

一般教育演習

プログラミングで問題を解く：

集計から人工知能まで

瀧川 一学

工学部 情報理工学コース

今日のお題：プログラミング事始

- 前回の復習とふりかえり
- コンピュータの仕組みとプログラム、社会における役割
- プログラミング言語とは？
- Pythonとは？
- Lightbotをやってみる / Scratchをやってみる
- (時間があれば) セットアップしたPython環境を試す
- ミニレポートタイム(15分)

注意

- 教室が変わるので注意してください。

4/19 (木) ・ 4/26 (木) ・ 情報教育館 B (情報教育館 2階)

5/10 (木) ~ 8/2 (木) ・ E209教室

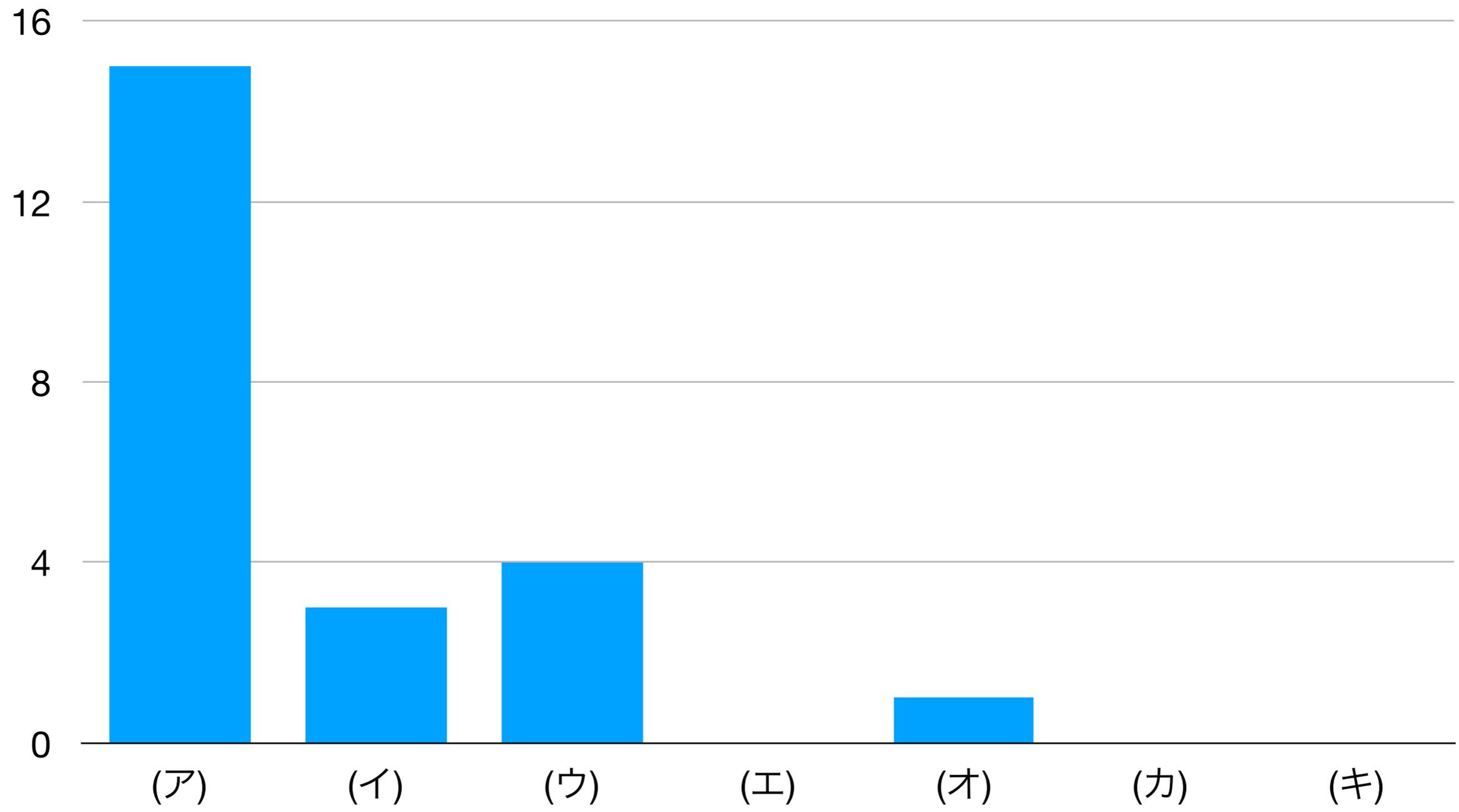
- 8G以上の容量の「USBメモリ」を持参してください。

授業の情報

<http://art.ist.hokudai.ac.jp/~takigawa/prog/>

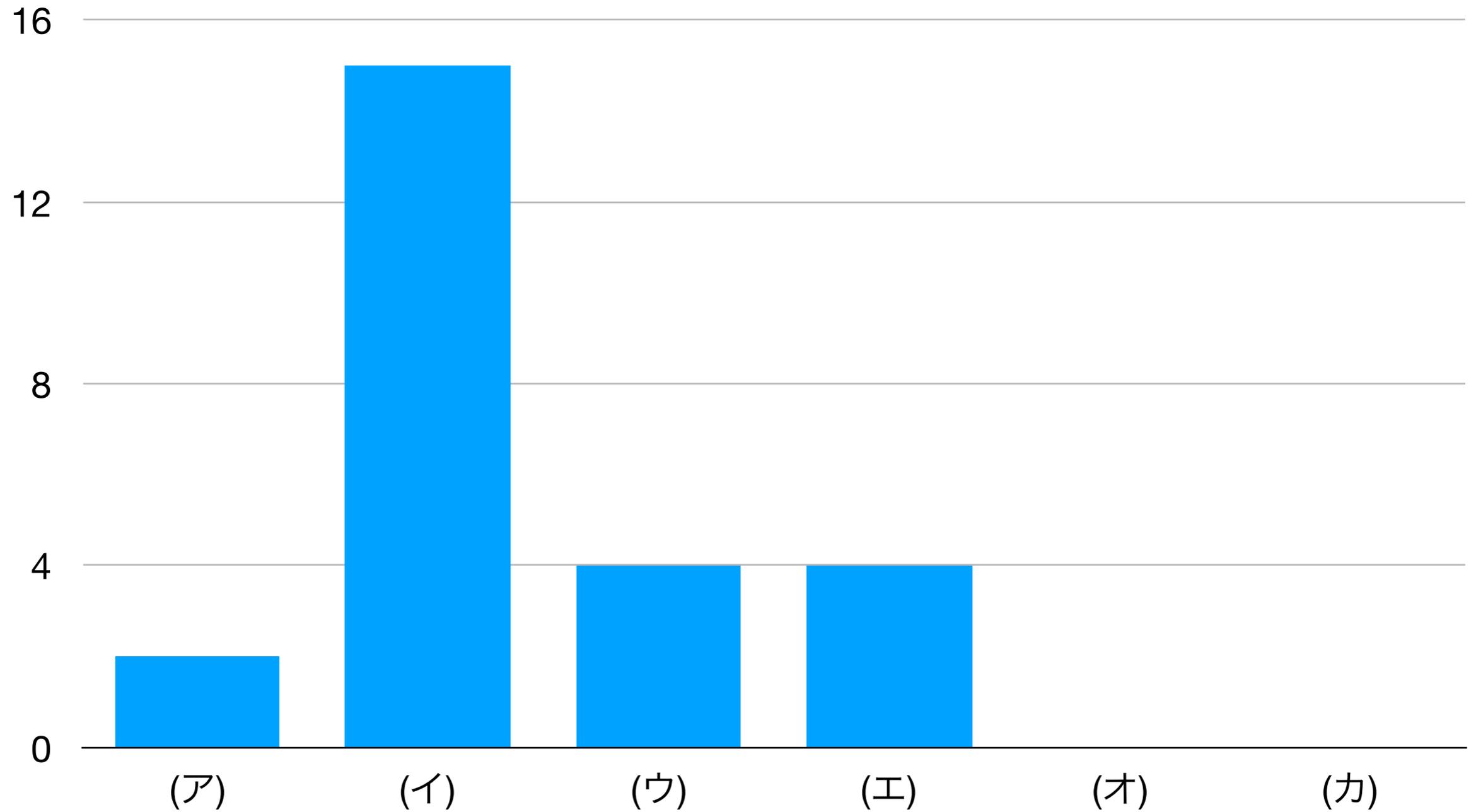
プログラミングについて

(ア)全く経験なし (イ)学校でやった程度 (ウ)少しやってみたことがあるがほぼ覚えてない
(エ)たしなむ程度 (オ)独学でかなりやっけてしまっている (カ)専門教育を受けた (キ)自由自在



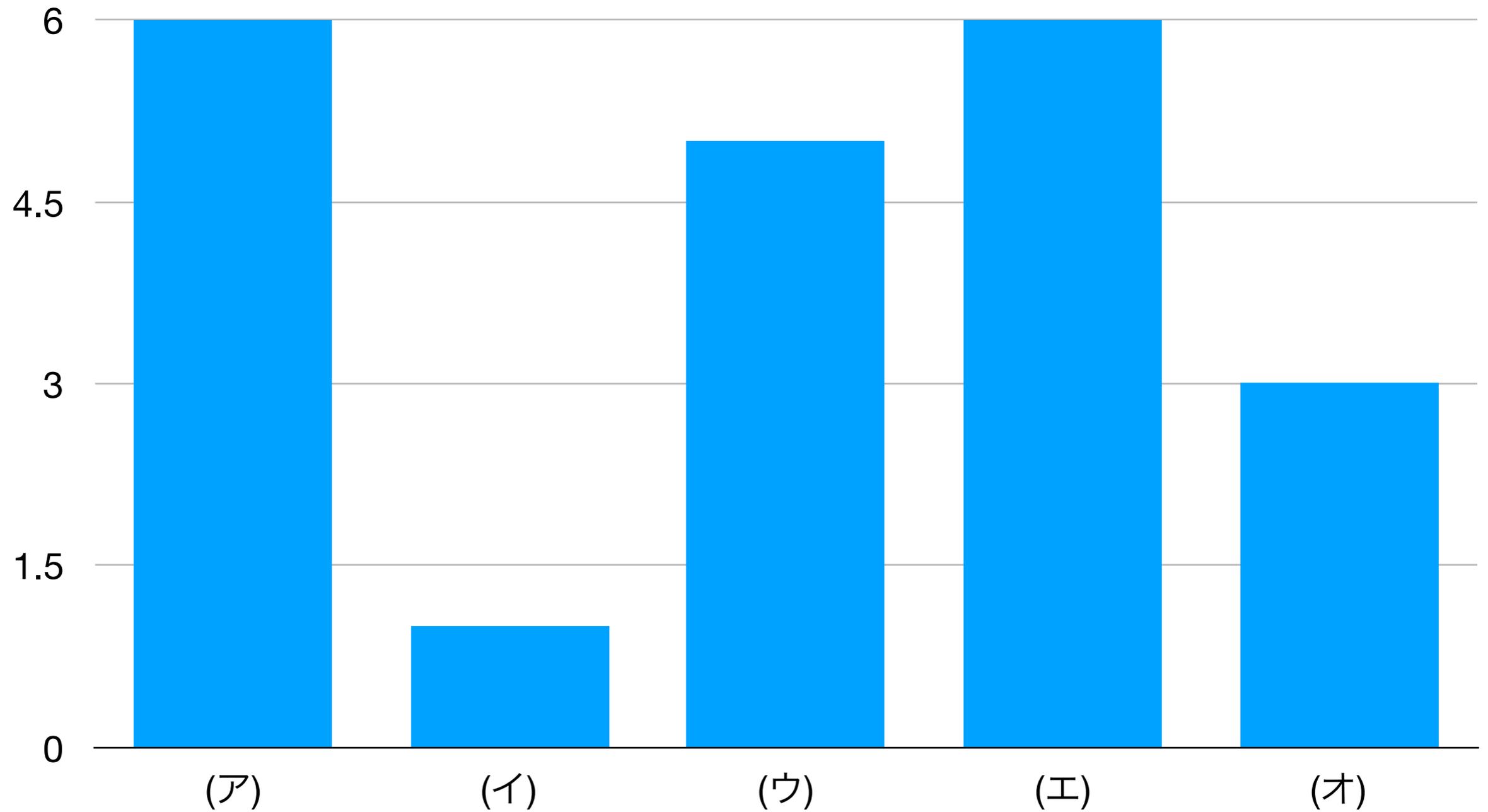
パソコンについて

(ア)ほとんど分からない (イ)ネットやメールやゲームはできる (ウ)そこそこ普段から使う
(エ)5年以上使っている (オ)UNIXを使ったことがある(カ)控えめに言っても達人とは私の事



