2022年度 プログラミング演習1
前半 (Python プログラミング)

情報理工学部 実世界情報コース

目 次

1　プログラミング演習　ガイダンス

1.1　授業の概要

プログラミング技術は、コンピュータを用いて種々の問題を解いたり、革新的なサービスを実現したりするために、基礎的でありかつ必須の技術である。本科目ではプログラミングにおける基礎技術として、プログラムの制御構造（逐次・選択・反復）、データ表現方式（変数・配列・構造体）、データ処理方式（演算）、抽象化手法（サブルーチン）、データ授受手法（値渡しと参照渡し）を習得する。さらに、与えられた演習課題を通じてプログラムの組み立て手順を習得する。

プログラムを学習するには、外国語を体得するのと同様に、自発的に学び、自らがプログラムを体験することが最も近道である。この観点から、本授業では、受講者が各回の授業前に予習としてビデオ教材を用いた自主学習を行って講義内容について把握し、授業では座学による要点の確認と、演習形式で学習した基本項目を理解することでプログラミング技法を学ぶ。

1.2　到達目標

本演習では、実習を通して Python 言語と Java 言語によるプログラミングを修得する。本演習における目標は次の通りである．

1. Python 言語と Java 言語によってプログラムを作成し，デバッグして動作させる能力
2. Python 言語と Java 言語の文法と基本的なモジュールの利用方法
3. プログラム内容を示す基本的な仕様書を作成する能力

1.3　講義の流れ

- 小テスト 10 分：宿題であるビデオ視聴を行ってきたか講義冒頭に小テストを行う．
- 講義 20 分：ビデオ視聴をもとに，課題の説明を行います. 宿題をしておかないと取り残される程度の難易度とする．
- 演習課題 60 分：講義内容をもとに，実際にコーディングを行う演習を行う．
- 宿題：
  - 授業中に終わらなかった演習課題：宿題とする．締め切りは次回講義開始時とする．
  - ビデオ視聴：指定した講義ビデオの視聴を 1〜9 週目の宿題とする．11〜14 週目はビデオ視聴の宿題はないが，小テストは実施する．

1.4　評価方法・基準

２回の実技試験を通じて，プログラミング言語の文法と基本的なライブラリ関数の利用方法平常点評価を含む，プログラム作成能力を評価する (30 %).

毎週提示される演習課題を通じて，プログラミング言語の文法と基本的なライブラリ関数の利用方法を含む，プログラム作成能力を評価する (60 %).

期末のレポートを通じて，プログラム内容を示す基本的な仕様書を作成する能力を評価する (10 %).

出席状況は毎週 manaba+R の「出席カード」機能をもって取得する．講義に出席し，積極的な姿勢で課題に取り組む姿勢も「プログラム作成能力」の一部として評価する．

各課題，何%の得点がとれたかなどはできるだけ公開するが，どの程度の重みで最終的な成績に反映されるかは非公開とする．

1.5　テキスト・参考書

以下をテキストとする．

- 世界標準 MIT 教科書 Python 言語によるプログラミングイントロダクション第 2 版: データサイエンスとアプリケーション／近代科学社／第 2 版／ (2017/9/1) ／ 978-4764905184 ／
- 本格学習 JAVA 入門 佐々木 整／技術評論社／ 978-4774146904 ／

以下を参考書とする．プログラミング言語の教科書は人によって合う合わないがある．自分にあった教科書を探してみるとよいだろう．

- Python チュートリアル 第 3 版 Guido van Rossum (著), 鴨澤 眞夫 (翻訳) ／オライリージャパン／ 978-4873117539 ／
- 入門 Python 3 Bill Lubanovic (著), 斎藤 康毅 他／オライリージャパン／ 978-4873117386 ／
- Python からはじめる数学入門 オライリージャパン／オライリージャパン／ 978-4873117683 ／
- やさしい Java 高橋麻奈／ Sb クリエイティブ／ 4797313935 ／
- スラスラわかる Java 中垣健志、林満也／翔泳社／ 4798130796 ／

1.6　ライセンス上の注意事項: 動画を絶対に配布しない

本講義で取り扱う講義ビデオは「Lynda.com」とのライセンス契約を持って，皆さんに動画視聴の権限を付与している．講義ビデオは manaba+R で視聴でき，しかもダウンロードできる状態にしている．**ライセンス契約上，動画をダウンロードすることは認められているが，動画の配布は認められていない．本講義を履修していない者に動画を絶対に渡さないこと．**

万が一第三者への動画配布が発覚した場合，違約金が大学に請求される．違約金の原資は皆さんの学費である．

1.7　トラブル：病欠や忌引き

病欠や忌引きで講義を休まざる得ない場合は，まずは，可能な限り速やかに教員に連絡すること．履修要綱に定めてある用件を満たしている場合，救済措置・配慮措置を取れることがある．

病欠する場合は病院に行って，病院から診断書を発行してもらって提出すること．書類上の問題だけではなく，あなたの体調を良くするためである．体をいたわろう．忌引きの場合も，証明となる書類のコピーをとり，教員に提出すること．証拠の書類があれば,「ずる休みでない」と他学生に説明した上で救済措置が取れる．

スポーツ大会で欠席せざる得ない場合は，病欠や忌引きと違って, スポーツ大会は事前に参加することがわかっているはずである．参加が決まり次第，事務からスポーツ大会参加に関する書類をもらい，記載したものを「事前に」提出すること．

その他，何か事情がある場合は早め早めに教員に相談すること．

1.8　トラブル：不正に対する考え方

1.8.1　不正をすると一番こまるのはあなた

情報理工学部の中でも，実世界情報コースは特に取り扱う幅が広い．ソフトウェアだけでなく，ハードウェアを取り扱うからである．全てを取り扱っていたらとてもではないが４年では終わらないため, カリキュラムを設計する際には細心の注意を払い,「最低限，これだけ抑えれば上回生講義に耐えられる」ものを厳選して講義として取り扱っている．

そのため，わからないまま次に進むと，雪だるま式にわからなくなるリスクが高い．今なら高校の勉強はほぼリセットされた状態で全員横並びである．１回生にどれだけ基礎をしっかり抑えるかで，上回生がどれだけ楽になるかがきまる．たとえば, 東京理科大学の統計で大学１回生終了時の成績順位と，４年生終了時の成績には強い相関があることも示されている．入試の説明会で上級生を招くと，後輩となる受験生へのアドバイスとして「低回生講義はしっかり抑えておけ」と皆が口をそろえて言う．ここがふんばりどころなので，しっかり抑えておこう．

特に，プログラミング能力は「情報系」なら絶対に避けて通れない.「情報」理工学部を卒業した学生は「当然，プログラミングは最低限レベルはできるんでしょ？」という目で就職先から見られる．ずるをして良い成績をとってもいいことは何もありません．卒業してから世間が要求する能力と自分の能力のギャップにものすごく苦しむことになります.

不正をして良い成績をとっても後で倍以上になってしんどくなるだけで，良いことは何もないことを知っておこう.

1.8.2　教員は学生を守る立場にある

不正な方法で勉強した人にも良い成績を与えてしまうと，外部より「あの講義は不正をしても単位がもらえるから，ちゃんと真面目に勉強しているかわからない」と評価を受けるようになる．真面目に受けた人は何も悪くないのに，信用されなくなってしまう．教員には，真面目に受けた学生を守るために，不正の防止措置を取る義務がある.

