

ネットワーク及び演習

第 1 回 イントロと Python 入門

佐野 彰

本日の内容

1. 科目の概要（15 分）

- 到達目標、実施方法、成績評価、質問方法

2. Python の実行環境（15 分）

- インタプリタ、対話モード、スクリプト実行、ノートブック

3. 演習課題の進め方（15 分）

- paiza ラーニング、aiJudge（オンラインジャッジ）

4. 教材ノートブックで Python 入門（30 分）

- C 言語との対比（インタプリタ、動的型付け、インデント構文）
- `python_tutorial.py` を実際に動かしながら

1. 科目の概要

この科目の到達目標

Network（を理解する）

- 計算機ネットワークの**基本概念**を説明できる
 - クライアント／サーバ、IP アドレスとポート、TCP、HTTP、スレッド
- ネットワークを介して通信する**プログラムの仕組み**を、実際に動かして理解する

Python（を使って）

- Python を利用した**簡単なプログラミング**ができる
 - 変数・制御構文・関数・リスト・文字列
 - `socket` , `threading` などの標準ライブラリ
- 「動くコードを自分で書いて確かめる」習慣をつける

****Python を使って、Network を理解する** 科目です。Python 自体を深く学ぶ科目ではありません。**

- 教職課程（情報）の必修科目でもある

なぜ Python なのか？ — 同じ「echo クライアント」を C と比べる

```
# Python: 11 行
import socket

HOST = 'localhost' # サーバのホスト名
PORT = 50007       # サーバのポート番号

with socket.socket(socket.AF_INET,
                    socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    s.sendall(b'Hello, world')
    data = s.recv(1024)
    print('Received', repr(data))
```

```
/* C: 40 行超 (一部省略) */
/* #include <sys/socket.h> など 6 行省略 */
int main() {
    int sock;
    struct sockaddr_in addr;
    if ((sock = socket(AF_INET, SOCK_STREAM,
                       IPPROTO_TCP)) < 0) exit(1);
    memset(&addr, 0, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    addr.sin_port = htons(50007);
    if (connect(sock, (struct sockaddr *)&addr,
                sizeof(addr)) < 0) exit(2);
    /* send / recv / close ... */
}
```

- Python の方が圧倒的に簡潔 → ネットワークの仕組みそのものに集中できる
- C 固有の構造体・バイト順・メモリ管理に煩わされない

科目の位置づけ — 上下 1 階層分の視野を持つ

レベル	内容	対応する科目
1. 基盤	計算機の仕組み（ハードウェア、CPU、OS）	情報処理システムⅠ・Ⅱ
2. コーディング	言語の構文、変数・関数・制御構造	プログラミングⅠ・Ⅱ・Ⅲ
3. 計算的思考	アルゴリズム、データ構造	アルゴリズム・データ構造
4. ソフトウェア設計	部品の組み合わせ、環境構築、開発ツール	ネットワーク、オブジェクト指向、グラフィックス
5. ユーザー・社会	UI/UX、倫理的・持続的な意義	—

- この科目はレベル 2 → 4 への橋渡し。
「言語が書ける」から「通信する部品を組み合わせで動かせる」へ
- 生成 AI はレベル 2～3 を肩代わりしつつある。
だからこそ上下を見渡して設計できる力が問われる

科目の実施方法

毎回の流れ（金曜 4・5 講時）

- 4 講時: 講義パート（約 1 時間、Teams でリアルタイム配信・録画あり）
- 講義パート終了後: 前回内容の小テスト（10 分）
- 5 講時: 実習（TA 対応）

課題の種類

- paiza ラーニングの指定講座を受講
- aiJudge（オンラインジャッジ）でプログラミング課題を提出・自動採点
- 受講修了証などのスクリーンショットも aiJudgeに提出