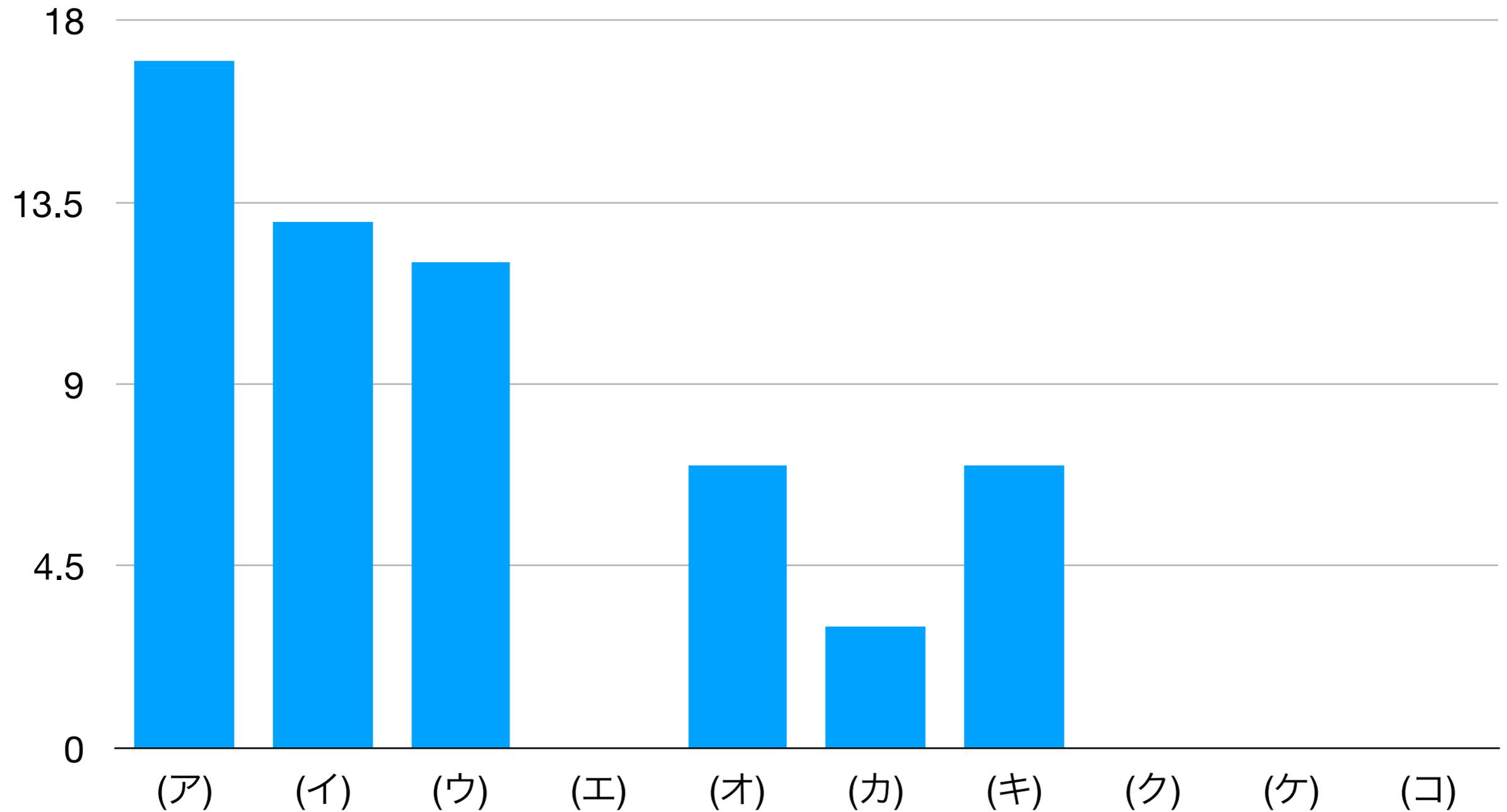
数学(もしくは数理的なこと)について

(ア)得意だ (イ)苦手嫌い (ウ)苦手だが好き (エ)ニュートラル (オ)これから頑張る予定



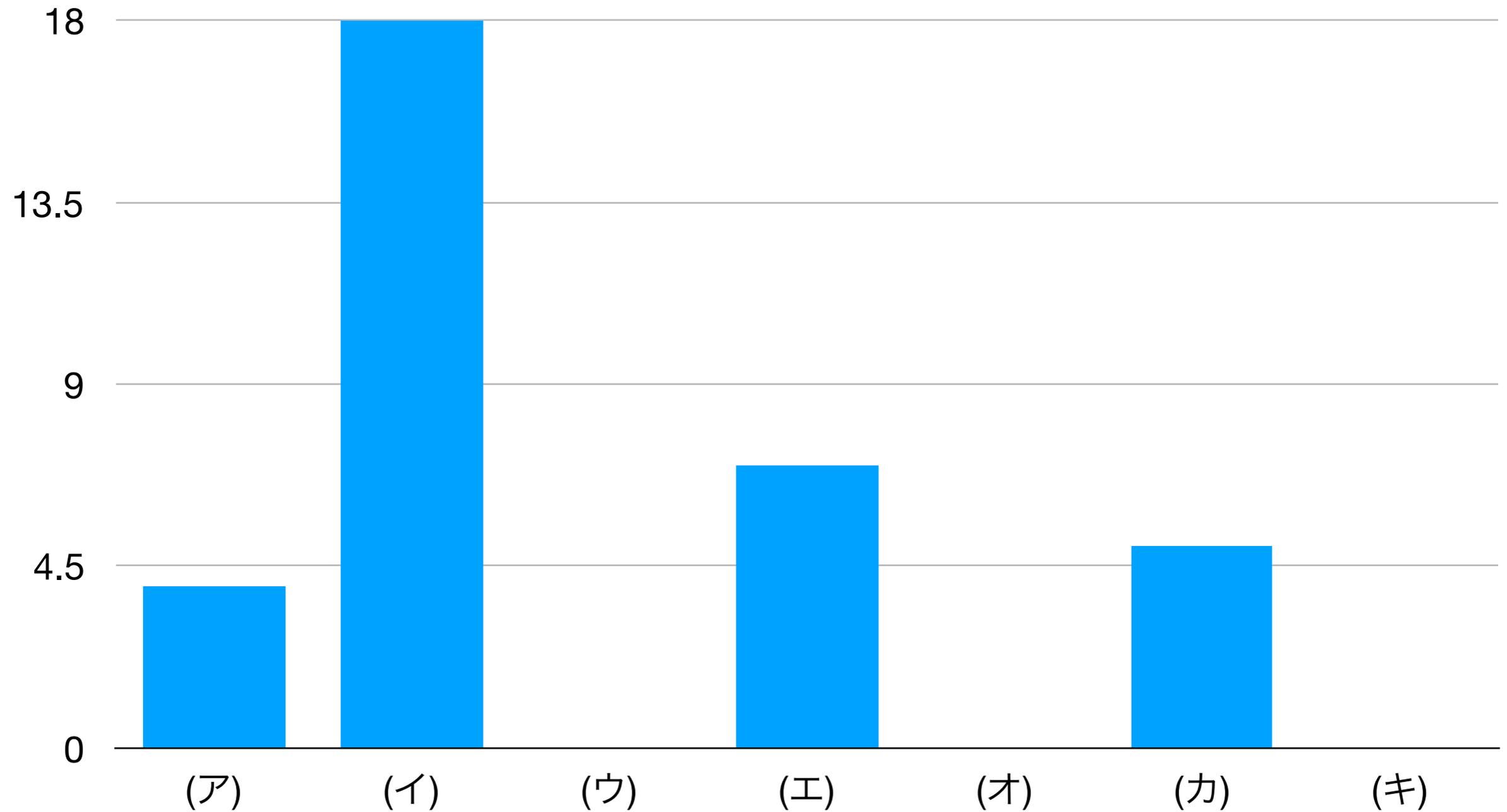
この授業を履修しようと思った理由

(ア)楽しそう (イ)役に立ちそう (ウ)情報系に進学希望 (エ)非情報系だが使いそうなところへ進学希望 (オ)将来プログラミングが関係する仕事に就きたい (カ)データサイエンスに興味がある (キ)人工知能に興味がある (ク)単位取得が楽そう (ケ)単位取得が厳しそう (コ)何となく



正直今の心境は？

(ア)全然知らないんだけど大丈夫だろうかと不安 (イ)まだよく分からないが少しワクワク
(ウ)授業の範囲で一通り知りたい (エ)授業外でも色々やってみようかな (オ)しぶしぶ出席
(カ)プログラミング王に俺はなる！ (キ)その他



先週のお題：プログラミング事始

- 先週のガイダンス内容の要点の確認
- 自己紹介タイム
- パソコン環境の準備
- プログラミングとは？

理屈→実践ではなく、逆(実践→理屈)でやってみよう！

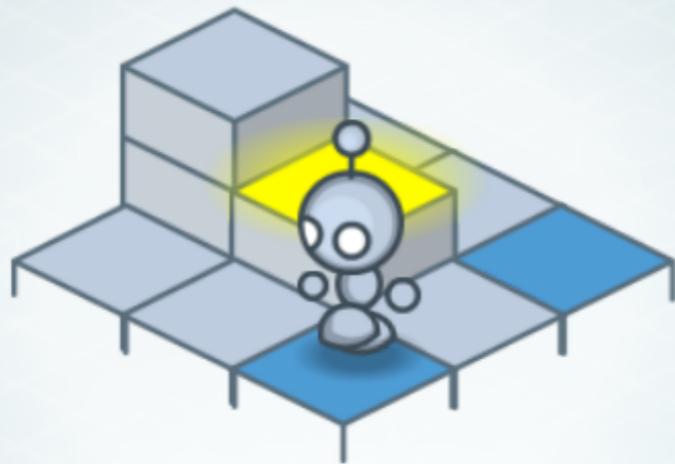
ということで細かいことは後でやることにして

プログラミングしてみよう！(lightbot)

- ミニレポートタイム(15分)