また，ずるをした人は後で大きなツケを結局払うことになるため, 不正をしようとしている学生を守る意味でも，不正の防止措置を取らなければならない.

一方で，教員は可能であれば不正防止措置を取りたくない．防止措置を取るための労力があるくらいなら，講義の内容の品質向上に労力をかけたいと思っている．また，不正防止措置は学生・教員の双方にとって気分がいいものではない．無駄にログをあさって不正をしていないかチェックする仕事や,「これをしたらＦにする」といったような脅しとも取れるアナウンス作業はお互いに気分がいいものではない.「みんなが不正しないように約束しましょうね」，で終わればそれが最高であると思っている.

教員は，学生に学問を教授するため，学生を守るために大学に勤めている．不正を防止することを目的に働いているわけではない．せっかく学費を払っているのだから，本来の使い方で教員を効率的に使おう.

1.8.3　どのような行為が推奨されて，どのような行為が不正として処罰されるか

大学は基本的に学ぶための場所である．そのため，**学ぶための行為は基本的におおらかに認められる**．恐れずどんどんやろう．教員は頑張って理屈をつけて学生を守る.

一方で，**「学びの趣旨（目的）をスポイルする行為，あるいは，本来やらなければならないことをやらずに良い評価を得ようとする行為」は基本的に不正行為**である．中でも特に，大学が「不正行為である」と明確に提示している行為は絶対にやめよう．ルールと前例に従って機械的に処罰が行われてしまうため，擁護したくても擁護できない.

たとえば, 他人と宿題について相談する行為は推奨される．どんどん相談しよう．自分のプログラムを他人に見せて，コメントを貰おう．他人のプログラムを見て，良いところを自分のプログラムに取り入れよう．コメントをもらうこと，良いところを取り入れることは上達への早道である.

一方で，他人の書いたソースコードを丸丸コピーして,「自分が作りました」と言う行為は不正行為である．先輩や友人からソースコードを貰って提出した場合は，不正をした本人とソースコードを提供した友人の両方が，前例に従って機械的に処罰される.

「参考にする」行為は推奨され「丸々コピーした」行為は不正として処罰される．これらの線引きは意外と難しいが，ひとつの基準として,「主たる部分を自分で行ったか」を知っておこう．他人のソースコード

を参考にして，自分なりにアレンジを行って取り入れたのであれば, あくまで主体は自分で，他人は従であるから，不正ではない．他人のソースコードをコピーして変数名だけ変えたというケースは，他人に主体があるから，明確な不正である．

自分の中で,「これこれこういうところを自分の力でやったから，これは自分の作品だ」と説明できるかどうか，セルフチェックするようにしよう．参考かコピーか判断がつかない場合は教員や TA に相談しよう．

注意）自分のレポートを他人に見せる行為そのものを禁止にしている講義もある．これはかなり厳しい不正防止措置であり，本来は推奨したいことを（嘆かわしいが）禁止せざるを得なくなった例である．このような講義は先輩が「やらかしている」ことが多い．

1.8.4 不正をするとどうなるか

不正が一度発覚すると，正当に講義を受けている学生の信用を守るため, 教員側は外部に対して「適切に不正防止措置を取りました」と示す義務がある．

不正を防止する措置は気分が悪いものがほとんどである．そして，不正防止措置には準備期間が必要であるため，**多くの不正防止措置は次年度の講義から適用されはじめる．かわいい後輩に対し，とても気分が悪い制度を先輩のあなたが残すことになる**．逆を言えば, 不正防止措置がゆるいこの講義の現状は，先輩のおかげである．

「単位が全部Ｆになって終わり，私の自己責任」ではない．後輩にとても大きな迷惑がかかることを自覚しよう．そして，良い空気を残してくれている先輩に感謝し，見習おう．

2　講義の進め方

本テキストに記載の内容は秋学期開始時の予定である．授業の実施方法等については状況によって変更される場合もある．

2.1　プログラムの開発環境

本講義の演習に取り組むには Python と Java の開発環境が必要である．開発環境には色々なものがあるが，それぞれの言語の基本的な機能はいずれの環境においても利用可能である．授業においては導入の簡単な web ブラウザ上で動作する環境を紹介し，利用する．

Python については Google Colaboratory を，Java については Online Java Compiler - online editor をそれぞれ利用する予定である．Google Colaboratory の利用には Google アカウントが必要となるので準備しておくこと．

インターネット上のサービスを利用し web ブラウザの上で動作する環境はソフトウェアのインストールも必要なく，簡単に利用できる．ただし、新しいウィンドウを開いたりするグラフィック機能を利用するのが難しく，長時間の計算ができない制限もある．

そこで，各自の PC に開発環境ソフトウェアをインストールすることを強く推奨する．多少の手間（といっても簡単である）はかかるが，インターネット環境がなくても使え，グラフィック機能や高度な計算も行える．

2.2　出席の取り方

manaba+R の「出席カード」機能にて，出席を取る．教員が各講義時に「出席カード番号」を示すのでその番号を以下 URL にアクセスして打ち込めば，出席とカウントする．

`https://ctat.ritsumei.ac.jp`

遅れて入室した場合は TA あるいは教員に申し出ること．申し出なかった場合，出席とカウントされない場合がある．

2.3　小テスト

毎週講義開始時から１０分間小テストを行う．小テストは manaba+R の「小テスト機能」で出題する．

小テストの開始は講義開始のチャイムが鳴ってからである．早めに入室し，テストを受けれる準備を整えておくこと．

システムトラブルなどがあった場合は速やかに TA・教員に申し出ること．学びとしての制限時間は１０分で，各学生のモラルのもと，時間内に提出することを求めるが，トラブル対応用にシステム自体は１５分まであけておく．

2.4　演習課題

演習課題は本テキストに掲載されている．各自，端末で課題を解き，manaba+R のレポート機能により課題を提出する．manaba+R には，ipynb 形式でエクスポートしたものを提出すること．

完了しきれなかった分の演習課題は宿題とする．提出締め切りは翌週の講義が始まるまでとする．

解いていない問題があった場合は厳しい減点が行われる．そのため，わからない点があれば，教員や TA に質問する，友人と相談するなど，講義開始時までに解決してくること．

※課題の回答方法

「確認せよ」「出力せよ」「表示せよ」などというような指示がある場合は print() やファイルの作成などを使って，確認した形跡，出力結果を示してください．

2.5　中間・期末テスト

10 週目と 15 週目にそれぞれ中間・期末試験を行う．中間・期末試験は実技試験であり，manaba+R の「小テスト機能」で問題を出題し，手元の端末でコーディングした結果を「小テスト」の回答として manaba+R に提出する．詳しくは中間・期末試験の前の週に説明する．

3　宿題：動画の視聴

毎週指示される動画を視聴し，内容を理解しておくこと．動画は manaba+R のコンテンツページにアップロードされる．

教員は各学生が動画を視聴したかチェックする権限を manaba+R 上に持っている．また，毎週行われる小テストにて，動画の理解度の確認を行う．

4　Google Colaboratory の利用方法

4.1　ログイン方法

1. https://colab.research.google.com/を開く．
2. Google Colaboratory の画面右上にある 「ログイン」をクリックして Google アカウントでログインする．
3. 最初は「Colaboratory へようこそ」というノートブックを開く．