提出期限

- 翌週の講義日（金曜）の 0:00 締切 — つまり前日（木曜）中に提出する

受講方法

- 4・5 講時とも 1-608 実習室を利用できる
- 講義パート（4 講時）は教員の自室から Teams で配信する。録画も閲覧可
 - 1-608 で視聴しても、自宅から視聴してもよい
- 5 講時の TA サポートは 1-608 でのみ受けられる
 - オンライン参加では TA のサポートを受けられない
- 小テストは 1-608 でのみ受験できる — オンラインでは受けられない

質問方法

- 科目内容の質問は Teams 「★ 質問用チャンネル」 へ
 - 他の受講者にも役立つ。時間外も歓迎（即答は期待しない）
- チャットやメールでの個別質問には原則応じない
- 演習時間中に対面参加なら TA に直接
- 投稿には 参加形態（対面なら座席番号）・PC 環境・やったこと・エラーメッセージ を書く
- 良い質問・悪い質問の具体例は科目ページ「この科目での質問方法」を参照

成績評価

配点

項目	配点
演習課題（ex1～ex7）	60 点
小テスト	40 点
└ 各回の小テスト（第 2～6 回の 5 回）	各 1 点・計 5 点
└ 最終回のまとめの小テスト（11/13）	35 点（各回小テスト 3 回以上で受験可）
任意提出レポート	加点（詳細は後日）

生成 AI の利用

- ・ 演習・デバッグでの利用は**推奨**（コードの解説、エラーの修正、24 時間対応）
- ・ ただし **小テストでの利用は禁止**（Web 検索は可）。利用が判明した場合は不合格
- ・ 「自分で Python の簡単なコードを書ける」ことがこの科目の単位取得の条件

本日の目標

- Python の実行環境を構築する（事前課題 ex0 — まだの人は今日中に）
- 対話モード・スクリプト実行・ノートブックの **3 つの実行方法**を使い分けられる
- paiza ラーニングを使えるようになる
- **aiJudge** にログインして課題を提出できるようになる
- 教材ノートブックをブラウザで開き、セルを実行・書き換えできる
- Python の特徴を **C との対比**で説明できる
 - インタプリタ、動的型付け、インデント構文

2. Python の実行環境

インタプリタ、対話モード、スクリプト、ノートブック

コンパイラ言語とインタプリタ言語

C — コンパイラ言語



- 実行前に機械語に翻訳する
- 翻訳済みなので実行は速い
- 型のエラーはコンパイル時に見つかる

Python — インタプリタ言語



- コンパイル不要、書いてすぐ動かせる
- 実行は C より遅い（この科目では問題にならない）
- エラーは実行してその行に到達したときに
出る
 - → 動かして初めて発覚することがある

Python の 3 つの実行方法

① 対話モード (REPL)

```
$ python
>>> 1 + 2
3
>>> name = 'sano'
>>> print('Hello', name)
Hello sano
>>> exit
```

- 1 行ずつ入力して即実行
- ちょっと試すのに最適
- `>>>` がプロンプト

② スクリプト実行

hello.py

```
name = 'sano'
print('Hello', name)
```

```
$ python hello.py
Hello sano
```

- ファイルに書いてまとめて実行
- 課題の提出はこの形式
- aiJudge もこの形で採点する

③ ノートブック

- ブラウザ上で「セル」ごとに実行
- 説明文・数式・コード・結果を一体化
- 教材はこの形式 (marimo)
- 科目ページから開けばインストール不要
- 中身はふつうの `.py` ファイル

macOS では `python3`、Windows では `python` または `py`。`python -V` でバージョン (3.x) を確認。

ノートブックの使い方 — セルと変数

- **セル**: ノートブックの最小単位。テキストセル (Markdown) とコードセル
- コードセルは `Shift + Enter` で実行。結果はセルの直下に表示
- セルは名前空間を共有する — あるセルで作った変数は他のセルで使える
- セル 1 の `a` を書き換えると、`a` を使うセル 2 は自動で再実行される

```
# セル 1  
a = 1234  
b = 5678
```

```
# セル 2  
b / a      # → 4.6013...
```