1-6

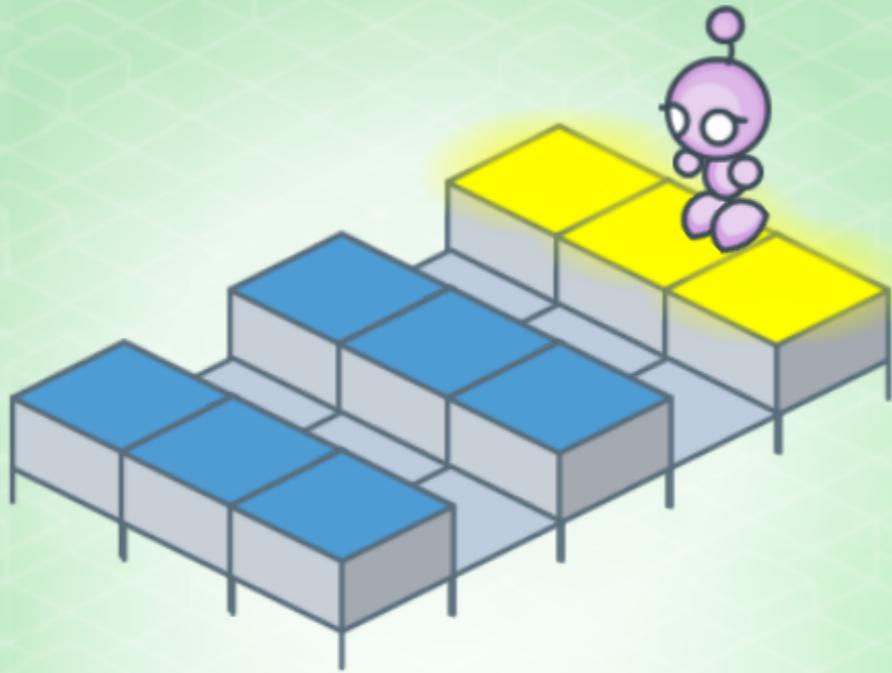


MAIN





2-3



MAIN



PROC1

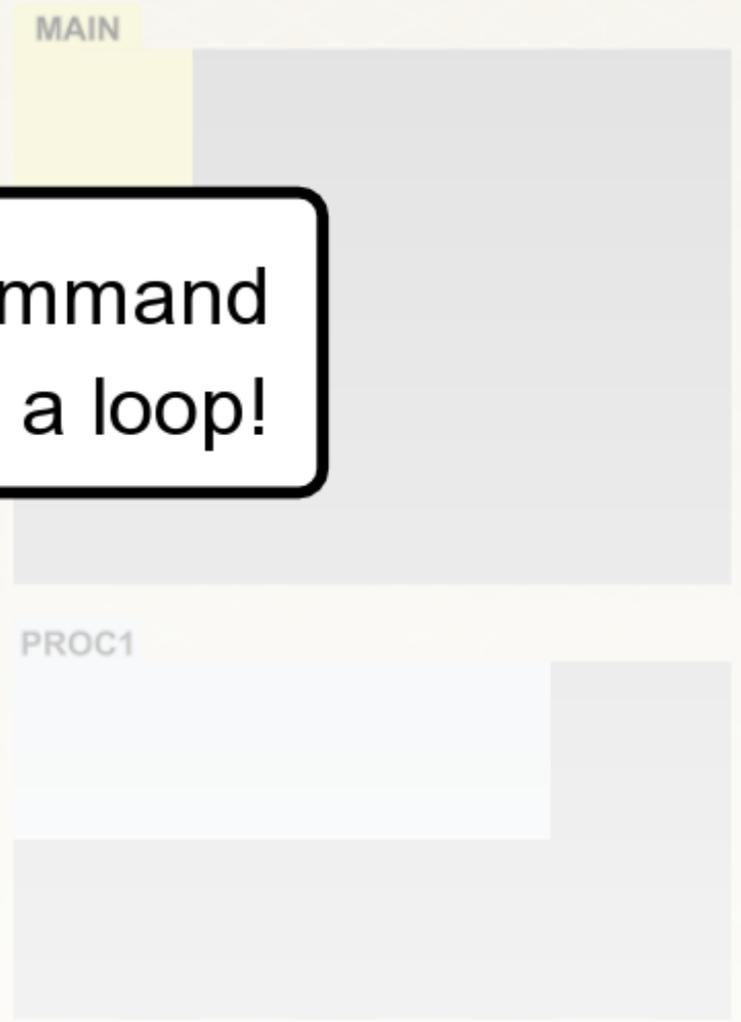
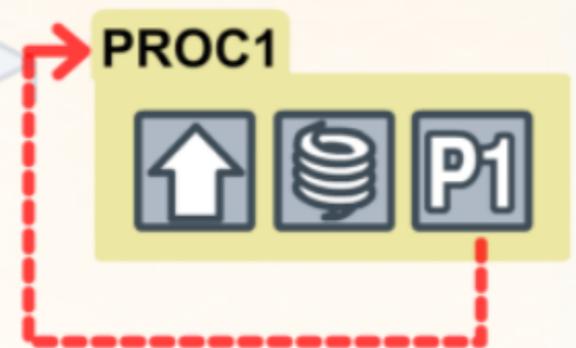




4-1

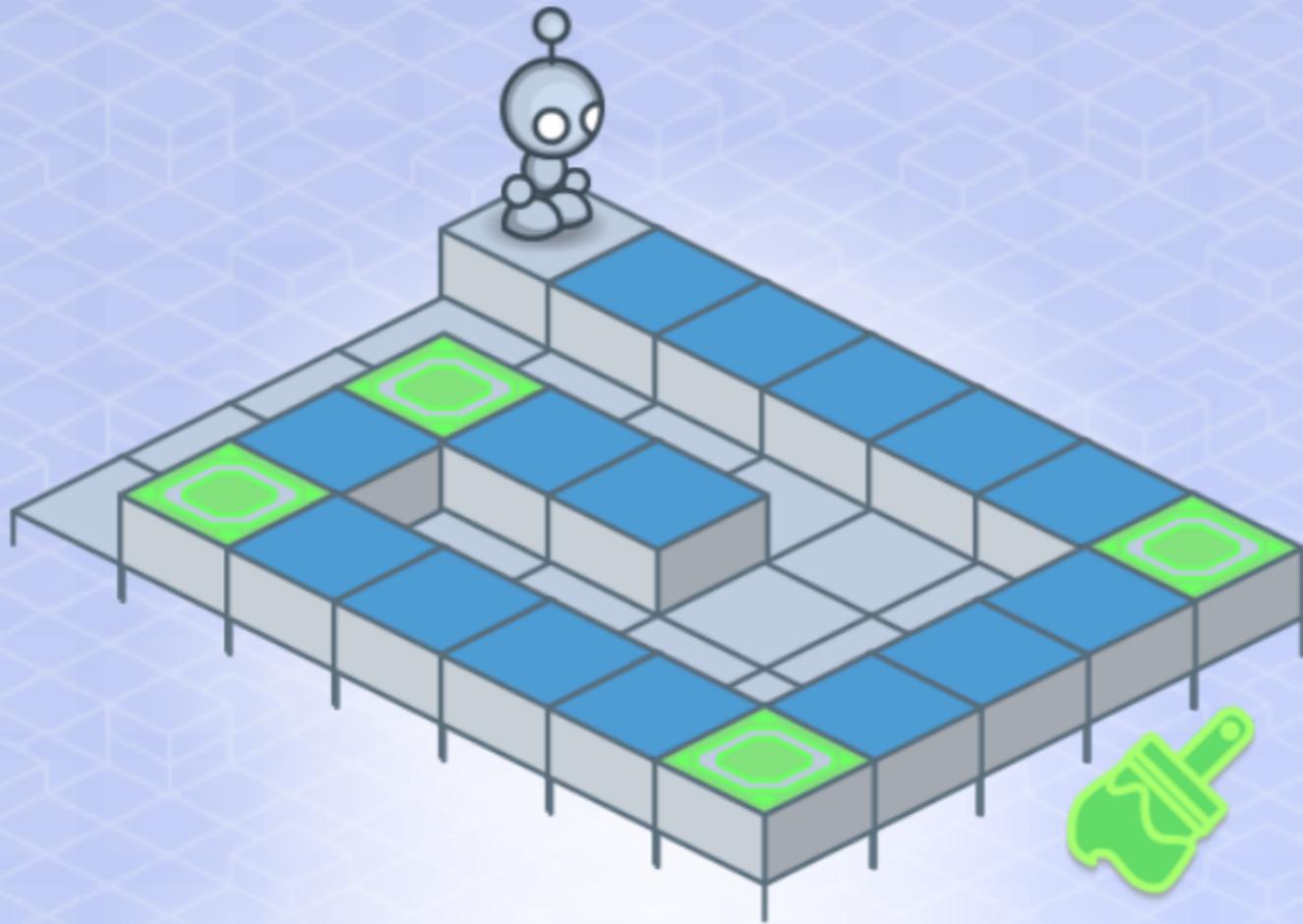


You can use the **P1** command inside **PROC1** to make a loop!





5-1



MAIN



PROC1



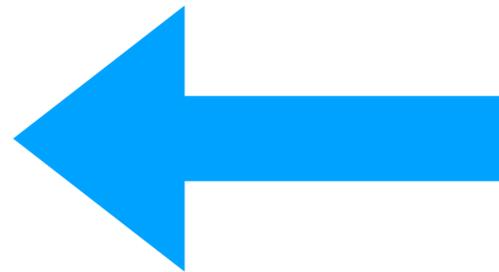
ロボが解するコマンド(命令)を

順番に伝える

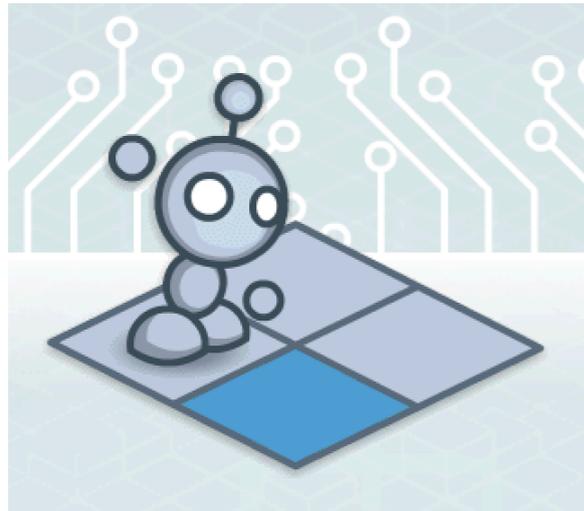


プログラミング
= プログラムする
= プログラムを作る

コンサートの「プログラム」
新人研修の「プログラム」
ロボの動作の「プログラム」
:



ロボ



特定のコマンド(命令)

しか解さない



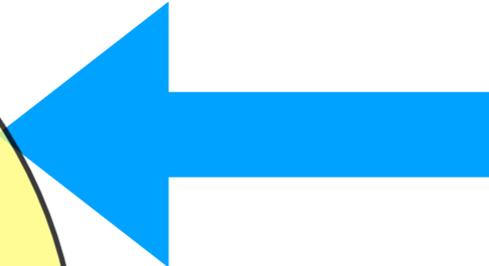
ロボが解するコマンド(命令)を
順番に伝える

ロボ

どういう
コマンドを
使える?
しか解さない
(知識の問題)

コマンドをどう
効率的に並べる?
(言語・表現の問題)

コンサートの「プログラム」
新人研修の「プログラム」
ロボの動作の「プログラム」
:

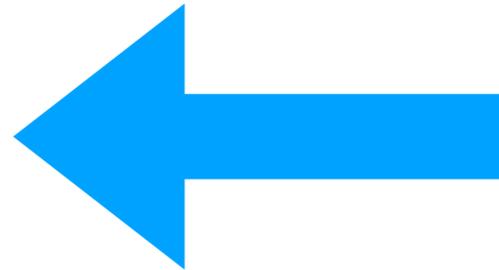


コンピュータが解するコマン

ド(命令)を順番に伝える



プログラミング
= プログラムする
= プログラムを作る



"コンピュータ"

特定のコマンド(命令)