4.2　プログラムの開発

ログインした直後，左ペインに目次，右ペインにノートブックの内容が表示される．

ノートブック内のテキスト部分やコード部分をダブルクリックすればその部分を編集するモードとなり，自由に編集できる．テキスト，コードそれぞれの固まりは「セル」と呼ばれる．セルとセルの間にカーソルを合わせると，そこに新たなセルを追加するためのボタンが現れるので，これをクリックすることで新たなセルを追加できる．

コードのセルをクリックするとセルの左端に実行ボタン (右向きの三角形) が現れる．これをクリックすると，セル内のプログラムが実行できる．コード中の `print()` 関数の結果等はコードセルの直後に出力される．

4.3　ノートブック (ipynb ファイル) のダウンロード

Google Colabratory の画面左上にある「ファイル」を クリックしてメニューから 「.ipynb をダウンロード」を指示すればよい．

4.4 ノートブック (ipynb ファイル) を開く方法

1. Google Colabratory の画面左上にある「ファイル」をクリックしてメニューから「ノートブックをアップロード」を指示する.（メニューから「ノートブックを開く」を指示し，現れたウィンドウで「アップロード」を指示する方法も可）

2. 「参照」をクリックしてノートブックのファイルを 指定するか， ノートブックのファイルを点線で囲われた領域にドラッグ&ドロップすればノートブックを開いた状態になる.

4.5 開発環境へのファイルのアップロード

1. Google Colaboratory の画面左端にあるファイルのアイコンをクリックし， 左ペインにファイル一覧を表示する.

2. ファイル一覧を表示する領域の上にある 「アップロード」をクリックしてアップロードしたいファイルを指定する.

4.6 開発環境からのファイルのダウンロード

1. Google Colaboratory の画面左端にあるファイルのアイコンをクリックし，左ペインにファイル一覧を表示する.

2. ダウンロードしたいファイルを右クリックしてメニューから「ダウンロード」を指示する． ファイル名の右にある 3 つの点が並んだアイコンを左クリックしてメニューを表示することもできる.

5 自分の PC 上で動作する開発環境構築について

個々人が所有するＰＣは千差万別のため，環境構築法を公式にサポートすることは実務上不可能である. そのため，自宅で宿題に取り組むための環境構築法は「ノンサポート扱い」せざるを得ない．可能な限り助けるように努力するが，環境によっては対応できないことがあることを理解いただきたい.

一方で，自宅に開発環境を作ると，演習が大変に捗る.（ノンサポート扱いにせざるを得ないことが心苦しいが）ぜひとも，自宅で環境開発の構築を行っておこう．以下に自宅で宿題を取り組むための手順を示しておく.

5.1 Python 開発環境の構築法

macOS および Windows における Python 開発環境の構築方法を Panopto に動画としてまとめた。一部音声が切れているものがあるが、仕様である。

それぞれ、macOS を使っているものは動画プレイリスト：macOS での Python 環境（Anaconda）の構築を、Windows を使っているものは動画プレイリスト：Windows での Python 環境（Anaconda）の構築を参照すること。

なお、macOS を M1/M2 チップを搭載したコンピュータで使っている場合は、動画プレイリスト：macOS での Python 環境（Anaconda）の構築中の「macOS での Anaconda のインストール M1 1」および「macOS での Anaconda のインストール M1 2」を参照すること。

5.2　トラブル：わからないことが出てきた

担当教員に質問に行こう．メールで質問しても良いし，訪問してもＯＫ．教員の連絡先は第 1 回講義時に皆さんに連絡する．教授というと近寄りがたい別人種のように感じるかもしれないが，皆さんと同じ人間である．我々は質問に来てくれるととてもうれしい．是非きてほしい．

情報理工学部ではクリコアラという，講義の質問を受けに行く場を設けている．基本的に毎日開催だから，そちらに質問に行っても良い．若手の優しい先生が対応してくれる．

講義時間中に TA や ES にたずねることも効果的である．彼らは君たちの先輩であり，後輩である君たちを大変かわいいと思っている．また，彼らは君たちの学費を原資とする資金で雇われているため，君たちは，正当に，質問する権利がある．

その他，友達を作ることも強くお勧めする．大学で作る友達は一生の友達になることがとても多い．お互いに良い意味で助け合い，一緒に勉強して仲良くなろう！

6　第 1 週課題

6.1　課題 1-1（教員と一緒にやろう）

「基本的なデータ型」について解説した事前学習用の動画を視聴せよ．動画と全く同じコードを記述・実行し確認せよ。

そのコードと実行結果を含む ipynb ファイル形式で manaba+R に提出せよ．

6.2　課題 1-2

「変数を使う」ならびに「いろいろな演算子」について解説した事前学習用の動画を視聴せよ．動画と全く同じコードを記述・実行し確認せよ。（続けて実行すればよい．途中で余計なコードが入ってもよい）

そのコードと実行結果を含む ipynb ファイル形式で manaba+R に提出せよ．

6.3　課題 1-3

変数 a,b,c,d の値がそれぞれ 2,3,4,5 の場合，計算結果が 10 になる計算式を考えよ．このとき、すべての文字 a,b,c,d を用いること.（四則演算（+,-,*,/），小数点以下を切り捨てる割り算の演算子，累乗の演算子を使えばできるだろう.）

変数 a,b,c,d にそれぞれ 2,3,4,5 を代入した上で，自分で考えた計算式を実行するコードを記述・実行し，確かに計算結果が 10 になっていることを確認せよ．

6.4　課題 1-4

自分の名前をローマ字で表現した文字列を変数 name に代入せよ．ただし，名前と苗字の間には半角スペースをひとつあけよ．たとえば，立命太郎（リツメイタロウ）の場合，`name = 'Ritsumei Tarou'` とすればよい．続けて変数 name の中身を表示した後，変数 name の長さを表示するプログラムを作成せよ．

6.5　課題 1-5

課題 1-4 で作ったプログラムを `Ritsumei Tarou` という名前の場合で実行すると，14 という数字が表示されていた．しかしこの 14 という数字は名前と苗字の間にある半角スペースの数も含めた文字の数である．このスペースを文字数としてカウントしないプログラムを作成したい．

課題 1-4 で作ったプログラムを改造し，変数 name の中身を表示した後，名前と苗字の文字数の和を表示するプログラムを作成せよ．(プログラムの実行中に余計な画面表示が行われても良い).

