē.		mccabe	:	0.6.1-py	37 1							
		pylint		2.2.2-py	37_0							
ىقە		wrapt:		1.11.1-p	y37he774522_0							
*	▶ OUTLINE	Proceed ([v]/n)? v									
Pytho	n 3.7.2 64-bit ('v_env': conda) 🛛 🗴 0 🛕 0					Ln 1, Col 1	Spaces: 4	UTF-8	CRLF	Python	٢	٠

- ③ これでひとまず VScode で python を使えるようになりました。
- ⑭ 続いて、デバッグモードの設定を行います。

先ほど作成したテストファイル(test.py)に以下の記述を追加してください。

import numpy as np

print ("numpy version:" + np.version.full_version)



(b) 虫が描かれたボタン(デバッグタブボタン)をクリックして、デバッグ画面を開きます。



¹⁶ 歯車のボタンをクリックすると、「launch.json」というファイルが生成されます。

I	ile <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>D</u> ebug <u>T</u> ermi	al <u>H</u> elp • test.py - pytest - Visual Studio Code	- 🗆 X
ß	DEBUG No Configurations	Welcome test.py test.py finder numpy as np	
Q		<pre>2 print ("numpy version:" + np.version.full_v 3</pre>	ersion)
¥			
8	▲ WATCH		
\$			
Pytho	a 3.7.2 64-bit ('v_env': conda) 🛛 😵 0 🛕 0	Ln 3, Col 1	Spaces: 4 UTF-8 CRLF Python 😌 🌲

「launch.json」は python の実行に関する設定ファイルで、自分で調節することも可能

ですが、基本的にはそのままで大丈夫です。

① 「launch.json」が精製されたら、緑色の▽を押せばデバッグできるようになります。

🧐 <u>F</u>	ile <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>D</u> ebug <u>T</u> ermi	nal <u>H</u> elp • test.py - pytest - Visual Studio Code	—	o x
b	DEBUC Python: Current File 🔻 🅸 🗈	Welcome test.py () launch.json		•••
Q	VARIABLES	<pre>2 print ("numpy version:" + np.version.full_version) 3 </pre>		
¥				
8	A WATCH			
	A WAICH			
Pythor	n 3.7.2 64-bit ('v_env': conda) 🛛 8 0 🛕 0	Ln 3, Col 1 Spaces: 4 UTF-8 CRL	F Pytho	on 🙂 🔔

18 今回の例では以下のように「numpy version:1.16.2」のように表示されれば成功です。

<u> 1</u>	ile <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>D</u> ebug <u>T</u> ern	inal <u>H</u> elp test.py - pytest - Visua	Studio Code	- 0	
F	DEBUG 🕨 Python: Current File 🔻 🅸 🗈	✓ Welcome	h.json		□ …
	✓ VARIABLES	1 import numpy as np	ncion full vencion)		
ρ		3	rsion.tuil_version)		
¥		PROBLEMS OUTPUT TERMINAL ···	1: Python Debug Consc 🔻 🕂		^ ×
8	▲ WATCH	<pre>P5 C:\pytest> cd 'c:\pytest'; \${env:PYTH ; & 'C:\Users\K\Anaconda3\envs\v_env\pytl</pre>	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	NUNBUFFERED sions\ms-py)}='1' /thon.
Ē		python-2019.2.5558\pythonFiles\ptvsd_lau st~port~50162 numpy version:1.162. rs_6:verset.cs//k/Anaconda3/Scrim	<pre>scher.py' 'default' 'client' ' scativate</pre>		icalho
\$		PS C:\pytest> conda activate v_env PS C:\pytest>	S Strate		
Pytho	n 3.7.2 64-bit ('v_env': conda) 🛛 🏵 0 🛕 0 🕨 Python: (urrent File (Integrated Terminal) (pytest)	Ln 3, Col 1 Spaces: 4 UTF-8 CRL	F Python	🙂 🌲

① ここまではデバッグモードの使用法を説明しましたが、デバッグではなく、通常実行する場合には、test.pyのプログラム内で右クリックします。すると、メニューの中に「ターミナルで Python ファイルを実行」という項目があります。これを選択すると、



これを選択すると、以下のように通常実行されます。

🧐 <u>F</u>	ile <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>D</u> ebug <u>T</u> ermi	nal <u>H</u> elp test.py - pytest - Visual Studio Code	- 🗆 X
Ŋ	EXPLORER	3 Welcome	
ρ	✓ OPEN EDITORS ≪] Welcome × ♥ test.py	<pre>print ("numpy version:" + np.version.full_version) 3</pre>	
¥	() launch.json .vscode A PYTEST	PROBLEMS OUTPUT TERMINAL ··· 2: Python T	□ <u> </u> ∧ ×
8	 ▶ .vscode est.py 	Windows PowerShell Copyright (C) Microsoft Corporation. All rights reserved.	
		<pre>PS C:\pytest> C:/Users/K/Anaconda3/Scripts/activate PS C:\pytest> conda activate v_env PS C:\pytest> & C:/Users/K/Anaconda3/envs/v_env/python.exe c:/pytest/test.py numpy version:1.16.2 PS C:\pytest></pre>	y
*	▶ OUTLINE		
Pytho	n 3.7.2 64-bit ('v_env': conda) 🛛 🛛 🗛 0 🌓 Python: Cu	rrent File (Integrated Terminal) (pytest) Ln 3, Col 1 Spaces: 4 UTF-8 CRLF	Python 🙂 🐥