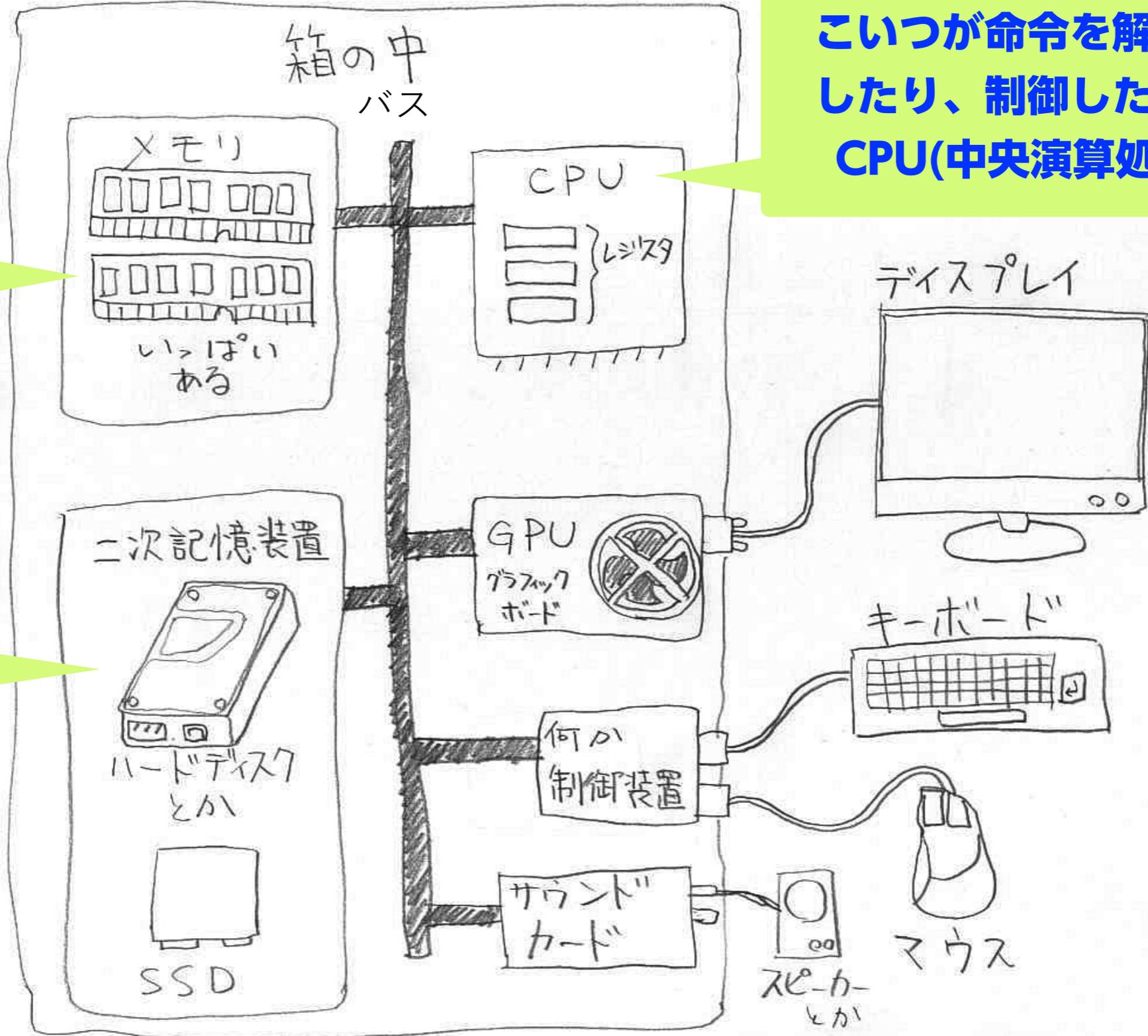
しか解さない



コンピュータの中身って???

コンピュータが動いている間、データを記憶しておく場所。
主記憶装置

電源が切れてもデータを記憶しておける。メモリより容量が大きい。



こいつが命令を解して演算したり、制御したりする。
CPU(中央演算処理装置)

バラしてみよう...

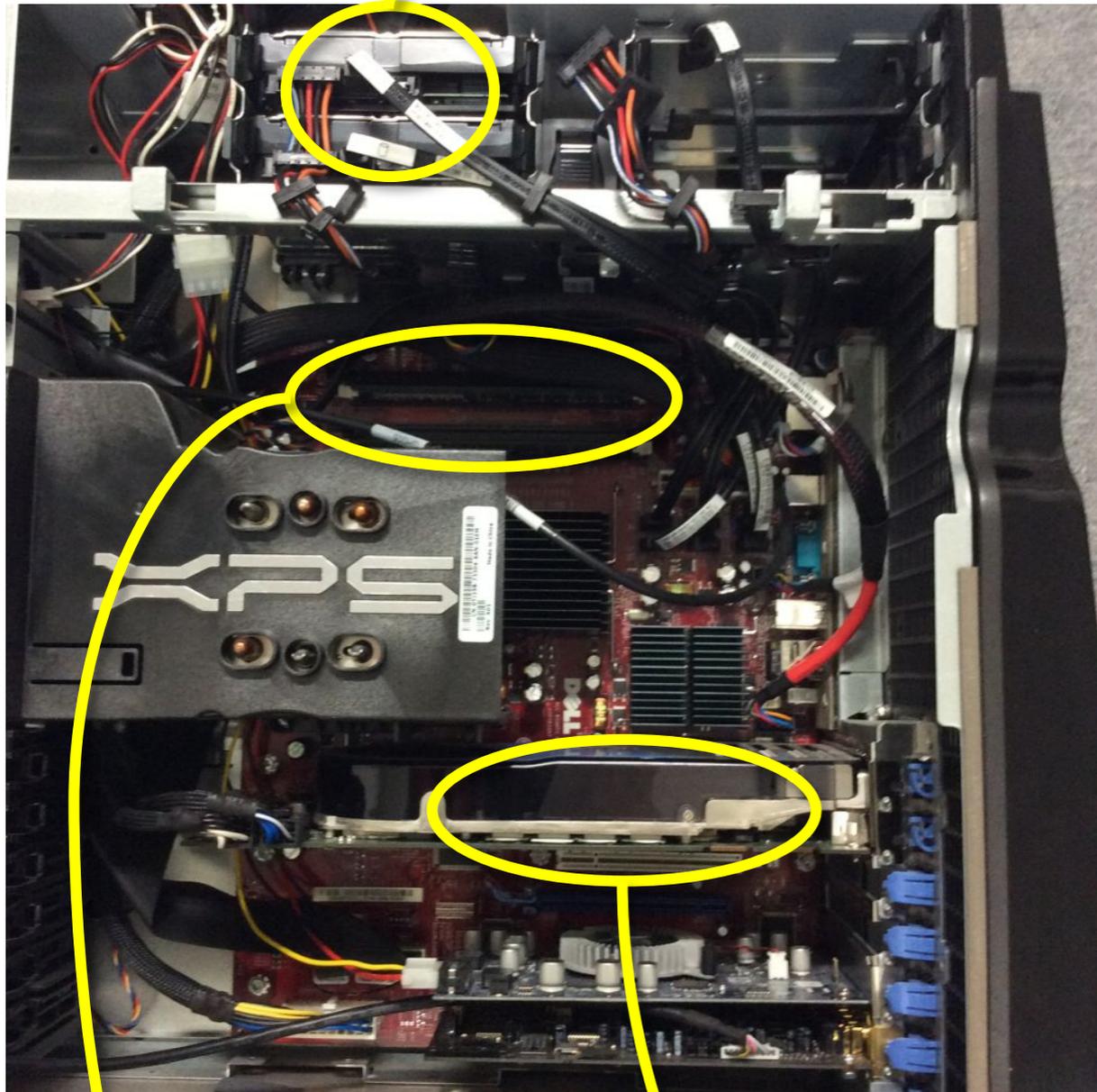


外面

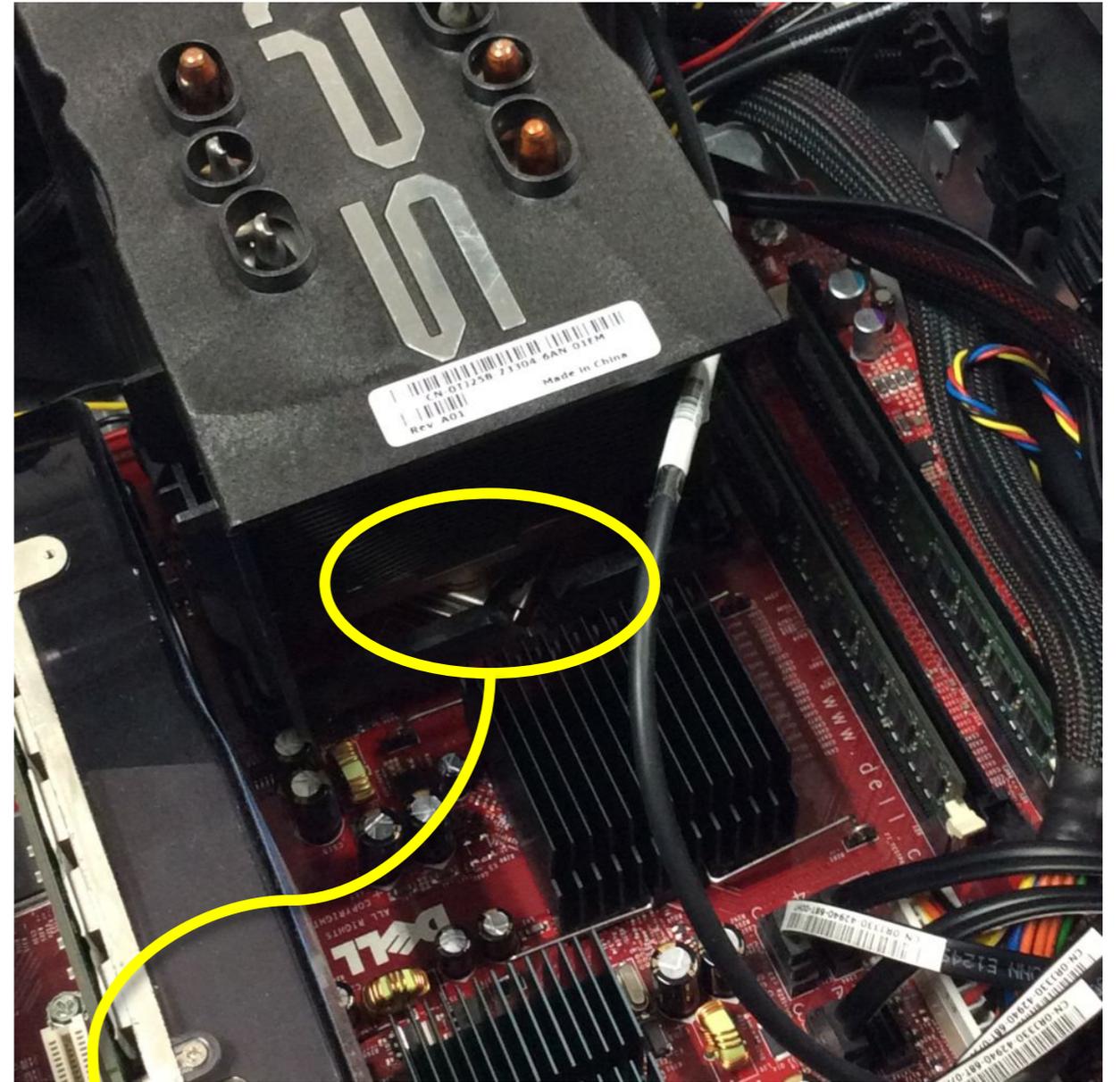


カバーを開けたところ

HDD (ハードディスク)



メモリ



CPU (中央処理装置)

GPU (グラフィックボード)