6.6　課題 1-6

Python における計算ではカッコを使った計算式も使うことができる．例えば，変数 x, y を使った $(x+y)^2$ という計算式は `(x+y)**2` というコードで実現でることもできる．

以下の手順でつるかめ算を計算するプログラムを作成せよ．

1. 変数 foot に鶴と亀の足の数の和として 26 を代入する．
2. 変数 head に鶴と亀の頭の数の和として 9 を代入する．
3. 鶴の頭の数を変数 foot,head を使って計算し，表示する．
4. 亀の頭の数を変数 foot,head を使って計算し，表示する．

7　第 2 週課題

7.1　課題 2-1

Python3.5 ではユーザーからの入力を受け付ける関数として input() という関数が用意されている．input() 関数は引数として文字列を受け取り，その文字列を表示した上で，ユーザーから入力を受け付け，受け付けた入力を文字列としてかえす．

例えば，以下のコードを考えよう．

```
name = input('What is your name? ')
print('Your name is ' + name)
```

このコード 1 行目では，What is your name?と画面に表示し，ユーザーに文字列を入力させる．そして，変数 name に入力された文字列が代入される．2 行目では，Your name is と表示し，その後，ユーザーから入力された文字列を表示する．(実際に確認してみよう)

キーボードから名前と趣味をそれぞれ入力させ,「〜〜〜の趣味は〜〜〜」であると画面表示するプログラムを作成せよ．

7.2　課題 2-2

2 つの数字をユーザーに入力させ，その和を計算するプログラムを作成したい．しかし，以下のコードは意図通りの動作を示さない (実行してみよ)．

```
a = input('Input a number:')
b = input('Input a number:')
c = a + b
print ('Sum is ', c)
```

理由は，input() 関数は受け付けた文字を文字列としてかえすからで，このままでは文字列の連結とみなされてしまうからである．

そこで，文字列を数字に変換する関数として，int 関数や float 関数を活用しよう．

```
d = int("14")
e = float("1.23")
print ('int("14") = ', d)
print ('float("1.23") = ', e)
```

int("14") というコードは文字列としての"14"を整数の 14 に変換する関数である．float("1.23") というコードは文字列としての"1.23"を小数の 1.23 に変換する関数である．

input(), int(), float() 関数などを使い，2 つの数字をユーザーに入力させ，その和を計算するプログラムを作成せよ．(計算結果を表示せよ)

7.3　課題 2-3

ひとつの文字列を入力させ，入力された文字列を大文字に変換した上で，それに含まれる「A」という文字を全て「B」に変換せよ．変換した文字列を表示するプログラムを作成せよ．例えば，abcdef と入力さ

れた場合は BBCDEF と表示されれば合格である．

7.4 課題 2-4

ロボットをはじめとする実世界に関連したプログラムを書くには，三角関数は必須であるといってもよいだろう．sin 関数や cos 関数は numpy モジュールに入っている．以下のように `import numpy as np` などを使用前に宣言する．

```
import numpy as np
print(np.sin(3))
```

上記の `import` は numpy モジュールを `np` という名前で使用可能にするものである．`np.sin` の部分が sin 関数という意味である．上記プログラムは sin(3) の値を表示する．

同様に，numpy モジュールには cos, tan などが入っている．使うときには `np.cos` や `np.tan` などとかく．

「numpy 円周率」などで Google 検索をかけ，円周率 π を表示するプログラムを作成せよ．

注）この課題は文法事項を自分で検索する能力をつけるための練習だから，友達に直接的な答えを教えないように．調べ方を教えてあげよう．

7.5 課題 2-5

あなたはある国民的アイドルグループのファンである．しかし，残念なことに，そのアイドルグループは解散してしまったようである．最終活動日は 2016 年 12 月 31 日であった．この最終活動日が何曜日か調べるプログラムをかけ．必ずしも○○曜日と表示する必要はない．例えば，月曜日なら数字の `0` を，火曜日なら数字の `1` を,...，日曜日なら数字の `6` を表示することで曜日の表示として代用してよい．

7.6 課題 2-6

大文字のみからなる文字列を入力させ，入力された文字列の「`A`」の部分を「`B`」に,「`B`」の部分を「`A`」に変更するプログラムを作成せよ．例えば,「`AAABBBCCC`」という入力をした場合,「`BBBAAACCC`」という出力が出れば合格である.「`AAAAAACCC`」や「`BBBBBBCCC`」とならないように注意せよ．

なお，入力された文字列が大文字のみからなることのチェックする機能は実装しなくてよい．

ヒント：`A` を何か適当な小文字に変換，`B` を `A` に変換，その後，ある操作をすれば達成できる．

8　第 3 週課題

8.1　課題 3-1

次の実行結果を確認せよ.

```
v = list('Rtsumeok')
v.insert(1, 'i')
v.remove('o')
v.extend(['a', 'n'])
v.insert(7,'i')
print(v)
```

8.2　課題 3-2

スライスを用いて次のリストの'c' から'e' までの部分についてのリストを表示せよ．また，lst[1:6:2] の結果を確認して，スライスを用いて ['c', 'e', 'g'] が出力されるようにせよ.

```
lst=['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
```

8.3　課題 3-3

以下の表を参考に 3 つの辞書を作成せよ.（辞書の変数はそれぞれ a, b, c とする）
※実行結果は表示すること

	a	b	c
name	John	Betty	Jason
age	21	22	22
score	78	86	99

8.4　課題 3-4

課題 3-3 で作成した変数 a,b,c を用いて，3 つ辞書を要素にもつリスト scores を作成せよ．リスト内の辞書の順番は変数名のアルファベット順とすること．また，scores[1].get('name') を実行して結果を出力するためのコードと，Jason の score である 99 を出力するためのコードを作成せよ.

8.5　課題 3-5

次の表を参考に 2 つのセットを作成せよ.（セットの変数はそれぞれ s1 と s2 とする）
※実行結果を表示すること

s1	s2
book	note
dog	pen
cat	dog
spoon	fork
fork	

8.6 課題 3-6

課題 3-5 の結果を用いて以下の内容を出力し確認せよ.

```
s1.intersection(s2)
s2.intersection(s1)
s1.union(s2)
s2.union(s1)
s1.difference(s2)
s2.difference(s1)
```

9　第 4 週課題

9.1　課題 4-1

0,1,2,3,...,9 の要素を持つリスト x を作成せよ．x の要素を for 文を使ってすべて表示せよ．

9.2　課題 4-2

あなたは今，冬休みを使ってアメリカへ旅行に来ている．せっかくだから現地のスーパーで肉や野菜を買ってみようと入り込んだはいいが，現地では 1 ポンド（pound, 重さの単位）あたりの値段で食品の値段が表示されており，自分のお小遣いで目の前のアメリカ産サーロインステーキが買えるかどうか判断がつかない.（注：外国の中にはヤードポンド法で重量を計測し，量り売りで食料を打っているところがあります.）

そこで，キーボードより，1 ポンドあたりの肉の値段，あなたのお小遣い（ドル）を入力させ，200 グラムの肉を購入しようとした場合に予算以内に収まるか判定するプログラムを作成せよ．

ただし，1 ポンドは 453.592 グラムとする．

9.3　課題 4-3

ユーザーから文字列を入力させよ．入力された文字列の文字数が 9 でない場合は，while 文を使い，入力された文字列の文字数が 9 となるまで，何度でもユーザーから文字列を入力させなおせるプログラムを作成せよ．

9.4　課題 4-4

ユーザーから 3 つの人名を文字列として入力させ boya,boyb,boyc に代入せよ．それぞれの文字列のうち，最も長い文字列を表示するプログラムを作成せよ．最も長い文字列が複数ある場合はその全てを表示するプログラムとせよ．