必要に応じてデバッグモードと使い分けてください。

<Visual Studio Code を使いやすくする>

① コードの自動整形機能を有効にする

・File から Preferences > Settings とクリックしていき、設定画面を開きます。



・次に、Search settings に FormatOn と入力します。



・「Format On Paste」や「Format On Save」などにチェックを付けてこの設定画面を

閉じると、勝手に見やすいコードに整形してくれるようになります。



・なお、初めて整形を行う際に「autopep8」がないといわれる可能性があります。

<u>ع</u> ال	ile <u>E</u> dit <u>S</u> election <u>V</u> iew <u>G</u> o <u>D</u> ebug <u>T</u> ermi	nal <u>H</u> elp	test.py - pytest - Visual Studio Code	- 🗆 ×
ı.	EXPLORER	🔹 test.py 🗙 👔	launch.json	···· ···
	▲ OPEN EDITORS	1 import num	py as np	
Ω	🗙 🅏 test.py	2 print ("nu	<pre>mpy version:" + np.version.full_version)</pre>	
~	() launch.json .vscode	3		
89	▲ PYTEST			
X	▶ .vscode			
~	🕏 test.py			
8				
			😵 Formatter autopep8 is not installed. Install?	¢ ×
4			Source: Python (Extension)	e black Use yapf
	▶ OUTLINE			
Pytho	n 3.7.2 64-bit ('v_env': conda) 🛛 😢 0 🛕 0		Ln 3, Col 1 Spaces: 4 UTF-8 CRL	F Python 🙂 🐥 2

この時は素直に Yes を選択し、Install using Conda をクリックして pylint のときと同様

にインストールしてください。

② 便利な拡張機能をインストールする

ここまで設定すると快適にコーディングできるようになっていますが、さらに快適に するために、おすすめの拡張機能を以下にまとめました。

- Japanese Language Pack:日本語化できる
- vscode-icons:アイコンが見やすくなる

ただ Install ボタンをクリックするだけです。

● Bracket Pair Colorizer:かっこを対応ごとに色分けしてくれて見やすい

これらは拡張機能タブを開き Search Extensions から検索すればインストールできます。 なお、インストールの仕方は最初に Python 拡張機能をインストールした際と同様で、

③ さらに VScode の設定を深めたいという場合には https://wemo.tech/1756 参照。

第5章 ~困ったときの解決法~

1. 一番大事なことは「検索力」

検索力とは、自分が欲しい知識や技術を Google で検索する能力のことです。企業や フリーランスで活躍されているプロの方々がどうしているのかは知りませんが、勉強を 始めたばかりの学生が自分の知識だけで解決できる問題はほとんどないです。

その一方で、初心者が困ったときに知りたい事の 90%は Google に答え(もしくは ヒント)が書いてあります。つまり、初心者の知識・技術不足を補うために重要なのが 「どれだけ正確に検索できるか」なのです。

2. 検索のコツ

自分ではあまり意識せずに検索していたので、「コツ」というようなものを考えた事 はありませんでした。「まぁ、たぶん誰かが解説してくれているだろう」と考えて検索 したところ、以下のようなホームページが見つかりました。現役のエンジニアが書いた だけあってわかりやすくまとまっているので参考にしてください。

□ エンジニアに学ぶ検索テクニック

https://www.sejuku.net/blog/879

□ 新人エンジニアに教えたい Google 検索のコツ

https://qiita.com/sta/items/7f600ac5497f81173fc7

3. 本当に困ったら素直に「質問」する

検索で見つかった解決策を理解しないまま流用すると、結局新しい知識や技術が 身に付かず、成長しません。それっぽいものは見つかったけど、自分だけではよく理解 できないという場合には、誰かに相談してできる限り理解できるように努めましょう。

また、Google で検索しても答えやヒントが見つからない場合もあります。そんな時には、先生や先輩に質問したり、質問サイトで質問したりしましょう。

ただし、検索しても質問してもどうしてもわからないことはあると思います。 そのような場合には、上記の「理解しよう」という話と矛盾してしまいますが、 「そのうち理解できる日が来るさ」と気楽に捉えて次のステップに進んでしまうことを お勧めします。

第6章 ~今後のステップ~

結局のところ、プログラミングに関する基本的な知識を身につけないと先に進むことは 難しいです。そのため、まずは第3章でまとめたおすすめのホームページ等を利用して基礎 知識を身に着けてほしいと考えています。

しかし、基礎ばかり勉強していても何に応用できるのかが分からず、勉強に集中できない という人も多いと思います。そこで、本研究室で扱う「ディープラーニング」の基礎に触れ られる比較的簡単なプログラムを紹介してくれているホームページをまとめました。

① Deep INSIDER https://deepinsider.jp/

ディープラーニングの基礎を学べます。読み物としての側面が強いので、自分で プログラムを動かしたいという場合には下のホームページをお勧めします。

② Keras による、ものすごくシンプルな画像分類

https://qiita.com/hiroeorz@github/items/ecb39ed4042ebdc0a957

りんごとオレンジの画像を分類するというシンプルな内容です。教科書によくある手書 き文字認識と異なり自分で用意した画像で学習することが出来るため、より楽しめると 思います。 ②はディープラーニングプログラムの基礎中の基礎レベルの内容ですが、pythonの知識が 身についていないと理解はおろか、実行することすら難しいという場合もあると思います。 そのような場合には、Paiza ラーニング <u>https://paiza.jp/works</u> などで改めて pythonの 学習に取り組み、力をつけてから再挑戦してみてください。