← グラフィックボード

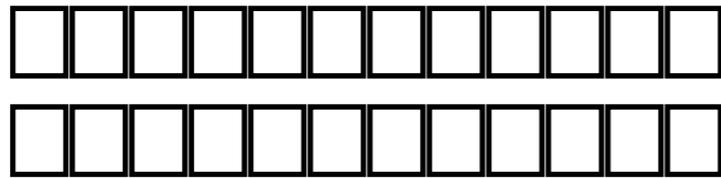
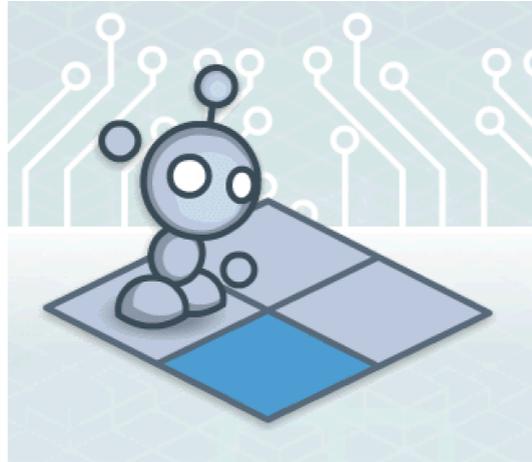


← ハードディスク



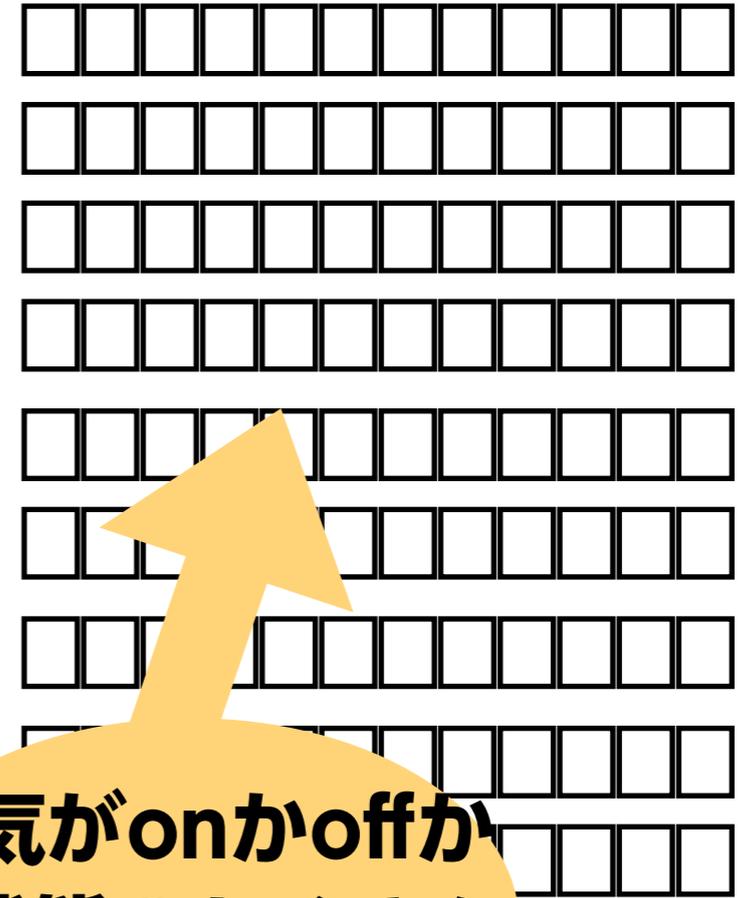
← メモリ

CPU(中央演算処理装置)



**レジスタ(作業用の
データ保存場所)**

メモリ(主記憶装置)

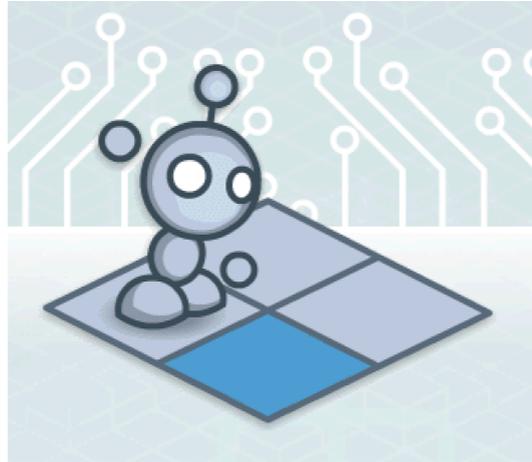


**電気がonかoffか
の状態のたくさん
の素子**

**そのほかの周辺機器(ハードディスク、USB、キーボード、
マウス、ディスプレイ、スピーカー、etc)**

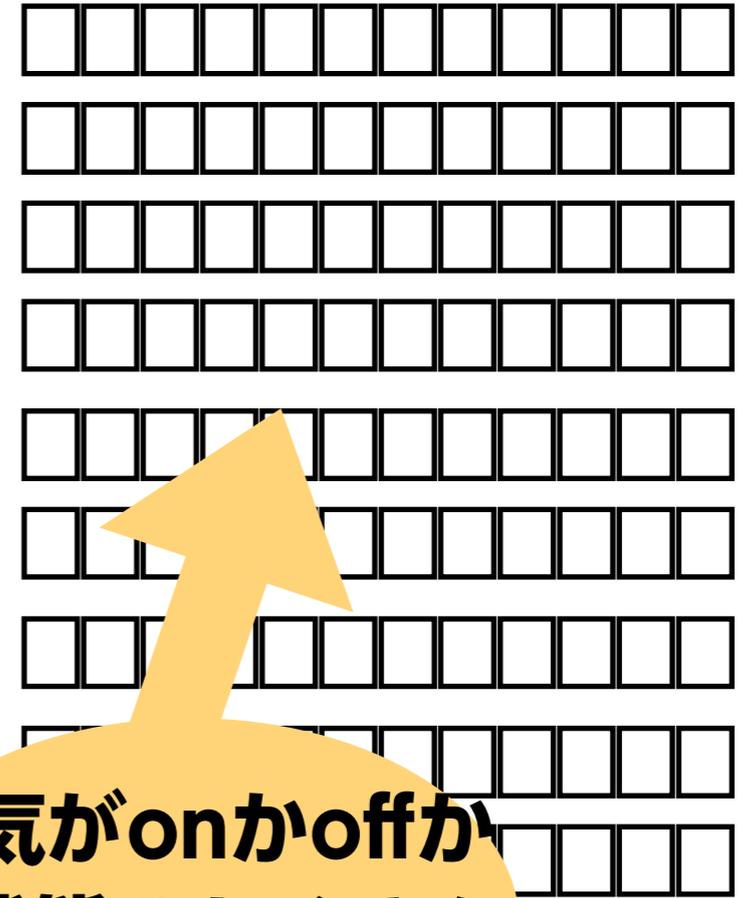
「"0"と"1"のデジタル」とかいうけど

CPU(中央演算処理装置)



レジスタ(作業用の データ保存場所)

メモリ(主記憶装置)

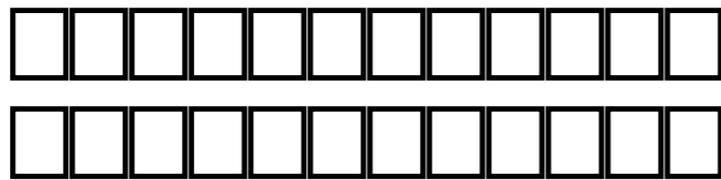
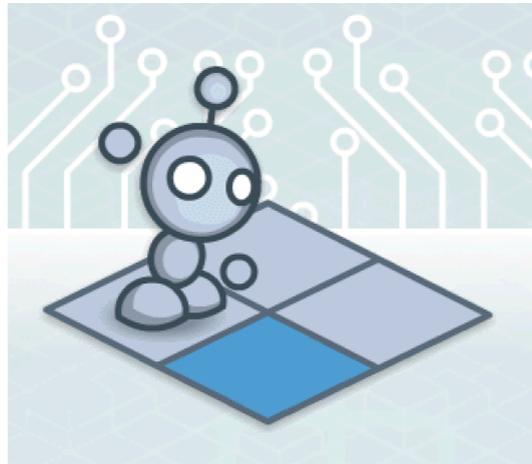


電気がonかoffか
の状態のたぐさの
素子

on=1 off=0

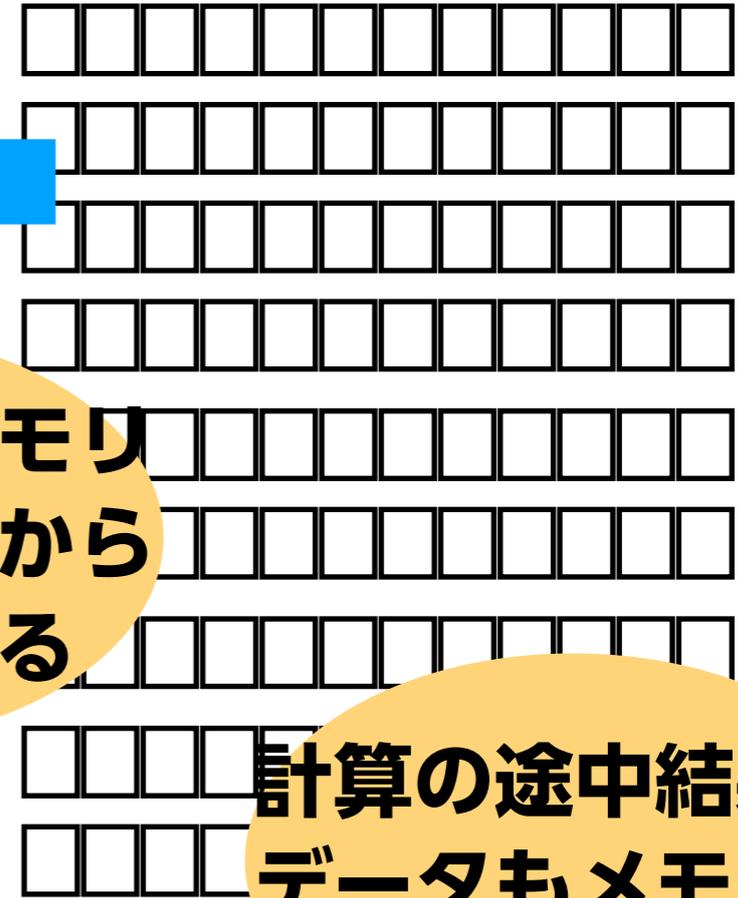
電気が流れたり流れなかったりを超高速で切り替えるため「半導体
(電気が流れたり流れなかったりする物質)」でできている

CPU(中央演算処理装置)



レジスタ(作業用の
データ保存場所)