9.5　課題 4-5

円周率を数値計算で計算することにチャレンジしよう．いくつか円周率を得る方法はあるが，最もシンプルな（だが性能はあまりよくない）もののひとつは，逆正接関数 $\tan^{-1}(x)$ をマクローリン展開した以下の公式を使うことである．

$$\tan^{-1}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots \tag{1}$$

$\tan^{-1}(1) = \frac{\pi}{4}$ であることから（幾何的な図で確かめよ），

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots \tag{2}$$

$$= \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} \tag{3}$$

である．上記式の右辺の級数を第１０項まで足した結果を表示するプログラムを for や while 文などを使って作成せよ．

ヒント：「第 n 項を計算する」という処理を n を増やしながら繰り返すプログラムを for や while を用いて作り，計算した第 n 項の値を別の変数 sum などに足していけば良い．

発展的内容：以下は成績には加算しないが，興味があるものはぜひ取り組んでみてほしい．第１項までなら小数点以下何ケタまで正しいか，第１０項なら何ケタまでが正しいかなど，足し合わせる項の数と計算結果の正確性との関係を調べてみよ．関係性はプログラムソース中のコメント文として記述せよ．ここで，コメント文とはプログラム中にメモを残すための文法であり，以下例で示すように，#を記述することで記述できる．

```
# プログラム中にシャープを記述すると，同じ行に書かれた以降の文章が無視される
# 無視されることを利用して，プログラムにメモを残せる
# わかりやすいプログラムを記述することに有用なので，たくさんコメントを書こう
a = 3
b = 4
print(a+b) # a と b の和を表示する
```

9.6 課題 4-6

以下のプログラムを書き換えて，continue 文を使わずに同じ出力を行うプログラムを記述し，実行せよ．但し，1 行目の for x in range(10): はそのまま変更せず，この for ループの中身だけを書き換えて記述すること．

```
for x in range(10):
  if x > 5:
    print ('large')
    continue
  print(x)
```

10　第 5 週課題

10.1　課題 5-1

次のコードを実行せよ.

```
# 課題 5-1 のサンプルプログラム
f=open('temp.txt', mode='w', encoding='utf-8')
f.write('test\n')
f.write('あいうえお\n')
f.close()
```

上記のプログラムを実行後，!more temp.txt を実行しテキストファイル temp.txt の内容を確認せよ．また，encoding='utf-8' を encoding='shift-jis' に変更して再度実行し，その結果を!more temp.txt で再度確認せよ．二つの違いはなぜ起こるのか簡単に記述せよ（コメント文として記述すればよい）.

10.2　課題 5-2

課題 5-1 のコード（utf-8 のほう）を実行したのちに次のコードを実行すると ['test\n', 'あいうえお\n'] のような結果がかえってくる．for 文，strip() メソッドを用いて ['test', 'あいうえお'] のような新しいリスト w を作成せよ.

```
f=open('temp.txt', encoding='utf-8')
v=f.readlines()
f.close()
print(v)
```

10.3　課題 5-3

a={'name':'John', 'age':21, 'score':78} を用いて次のようなファイルを生成するプログラムを空欄を埋めて完成せよ.

```
name/John
age/21
score/78
```

```
data={'name':'John', 'age':21, 'score':78}

with open('dict.txt', mode='w', encoding='utf-8') as f:
＜空欄部分＞
```

10.4 課題 5-4

input() 関数は x = input('Please Enter Number: ') のように使用することができ，実行すると画面に Please Enter Number: が出力され，ユーザのキー入力を待つ．ユーザがキー入力したのちエンタキーを打つと x にユーザが入力した内容が文字列をして格納される．input() 関数を用いてユーザに入力を実行させ，入力結果をファイルに書き込むプログラムを作成せよ．ただし，ファイル名は user.txt とする．

10.5 課題 5-5

課題 5-4 を，繰り返しユーザの入力を受けてファイルに書き込むプログラムに変更せよ．ただし，ユーザが q を入力すると繰り返し作業を終えプログラムを終了するようにせよ．また，各入力をファイルに書き込む際は改行コードを追加してファイルを見やすくすること．

ただし，ファイル名は「user2.txt」とし,「q」はテキストファイルに書き込まないこととする．

10.6 課題 5-6

メモ帳を開いて Python の作業フォルダに次の内容を持つファイル fruits.txt を作成せよ．

```
apple
banana
apple
lemon
apple
orange
```

このファイルを読み込んで apple の数を数えて出力するプログラムを作成せよ．

11　第 6 週課題

11.1　課題 6-1

2 つの引数を受け取り，足し算，引き算，掛け算，割り算を行う関数 add, sub, mul, div を def 命令を使って定義せよ．これらの関数を使って，$2+3$, $4-1$, 2×3, $6/2$ を計算し結果を出力せよ。

11.2　課題 6-2

2 つの引数を受け取り，足し算，引き算，掛け算，割り算を行う関数 add, sub, mul, div を lambda 式を使って定義せよ．これらの関数を使って，$2+3$, $4-1$, 2×3, $6/2$ を計算し結果を出力せよ．

11.3　課題 6-3

課題 6-1 で作成した add, sub, mul, div を要素に持つリスト（あるいは辞書型）ope を定義せよ．キーボードから 2 つの数字を入力させよ．入力させた数字は変数を a,b に保存するとする．キーボードからさらに一つ数字を入力させ，その数字が 0 ならば，a+b を，1 ならば a-b を，2 ならば a*b を，3 ならば a/b の計算結果を表示するプログラムを作成せよ．ただし，四則演算すべての処理を実行すること．

ヒント：

```
ope = [add, sub, mul, div]
```

```
ope[x](a,b)
```

などのコードを使う．これらのコードの意味をよく考えよう（リストの要素として関数を取り扱っていることに注意せよ）．

11.4　課題 6-4

これまでは数字を受け取って数字をかえす関数ばかりを扱っていた．関数は数字だけでなく，文字列，リスト，タプル，はては関数まで，多くを引数に取ることが可能である．ここでは，関数とリストを受け取ってリストを返す関数を作成する練習をしよう．

以下プログラムを改造し，関数 f とリスト x を受け取り, list(map(f,x)) の計算結果と同じオブジェクトを返す関数 mymap() を作成し，実行せよ

```
vect = [1,2,3,4,5]
func = lambda x:x**2
def mymap(f, x):
   [ここを埋める]
print(mymap(func, vect))
```

ただし，Python の組み込み関数のひとつである map 関数を使わずに実装すること．

ヒント：for 文を使い，x の各要素 x_i に対して $f(x_i)$ を求めればよい．

11.5 課題 6-5

なめらかな関数 $f(x)$ を考えると，その微分 $f'(x)$ は以下で定義される（高校の教科書を参照せよ）.

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{4}$$

この定義によると，h を 0 に近づけると右辺にある lim 内の数式は，$f'(x)$ に近づくことがわかる．したがって，小さな h を取り（例えば $h = 0.00001$），以下で関数 $g(f, x)$ を定義すると，$g(f, x)$ は $f(x)$ の微分 $f'(x)$ の近似値を計算することがわかる.

$$g(f, x) = \frac{f(x + 0.00001) - f(x)}{0.00001} \tag{5}$$

ただし，上記関数は関数と実数を引数に取る関数であることに注意する．厳密にいえばおそらく皆さんはこのような関数を引数に取る関数を数学の講義で習っていない．しかし，定義を理解するだけなら容易であろう.