ノートブックの注意点

ルール: 同じ変数は 1 つのセルでしか定義できない

別のセルで `a = ...` と定義し直すとエラーになる。marimo は変数の依存関係を追跡して関係するセルを自動で再実行するため、「どのセルを何番目に実行したか」で結果が変わることはない。

- 変数を変えたいときは、定義しているセルを書き換える
- 教材ノートブックは科目ページからブラウザで開ける
 - `jupyter.py` (ノートブックの使い方)
 - `python_tutorial.py` (Python 入門)

3. 演習課題の進め方

演習課題 ex1

課題 1: paiza ラーニングの認定証

1. 科目ページ「paiza ラーニングでクーポンコードを使う」に従い、学校フリーパスを登録
2. 講座 「Python 体験編 1: Python をはじめよう」(全 15 チャプター) を修了
3. 認定証の画面キャプチャを aiJudge の課題 「認定証提出」 に提出
 - 認定証のニックネームに学籍番号を含めること

課題 2: aiJudge のプログラミング課題

1. aiJudge に 大学アカウント (全学認証) でログイン。個別の ID 配布はない
2. ex1 の 3 問 (`hello.py` , `input.py` , `nhello.py`) ですべて満点を取る

aiJudge の使い方

- <https://judge.math.ryukoku.ac.jp/>
 - 使い方ガイド: <https://sanoakr.github.io/aijudge/student/>
- 問題ごとに Python ファイル (`.py`) を提出
- 提出直後に自動採点。テストケースごとの結果と理由が表示される
- 満点になるまで何度でも提出可
- 採点は「標準入力 → 標準出力」の一致で判定
 - 余計な出力 (プロンプトなど) は不正解になる
- 自動採点のあとも TA・教員がコードに目を通す。他人のコードを写さない

締切: 9/25 (金) 0:00 — 9/24 (木) 中に提出

4. 教材ノートブックで Python 入門

ここからはスライドではなく、ノートブックを動かしながら進める

Python の 3 つの特徴（C と比べて）

	C 言語	Python
① 実行方法	コンパイルして実行	インタプリタが直接実行
② 変数の型	宣言が必要（ <code>int a;</code> ）	動的型付け — 代入した値で決まる
③ ブロック構文	<code>{ }</code> で囲む（インデントは飾り）	インデントがブロックを表す

この 3 つを、教材ノートブック `python_tutorial.py` で動かしながら確認する。

細かい違い（文末の `;` が不要、コメントは `#`、出力は `print()` など）は出てきたときに説明する。
ほかにもガベージコレクション、オブジェクト指向など違いは多いが、この科目ではこの 3 つを押さえれば十分。

教材ノートブック `python_tutorial.py`

- 科目ページ「教材ノートブック」から Python 入門 を開く（ブラウザで動く。インストール不要）
- 値を書き換えて再実行し、結果がどう変わるか自分で試す
- 変数・`print`・`input`・条件分岐・繰り返し・リストは **paiza ラーニング**で自習する
 - 今日の課題 ex1 で使う `print()`・`input()`・`int()` は課題ページに説明がある

先に `jupyter.py`（ノートブックとは）を開くと、セルの操作と `marimo` のルールが分かる。

次回予告と今日やること

次回（9/25）：計算機ネットワークの基礎、Python のデータ型

- インターネットの仕組み: IP アドレス、ポート、TCP/UDP
- Python の基本（変数・入出力・制御構文・文字列・リスト）の概要
- 教材: paiza 「新・Python入門編」、ノートブック `python_string-list.py`

今日の 5 講時（実習）

1. 事前課題 ex0 が済んでいない人は環境構築（TA に相談）
2. aiJudge にログインできることを確認
3. 教材ノートブック `python_tutorial.py` を一通り実行
4. paiza と aiJudge の課題に着手

質問は Teams 「★ 質問用チャネル」 か、教室の TA へ。