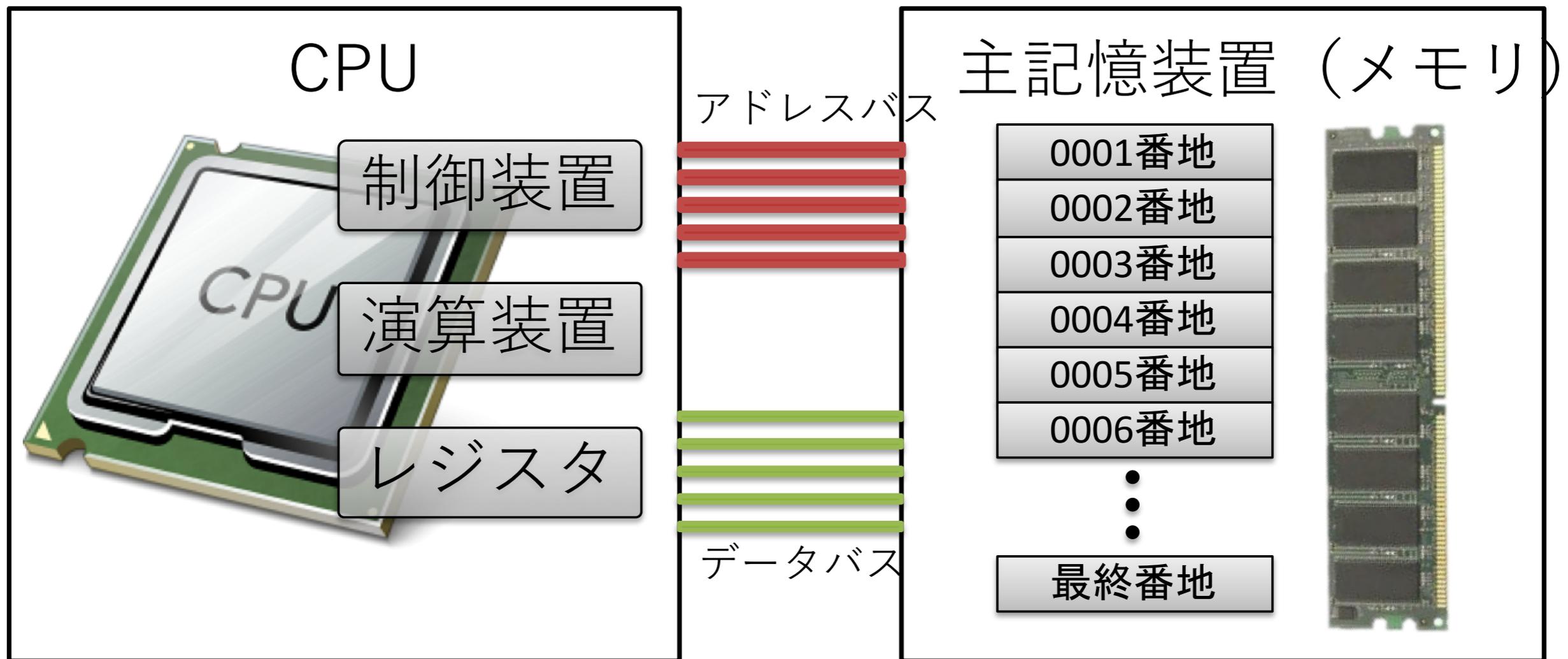
メモリ(主記憶装置)



コマンドはメモリ
に置いてそこから
読み出される

計算の途中結果や
データもメモリや
レジスタにおく

そのほかの周辺機器(ハードディスク、USB、キーボード、
マウス、ディスプレイ、スピーカー、etc)



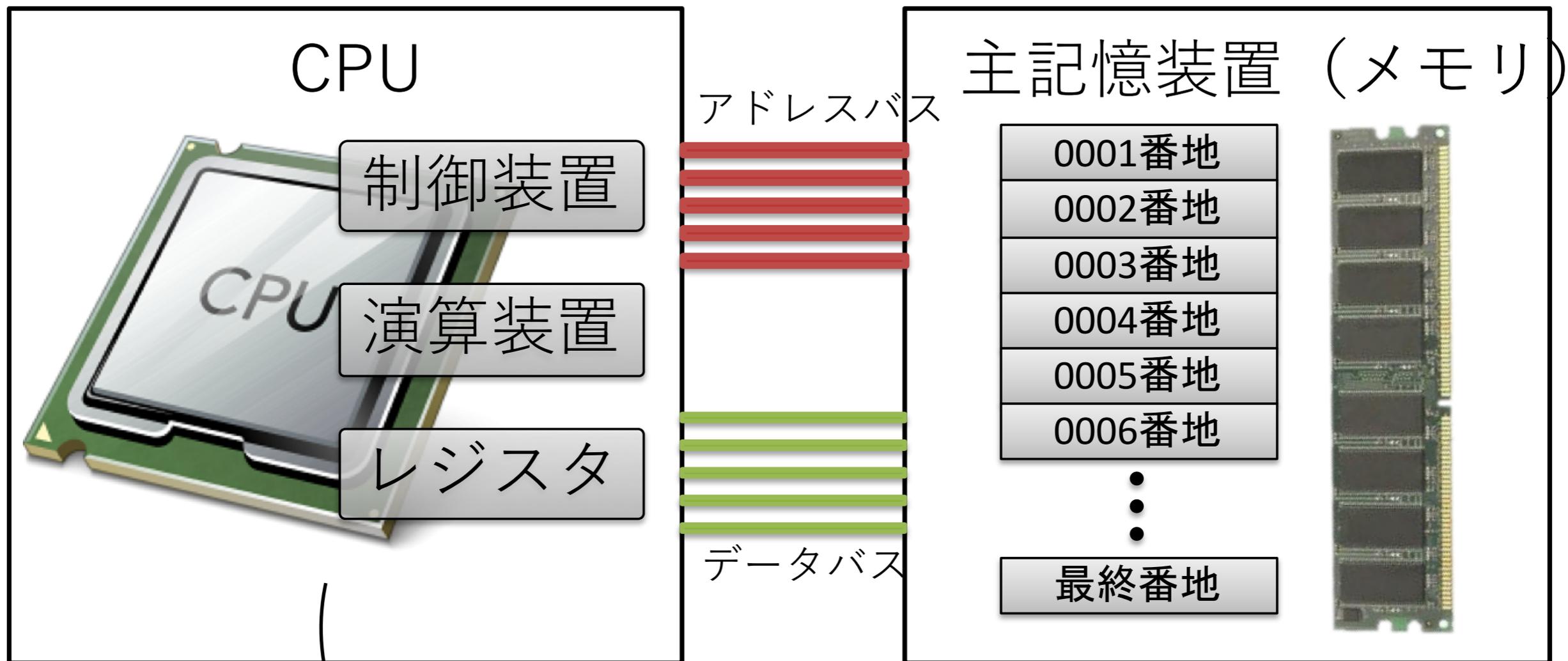
制御装置： メモリから命令を読み込み、演算や入出力をする。

演算装置： データの加算・減算・論理演算を行う。

レジスタ： CPUに内蔵された高速なメモリ。複数あり、それぞれ役割が違う。

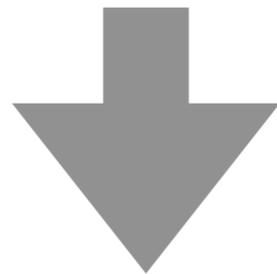
メモリ： データやプログラムが一時記憶される。

番地 (アドレス) で位置を指定する。



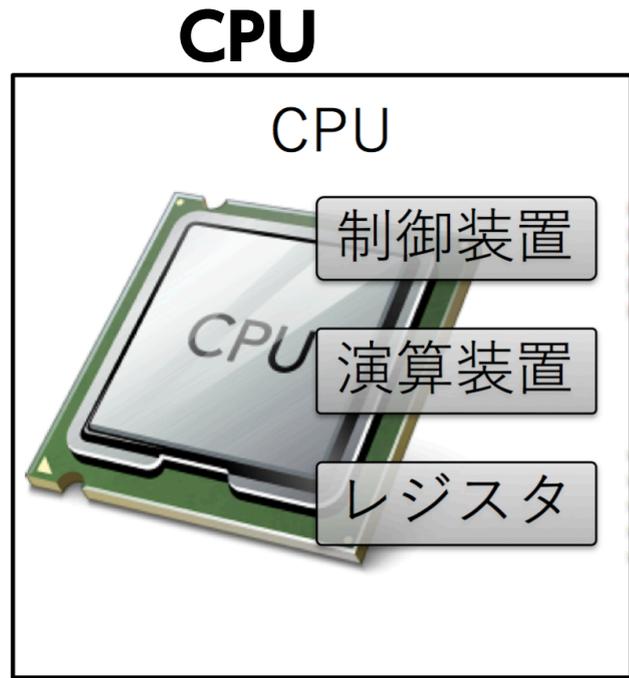
つまりCPUは「0と1の列(デジタル情報)」を読み出して足したり掛けたりしたものをまた別のところへ書き出す、的なことをただひたすらウルトラ超高速でやっているだけ。

- **ゲームや音楽や動画の再生**
- **給与計算や商取引や株取引**
- **原発や航空機や自動車の制御**
- **科学技術計算やAI**
- **メールや通信やインターネット**



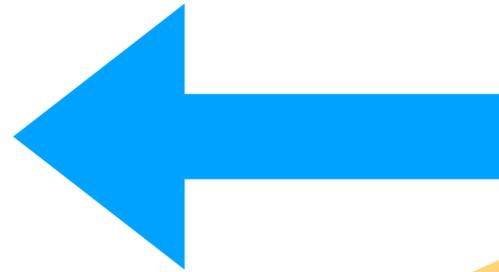
**何をやらせるにも、この「0と1の列の書き換え作業」に
帰着する必要がある！！！！🤪🤪**

ロボが解するコマンド(順番)を
順番に伝える



特定のコマンド(命令)

しか解さない



プログラミング
= プログラムする
プログラムを作る

大問題！ 🙄

これだと人間が長大な0と
1の列を一つのミスもなく
並べなければならない！

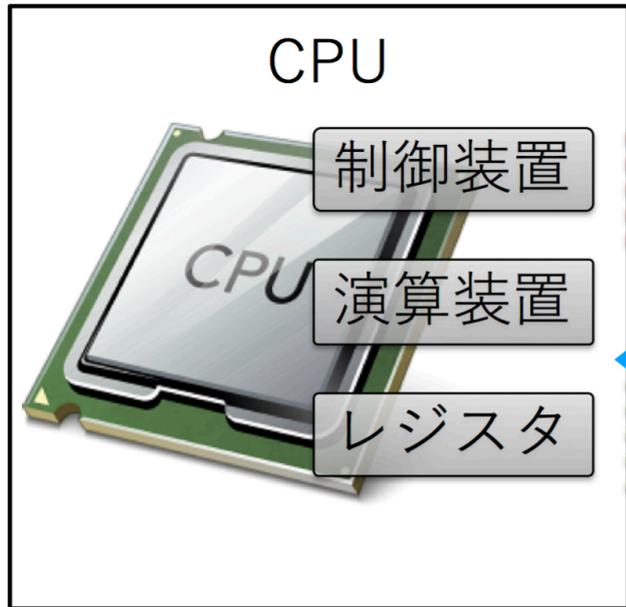


プログラミング言語登場!

CPUが理解できる
コマンド列(機械語)

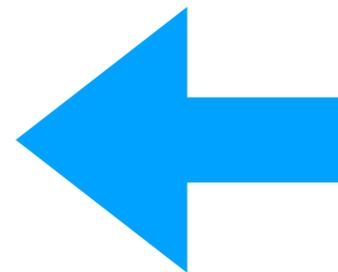
人間が理解できる
レベルでやってほしい
ことを記述
(プログラミング言語)

CPU



特定のコマンド(命令)

しか解さない



翻訳

1. 画面を真っ黒に塗りつぶす
2. 左上から100pixel右、2000pixel下の位置に大きさ50pixelの☆を描画
3. それを黄色で塗りつぶす
4. 同じ作業を右に400pixelずつづらしながら5回繰り返す