以上の前提知識のもと，(5) で定義される関数 $g(f, x)$ を実装せよ．以下の関数を (`def` 命令などで) 実装し，それぞれの関数の $x = 3$ の場合の値を出力せよ

$$f_1(x) := x^2 \tag{6}$$

$$f_1'(x) := 2x \tag{7}$$

$$\tilde{f_1'} := g(f_1, x) \tag{8}$$

ただし，:= 記号は左辺の関数を右辺で定義するという意味である.

注意：sin 関数や cos 関数を使う場合は 2 週目の課題を思い出そう．以下のように `import numpy as np` などを使用前に宣言する.

```
import numpy as np
print(np.sin(3))
```

`np.sin` の部分が sin 関数という意味である (numpy モジュールに sin 関数が含まれている).

コメント１：手計算で関数の微分を計算しなくても，微分の近似値を求められることに驚くと良いだろう．例えば，$(x^2 + 2)/(x + 1)$ の微分を計算せよと言われると，少しだけ面倒に思うだろう．特定の値に対する微分のおおざっぱな値を知りたいときには，微分を手計算で求めず，上記議論により近似的に計算すれば良いのである.

コメント２：実は，このような差分による微分の近似的計算方法はロボット制御にも使われている．例えば，多くの車両ロボットでは車輪の角度を計測するために「エンコーダ」というデバイスが使われている．エンコーダは車輪の角速度を直接計測できない．しかし，車輪の角度の微分を上記式で計算することで，近似的に車輪の角速度を求めることができる．本来は角度しか計測できないデバイスから車輪の角速度を計算することができたのである．より高速あるいは高精度に動くロボットの制御法を開発するには，角度だけでなく，角速度も計測しておく必要がある（詳しくは上回生での講義で．お楽しみに！）．こういった，数学の知識，物理学の知識，コーディング技術を融合させることがロボット制御において重要になってくることを認識し，実際に役立つと感じておこう.

11.6 課題 6-6

課題 6-5 で作成した `g(f,x)` は,「関数 f と値 x を引数に取り，微分 f' の x における値 $f'(x)$ の近似値をかえす関数」であった．課題 6-5 のプログラムを改造し,「関数 f」を引数に取り,「微分 f' の近似値をかえ

す関数 g」を返す作用素 d を作成せよ．以下の関数を (def 命令などで) 実装し，それぞれの関数の $x = 3$ の場合の値を出力せよ

$$f_2(x) := \sin(x) \tag{9}$$

$$f_2'(x) := \cos(x) \tag{10}$$

$$\tilde{f_2'}(x) := d(f_2)(x) \tag{11}$$

ヒント１：数学では，関数を受取り，関数を返す写像のことを作用素とよぶ．この課題は微分の近似関数を計算する作用素 $d; f \mapsto g$ を実装する課題である．(11) の右辺にはカッコが二つあって不自然に思うかもしれないが，d は関数を受け取って関数を返す作用素（写像）だから，$d(f_2)$ の部分でひとつの関数をあらわす記号と思えば自然に思うことができる．

ヒント２：以下のようなコードで実装することができる．

```
import numpy as np
def d(f):
    return 「ここを埋める１」
f2  = lambda x:np.sin(x)
f2d = lambda x:np.cos(x)
f2t = 「ここを埋める２」
print(f2(3), f2d(3), f2t(3))
```

「ここを埋める１」部分は lambda 式を使って関数を定義したものを記述するか，課題 6-5 で使った関数を使って記述すればよい.「ここを埋める２」部分は d は関数を受け取って関数を返す関数であることに注意しつつ，コーディングしよう．

コメント：作用素 d を数式で書き下すこともできるが，コーディングとやや離れた表現になってしまうため，本課題が終わるまでは挑戦しないほうが良い．

11.7 発展課題 6-1(成績に含めません)

d は f の１階微分 f' の近似値を計算する関数であった．この考え方を応用し，

```
def d2(f):
   return d(d(f))
```

と関数 d2 を定義すれば，この関数は f の２階微分の近似値を計算する関数を返す作用素であることがわかるだろう（なぜこれで２階微分の近似値を求めることができるか理屈をよく考えよ）．上記コードを応用すれば，3 階微分，4 階微分の近似値を計算する作用素も簡単に実装できそうである．

ところで，関数 sin は微分を何度も繰り返すと，$\cos, -\sin, -\cos, \sin$ というように，４階の微分でもとの sin 関数が得られる；

$$\sin'(x) = \cos(x) \tag{12}$$

$$\cos'(x) = -\sin(x) \tag{13}$$

$$-\sin'(x) = -\cos(x) \tag{14}$$

$$-\cos'(x) = \sin(x) \tag{15}$$

それでは，sin の４階微分の近似値を返す関数と，もとの関数 sin はどの程度異なる値をとるのであろうか？

本課題では，与えられた関数 f の４階微分の近似値を計算する関数を実装し，$x = 10$ における sin の４階微分の近似値と，$x = 10$ における sin の値を表示するプログラムを作成せよ．

コメント：なぜ同じ値を取らないか考えてみよ．

ヒント：「浮動小数点型」でキーワード検索してみよ．ある数字をきわめて小さい値で割り算した場合，正しい計算結果が得られるだろうか？例えば，`(10.0/0.00001)*100000` や `(10/0.0000000001)*10000000000` というコードはどちらも手計算では 10 という計算結果になるが，Python で実行した場合，`10` という計算結果を得ることができるか？

11.8 発展課題 6-2(成績に含めません)

せっかくなので，発展課題 6-1 で作成した sin の４階微分の近似関数と，sin 関数をグラフとしてプロットしておこう．

以下のコードは sin 関数，cos 関数をグラフプロットするコードである．試しに実行してみよ．

```
%matplotlib inline

import numpy as np
import matplotlib.pyplot as plt
x = np.arange(-10,10,0.1)
plt.plot(x,np.sin(x), x, np.cos(x))
plt.show()
```

上記で用いた `np.arange` 関数は，`range` 関数を拡張して小数も扱えるようにした関数である．例えば `np.arange(-10,10,0.1)` は `[-10, -9.9, -9.8, ..., 9.9, 10.0]` というように，`-10` から `0.1` 刻みで値を増やしていって得られる数字を集めたリストを返す（厳密にいえば `np.ndarray` という型のオブジェクトを返すのだが，とりあえずはリストを返すと思っておくと良いだろう）．

以下のコードは sin 関数，cos 関数，$f(x) = x + 1$ をグラフプロットするコードである．試しに実行してみよ．

```
%matplotlib inline

import numpy as np
import matplotlib.pyplot as plt

def func(x):
  return x+1
x = np.arange(-10,10,0.1)
plt.plot(x,np.sin(x), x, np.cos(x), x, func(x))
plt.show()
```

上記コードを改造し，課題 6-5 で作成した sin の４階微分の近似関数と，sin 関数をプロットせよ．**ただし，$h = 0.0001$ の設定のもと，差分微分の近似計算を行うこと．**

コメント：理屈通り，sin の４階微分の近似関数は sin を近似できているかグラフを見て判断してみよ．近似できていないと判断するのであれば，なぜ近似できていないものができてしまったか考察してみよ．h を

いろいろな値に変えて実行してみると，面白い結果が出てくる．計算機上の計算結果と，手計算による厳密な計算結果の違いを感じ取ろう．

12　第 7 週課題

12.1　課題 7-1

個人の情報を扱うクラス PeopleData を作成せよ．ただし，PeopleData で作成されたインスタンスは以下のような動作ができるようにする．

```
In [1]: d = PeopleData('Ritsumekan Taro')
In [2]: d.get_name()
Out[2]: 'Ritsumekan Taro'
```

12.2　課題 7-2

課題 7-1 で作成したクラス PeopleData を継承する BigPeopleData を作成せよ．ただし，BigPeopleData で作成されたインスタンスは以下のような動作ができるようにする．

```
In [1]: e = BigPeopleData('Ritsumeikan Hanako', 20, 165)
In [2]: e.get_name()
Out[2]: 'Ritsumeikan Hanako'
In [3]: e.get_age()
Out[3]: 20
In [4]: e.get_height()
Out[3]: 165
```

12.3　課題 7-3

manaba+R から data.txt をダウンロードし，作業フォルダに置く．以下のコードを入力して実行し，data[5].get_age()，data[6].get_name() などで結果を確認せよ．（ただし，課題 7-2 のクラスが使えることが前提である）

```
with open('data.txt', 'r') as f:
    size = int(f.readline())
    data = []
    for n in range(0, size):
        name = f.readline()
        age = int(f.readline())
        height = int(f.readline())
        data.append(BigPeopleData(name.strip(), age, height))
```

12.4　課題 7-4

課題 7-3 を用いて，読み込んだ BigPeopleData のインスタンスから age が 20 のインスタンスを探し出し，その人の情報（名前，年齢，身長）を表示するプログラムを作成せよ．

12.5 課題 7-5

インスタンス生成時の時刻を記録して，time_diff メソッドを実行すると現在の時刻との時間差を出力するクラス TimeCount を作成せよ．

```
In [1]: t=TimeCount()
In [2]: t.org
Out[2]: datetime.datetime(2117, 12, 14, 17, 33, 30, 380427)
In [3]: t.time_diff()
      0:00:14.251754
```

12.6 課題 7-6

課題 7-5 に__str__メソッドを追加し，インスタンス生成時の時刻を出力するようにせよ．

```
In [1]: t=TimeCount()
In [2]: print(t)
     2117-12-14 17:40:31.042797
```

13　第 8 週課題

13.1　課題 8-1

現在の時刻を自動取得し，アメリカ式（March 26 2012 のように月 日 年）で表示するプログラムを作成せよ．

ただし，月は数字ではなく，英語で表示せねばならぬものとする．したがって，

```
import calendar
import datetime as dt
t = dt.datetime(2015,6,18,10,49,14,416928)
tstr = t.strftime('%M %d %Y')
print(tstr)
```

のようなコードで得られる表示形式では不正解とする．英語による月の表示名は省略したものでよい．たとえば，March は Mar でもよい．

ヒント：複数解答があるので各自創意工夫せよ．月の名前を文字列として保存したリストを用意し，`t.month` を参照することはひとつの解答方針例である．Google 検索でよい方法はないか検索することも大変有用である．たとえば，少し検索すれば月の名前を文字列として保存したリストが詰め込まれたパッケージを発見することができる．さらにがんばって探せば，実は月の名前を文字列として書式整形するフォーマットが発見できる．

この課題は Python が持つ未知の機能を自分で検索して発見する能力を磨くための練習だから，もし仮に答えがそのまま書いてあるようなページを発見したとしても友人に教えないように．上回生になると web 検索をはじめとして,「探す能力」が大変重要になってくる．ここで答えだけ友人に教えてしまうとその大事な友人が上回生になって大変苦労することになる.「探し方のコツ」を教えてあげよう．

13.2　課題 8-2

```
import copy
v=[1,2,3]
u=v
print(v)
print(u)
v[1] = -1
print(v)
print(u)
```

を実行せよ．この実行結果をよく観察し，`u=v` を実行すると，`u` と `v` は同じ実体を指している（2 つのリストが連動して値がかわってしまう）ことを確認せよ．

```
import copy
v=[1,2,3]
u=copy.copy(v)
print(v)
print(u)
v[1] = -1
print(v)
print(u)
```

を実行せよ．この実行結果をよく観察し，u=copy.copy(v) を実行すると，u と v は異なる実体を指している（u に v と同じ値をもつ別のリストが代入される）ことを確認せよ．

13.3 課題 8-3

0 以上 10 未満の整数からなるリストを作成し，v に代入せよ．v と同じ値を持つリスト w をリスト v からコピーする形で作成せよ．w を，各要素が元の 2 倍になるよう変更せよ．v と w の中身をすべて表示し，確かに異なる値を持っていることを確認せよ．

13.4 課題 8-4

以下のコードを実行せよ．

```
import copy
v=[[1,2],[3,4]]
u=copy.copy(v)
print(v)
print(u)
v[0][0]=-1
print(v)
print(u)
```

実行結果をよく観察し，u[0][0] と v[0][0] のペア，u[1][0] と v[1][0] のペアなどはそれぞれ同じ実体をあらわしているだけであることを確認せよ．リストのリストに対して copy.copy() を適用しても,「[1,2] を指すこと」ならびに「[3,4] を指すこと」のコピーが実行されるためである．

さらに，以下のコードを追加で実行せよ．

```
import copy
v=[[1,2],[3,4]]
u=copy.deepcopy(v)
print(v)
print(u)
v[0][0]=-1
print(v)
print(u)
```

実行結果をよく観察し，リストのリストに対して copy.deepcopy() を行えば，リスト内にあるリストの全ての要素に対して，値のコピーが行われ，u[0][0] と v[0][0] のペアも u[1][0] と v[1][0] のペアもそれぞれ別々の実体をもてるようになったことを確認せよ．

13.5 課題 8-5

copy.deepcopy() では，オブジェクトの中にオブジェクトがあった場合，再帰的に値のコピーを行う．ややこしいが，リストのリストのリストのコピーも同様にできる．

```
import copy
v=[[[1,2],[3,4]],[[5,6],[7,8]]]
u=copy.copy(v)
print(v)
print(u)
v[0][0][0]=-1
print(v)
print(u)
```

を実行し，課題 8-4 と同様に実体のコピーが結局できていないことを確認せよ．

```
import copy
v=[[[1,2],[3,4]],[[5,6],[7,8]]]
u=copy.deepcopy(v)
print(v)
print(u)
v[0][0][0]=-1
print(v)
print(u)
```

を実行し，再帰的に実体のコピーが行われ，u[0][0][0] と v[0][0][0] のペアや u[0][1][0] と v[0][1][0] のペアなどがそれぞれ別々の実体をもてるようになったことを確認せよ．

13.6 課題 8-6

モンテカルロ法と呼ばれる方法で円周率 π を求めてみよう．0以上1以下の乱数 x, y で定義される点 (x, y) は図 1 左で示すように，二次元平面の第一象限にある長さ 1 の正方形内の点とみなすことができる．