言語のルールを決めて
翻訳プログラム
も用意しておく

翻訳のやり方は大きくふた通り

1. コンパイル方式
2. インタプリタ方式

1. コンパイル方式

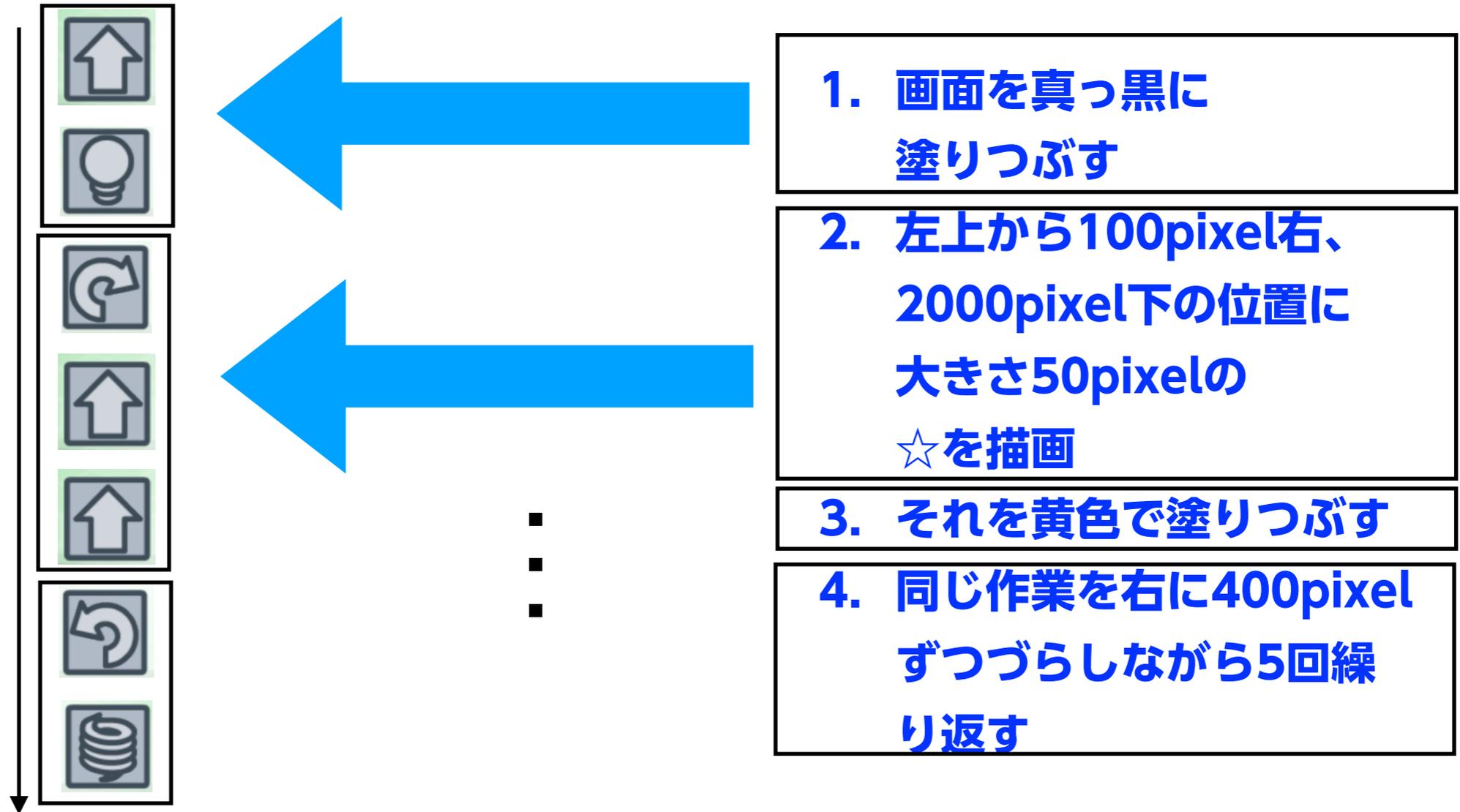


1. 画面を真っ黒に塗りつぶす
2. 左上から100pixel右、2000pixel下の位置に大きさ50pixelの☆を描画
3. それを黄色で塗りつぶす
4. 同じ作業を右に400pixelずつづらしながら5回繰り返す

翻訳業者方式？

- 全体を一気にまとめて訳す
- 全体の文脈がわかるので無駄がない
(質の高い)コマンド列が作れる

2. インタプリタ方式



同時通訳方式？

- 一まとまりずつ訳す
- 全体を作らなくても部分的に実行可能
- ちょっとずつ試しながら作れる

メジャーなプログラミング言語

コンパイラ方式

- C言語
- Java
- C++
- C#
- FORTRAN
- Swift

※インタプリタモードもある

インタプリタ方式

- Python
- PHP
- Ruby
- Perl
- JavaScript
- BASIC

人気ランキングその1 <https://www.tiobe.com/tiobe-index/>

Apr 2018	Apr 2017	Change	Programming Language
1	1		Java
2	2		C
3	3		C++
4	5	↑	Python
5	4	↓	C#
6	7	↑	Visual Basic .NET
7	6	↓	PHP
8	8		JavaScript
9	-	↑↑	SQL
10	11	↑	Ruby

人気ランキングその2

<http://redmonk.com/sogrady/2018/03/07/language-rankings-1-18/>

- 1 JavaScript
- 2 Java
- 3 Python
- 4 PHP
- 5 C#
- 6 C++
- 7 CSS
- 8 Ruby
- 9 C
- 10 Swift
- 10 Objective-C

プログラミング言語

Java ... 最も使われている言語のひとつ。業務システムで使われることが多い。AndroidアプリはJavaで作る。ただ本当の初心者には少々学習障壁があるかも。

参考) 瀧川は情報理工学コース 3年生のJava演習を設計・担当

情報理工学実験I: オブジェクト指向プログラミング

https://art.ist.hokudai.ac.jp/~takigawa/csit_java/

C# ... 言語としてはJavaに似てる。Javaより後発なので描きやすい面もある(Javaも言語仕様をupdateし続けて流けど)。Windows向けのゲームやVR開発に良いかも。

C/C++ ... ハードを明確に意識する必要があるがその分非常に実行速度の速いソフトウェアが作れる。レスポンス速度が問われるゲームや組み込み機器の開発などでは必須。

JavaScript/PHP ... Webサービスやりたいなら必須。それ以外の用途では汎用性がいまいちなので今回はパス。なお、JavaとJavaScriptは全く別の言語でほぼ無関係。

Ruby ... Pythonと立ち位置は似ているがやはり最近ではWebサービス向け。JavaScript/PHPより汎用的。

Swift ... iPhoneアプリ開発向け。新しい言語(2014年)。

FORTRAN ... 最も古い言語の一つ。科学技術計算向け。

Python

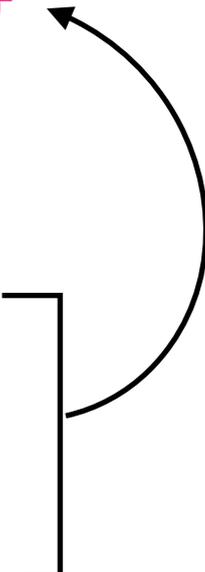
- **非常に読みやすく書きやすい言語で最初に学ぶ言語としてもとても良い。実際海外の大学ではほぼPythonになりつつある。Webに情報や教材が多く自学しやすい。**
- **インタプリタ方式でちょっとずつ実行もできるし、AnacondaやJupyterなどの環境も手軽に利用可能**
- **使い捨てのちょい作業から数百万行あるような大規模システムまで非常に幅広く使われており実用的な言語**
- **本演習で挙げた「集計から人工知能まで」を考えると現在データ解析や人工知能実装で最も使われている。**

大雑把なビジョン

- この演習でやりたいことは大きく2つ
 1. 「プログラミング」とは何かを理解しそれをどうやって学んでいけば良いかの指針を与える
 2. 「Python」というプログラミング言語の基礎と実習を実際にやってみる

ちらっと説明

この2つは演習で必要なだけ



1. Anacondaの最新版をPCにインストール
2. USBメモリを「exFAT」形式でクイックフォーマット
3. PCのフォルダごとUSBにコピー
4. Anaconda Navigator経由でJupyter Notebookを起動

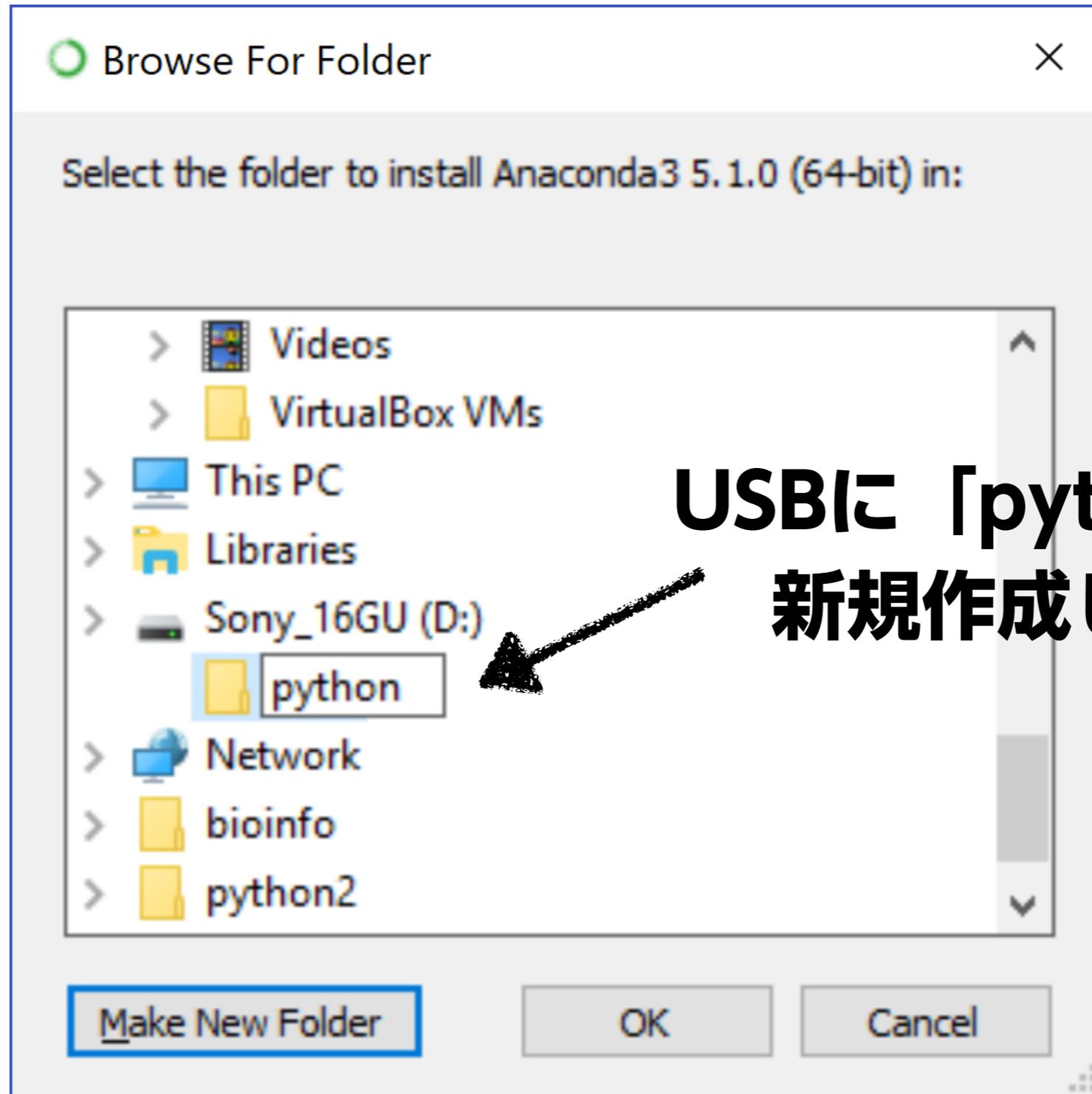
3種類出てきますが説明は追い追い

1. Python (パイソン)
2. Anaconda (アナコンダ)
3. Jupyter (ジュピター)

USB (ユーエスビー)

exFAT (イーエックスファット)