正方形内に大量の点を打つと（例えば１０００個），乱数の性質により，ランダムに作成した点は，図 1 右のように正方形の中に均等に分布する．

ここでは中心 0，半径 1 の扇を単に「扇形」と呼ぶことにする．乱数が一様に分布しているのであれば，扇形の中にある点と，ランダムに打った点の総数の比は，扇形の面積と正方形の面積の比とほぼ一致するはずである．

扇形の面積は $\pi/4$，正方形の面積は 1 だから，円周率の近似値を以下で求めることができる．

$$\pi \approx 4\frac{\text{正方形内に配置された扇形内に打たれた点 } (x, y) \text{ の個数}}{\text{ランダムに正方形内に打った点の総数}} \tag{16}$$

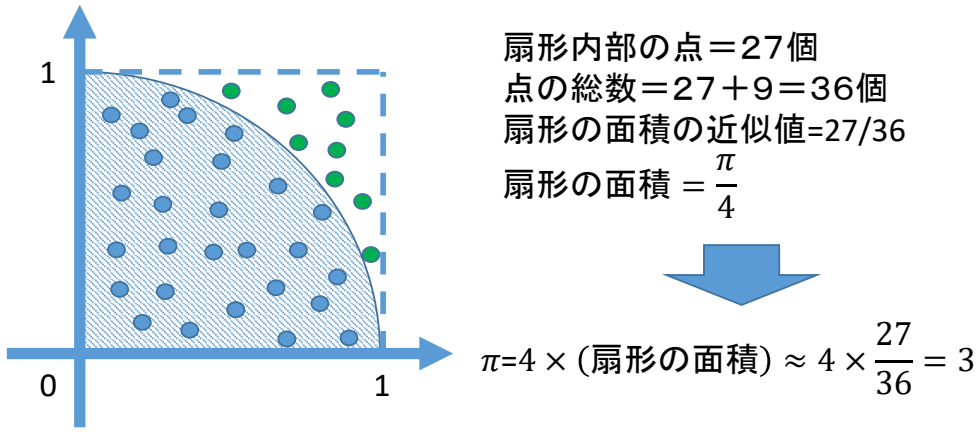

図 1: 長さ 1 の正方形内に打ち込まれたランダムな点

ただし，ここで ≈ は「ほぼ等しい」という意味である．数学記号としてはあまり厳密な用法ではないことに注意しよう．

扇形内に点 (x, y) があるための必要十分条件は，以下のとおりである．

$$x^2 + y^2 < 1 \text{ かつ } 0 \leq x \leq 1 \text{ かつ } 0 \leq y \leq 1 \tag{17}$$

したがって，円周率の近似値は以下で求められる．

$$\pi \simeq 4 \frac{x^2 + y^2 < 1 \text{ を満たす正方形内の点 } (x, y) \text{ の個数}}{\text{ランダムに正方形内に打った点の総数}} \tag{18}$$

上記の関係式を用い，10000 個のランダムな点による π の近似値を求めるプログラムを作成せよ．

14　第 9 週課題

14.1　課題 9-1

ユーザから入力された文字列から小文字のアルファベット，数字，数字という並びでできている文字列を検索して出力するプログラムを完成せよ．ただし，検索結果は Searched Result:　という文字列に続けて出力するものとする．また，複数見つかった場合はそれらの文字列を空白文字を区切り文字として全て表示し，見つからなかった場合はその旨表示すること．

プログラムの構造を以下に示す．

```
x = input("Please Input String: ")
＜この部分を埋める＞
```

想定される実行例は以下の通りである．

```
Please Input String: This model is m16.
Searched Result: m16
```

```
Please Input String: This model is m16, x76 and B18.
Searched Result: m16 x76
```

```
Please Input String: My name is Ritsumei Taro.
Searched Result: Not Found
```

14.2　課題 9-2

四則演算結果を返す関数 calc(num1，num2，symb) を作成せよ．num1, num2 は実数とし，symb は’add’, ’sub’, ’mul’, ’div’ のいずれかとする．ただし，引き算と割り算の計算順は num1-num2，num1/num2 とし，精度を 3 桁とする．

```
In [1]: calc(3.42344, 5.4543, 'add')
Out[1]: Decimal('8.88')
```

14.3　課題 9-3

Passwd という名前のクラスを生成し，文字列として与えられたパスワードが初期パスワードと一致するかどうかをハッシュを用いて確認するプログラムを作成せよ．

```
In [1]: p = Passwd('djle3Z4j')
In [2]: p.cmp('djle3Z4j')
SAME
In [3]: p.cmp('djle3Z4k')
NOT SAME
```

14.4 課題 9-4

以下のコードを入力して実行してみよう（わからないものがあれば調べよう）.

```
import numpy as np
import matplotlib.pylab as plt

x = np.linspace(-np.pi, np.pi, 201)
plt.plot(x, np.sin(x))
plt.xlabel('Angle [rad]')
plt.ylabel('sin(x)')
plt.axis('tight')
# plt.show()
```

末尾の `plt.show()` は Google Colaboratory で実行する際には不要であるのでコメントとしている.

内容が理解できたらプログラムを修正し，`x` の範囲を $-2\pi \sim 2\pi$ とし $2\sin(x) + \cos(2x/3)$ のグラフを描け (定義域を $[-2\pi, 2\pi]$ とした関数 $2\sin(x) + \cos(2x/3)$ のグラフを描け).

14.5 課題 9-5

numpy モジュールを用いると行列演算が容易にできる．以下の例を参考にその使い方を理解せよ.

```
In [1]: import numpy as np
In [2]: mat1=np.matrix([[1, -2, 3], [2, -1, 1], [1, 3, -5]])     #行列生成
In [3]: mat1
Out[3]: matrix([[ 1, -2,  3],
                [ 2, -1,  1],
                [ 1,  3, -5]])
In [4]: mat1 + mat1      # 行列の足し算
Out[4]: matrix([[  2,  -4,   6],
                [  4,  -2,   2],
                [  2,   6, -10]])
In [5]: mat1 - mat1      # 行列の引き算
Out[5]: matrix([[0, 0, 0],
                [0, 0, 0],
                [0, 0, 0]])
In [6]: mat1 * mat1      # 行列の掛け算
Out[6]: matrix([[  0,   9, -14],
                [  1,   0,   0],
                [  2, -20,  31]])
In [7]: mat1.getI( )     # 逆行列
Out[7]: matrix([[  2.,  -1.,   1.],
                [ 11.,  -8.,   5.],
                [  7.,  -5.,   3.]])
```

以上を用いて次の連立方程式の解を求めるプログラムを作成せよ.

$$x - 2y + 3z = 5 \tag{19}$$

$$2x - y + z = 6 \tag{20}$$

$$x + 3y - 5z = 2 \tag{21}$$

14.6 課題 9-6

式を表す文字列を入力すると数値の部分と演算子の部分で分けて以下の例のように出力するプログラムを作成せよ．演算子としては四則演算についての+, -, *, / についてだけ，式としては整数とこれらの演算子のみからなるものについて動作すれば良いものとする.

```
Input Equation: 4+3*3-2
[4, 3, 3, 2]
['+', '*', '-']
```