インストール先を自分のPCかUSBに



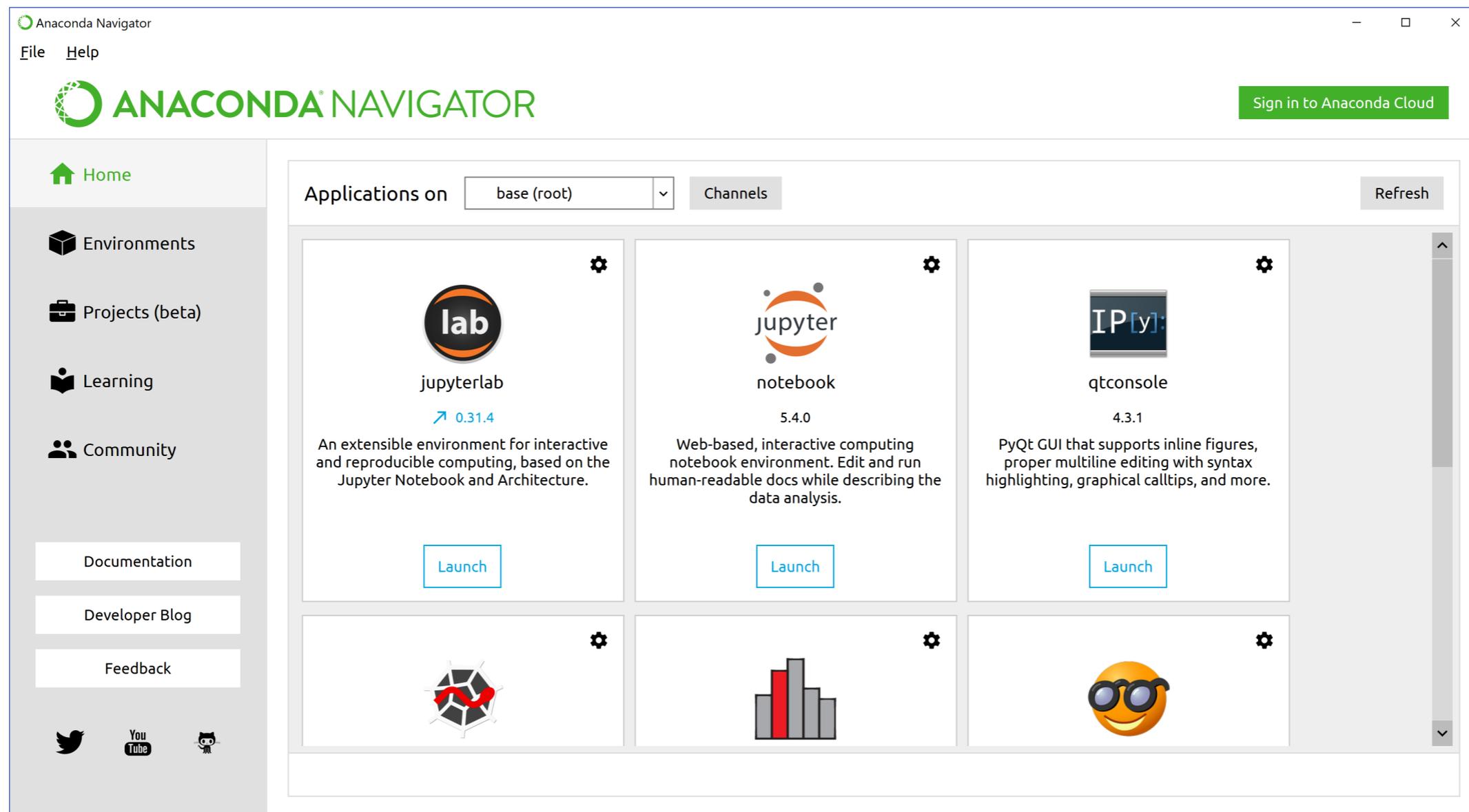
USBに「python」を
新規作成し選択

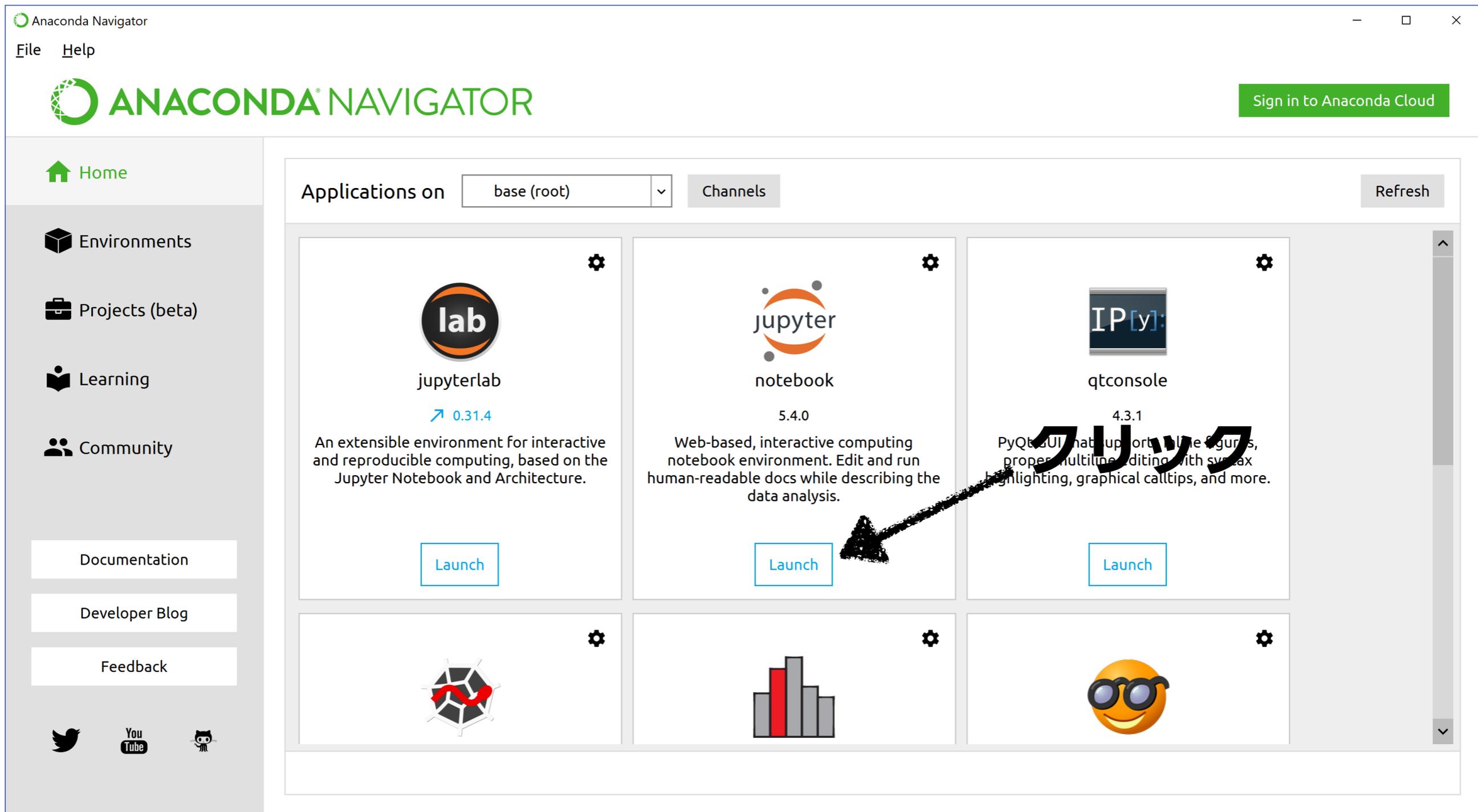
セットアップしたPython環境を起動してみる

インストールしたフォルダの中の

「**Scripts**」の中にある

anaconda-navigatorをダブルクリック





注：なお時期versionではJupyterLabに移行なのでそっちでもいいかも



Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↕ ↻

<input type="checkbox"/> 0	▼	📁 / Desktop / mypython	Name ↓	Last Modified
		📁 ..		数秒前
The notebook list is empty.				

localhost:8888/tree/Desktop/mypython

jupyter Logout

Files Running Clusters

Select items to perform actions on them. Upload New ↕ ↻

	Name ↓	Last Modified
<input type="checkbox"/> 0	/ Desktop / mypython	
<input type="checkbox"/>	..	数秒前

The notebook list is empty.

見たいフォルダを指定できる

新しいフォルダを作成したりもできる

Select items to perform actions on them.

0 / Desktop / mypython

..

The notebook list is empty.

Upload New

- Notebook:
 - Python 3
- Other:
 - Text File
 - Folder
 - Terminal

ここにPythonコードを打ち込む
「Run」をクリックするか、ShiftとEnterの同時押しで実行

「1 + 2」 「print("Hello, World!")」 「import this」
などを入力・実行してためしてみよう

作業：授業のホームページから「[myfirstcode.ipynb](#)」をダウンロードして開いて実行してみよう。

<http://art.ist.hokudai.ac.jp/~takigawa/prog/>

Pythonが起動しているフォルダにコピーし、
クリックして開く！

 jupyter

Logout

Files

Running

Clusters

Nbextensions

Select items to perform actions on them.

Upload

New ▾



<input type="checkbox"/> 0		Name ↓	Last Modified
<input type="checkbox"/>	files		7 days ago
<input type="checkbox"/>	python		7 days ago
<input type="checkbox"/>	 myfirstcode.ipynb		Running seconds ago

**今日は「Hour of Code」のデモ版というのを
やってみましょう！**

<http://lightbot.com/hoc.html>

1. 基本

2. プロシージャ

3. ループ

**このゲームがコンピュータ
の仕組みやプログラミング
とどう関係するのかは
次週以降解説！**

**さらに自習したい人は下記の好きなものを
(登録しなくても試せるが結果を残したかったら登録)**

1. Hour of Code

<https://hourofcode.com/jp/learn>

2. Code Monkey

<https://codemonkey.jp>

3. Scratch

<https://scratch.mit.edu>

4. Code Combat

<https://codecombat.com>

Scratch <https://scratch.mit.edu>



作る

見る

ヒント

Scratchについて

検索

Scratchに参加しよう

サインイン

物語やゲーム、アニメーションを作って
世界中の人と共有しよう
クリック



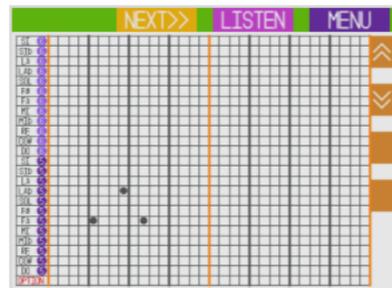
30,824,336 プロジェクトが共有されているクリエイティブ・ラーニング・コミュニティ

[Scratchについて](#) | [教育関係者の方へ](#) | [保護者の方へ](#)

注目のプロジェクト



Fire pen project (with mrgreengoody)



SONO KREO ver 1.3. mochimochiking



Colour Palette Gener CaramelCloud



Challenger Qwerty



Flashlight | Complete catak



Scratch <https://scratch.mit.edu>

The image shows the Scratch web interface in Japanese. At the top, the Scratch logo and the URL <https://scratch.mit.edu> are displayed. The main workspace is titled "Untitled" and shows the Scratch cat sprite on a stage. The right sidebar contains a "Scratchをはじめよう" (Getting Started) panel with a "始める" (Start) button and a "作成手順一覧" (List of creation steps) button. The bottom left shows the "Sprites" panel with "Sprite1" selected. The right sidebar also features a "Scratchをはじめよう" (Getting Started) panel with a "始める" (Start) button and a "作成手順一覧" (List of creation steps) button. The right sidebar also features a "Scratchをはじめよう" (Getting Started) panel with a "始める" (Start) button and a "作成手順一覧" (List of creation steps) button.

今日のお題：プログラミング事始

- 前回の復習とふりかえり
- コンピュータの仕組みとプログラム、社会における役割
- プログラミング言語とは？
- Pythonとは？
- Lightbotをやってみる / Scratchをやってみる
- (時間があれば) セットアップしたPython環境を試す
- ミニレポートタイム(15分)

注意

- 教室が変わるので注意してください。

4/19 (木) ・ 4/26 (木) ・ 情報教育館 B (情報教育館 2階)

5/10 (木) ~ 8/2 (木) ・ E209教室

- 8G以上の容量の「USBメモリ」を持参してください。