

Python Beginners Okinawa 22

初心者向け勉強会

じこしょーかい



- ・ 名前：大城 慶知(よしとも)
- ・ 琉球大学 情報工学科 3年次
Hugkun 学生CTO
- ・ 経歴
小中 -> 球陽高校 -> 琉球大学 学生(15年目)
- ・ 好きなもの：
コーヒー(最近好きになった)
ボードゲーム

↑ Twitterアイコン(変えたい)



@bouch_666



/yoshitomo.oshiro

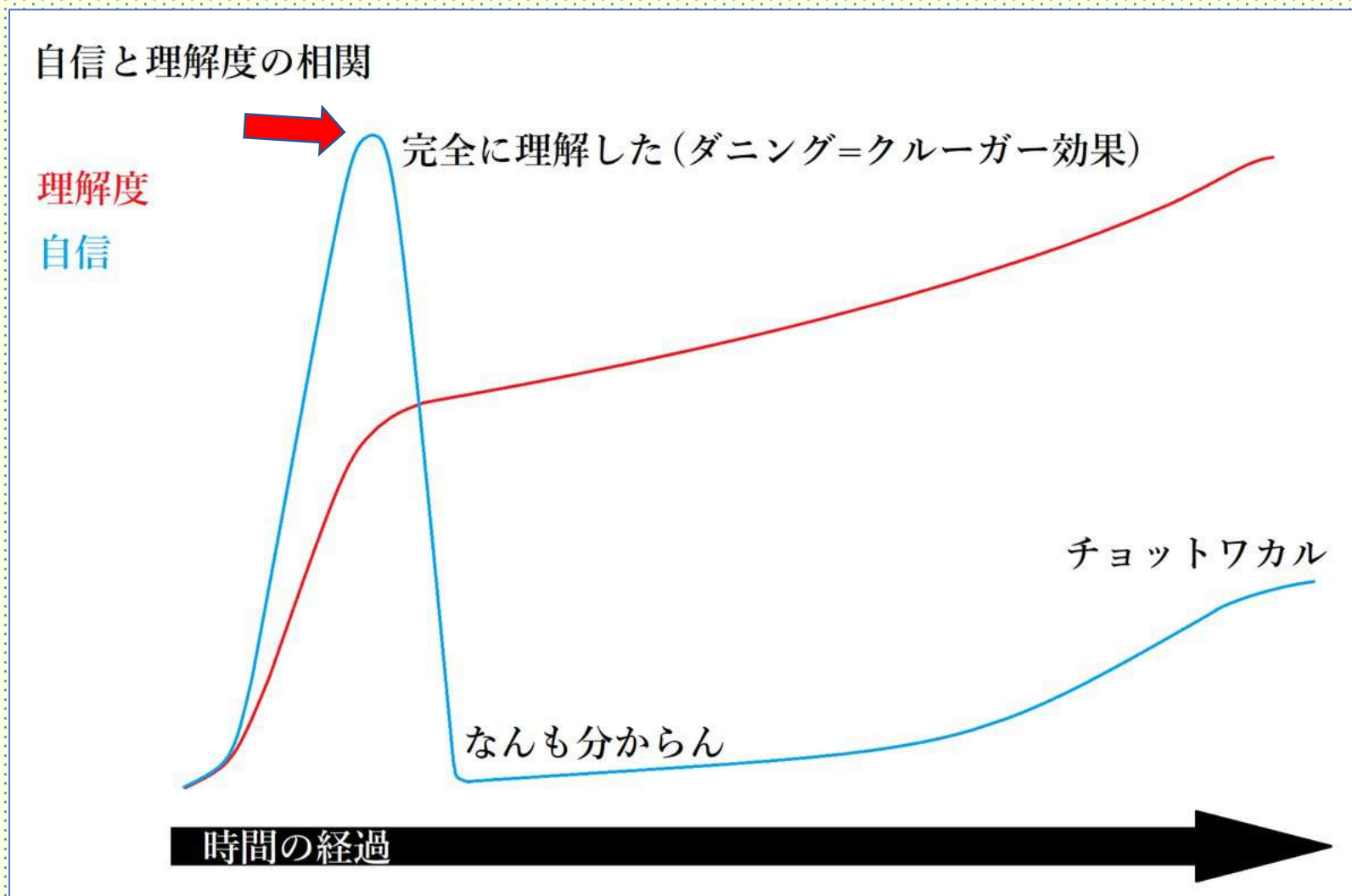
目次

1. 環境構築 とプログラミング周辺話 ・ ・ ・ 15分
2. 型, 構文, 関数 を使ってみるー ・ ・ ・ 30分
3. 機械学習 MNIST ハンズオン ・ ・ ・ 45分
4. クラス、コンストラクタを用いた設計 ・ ・ ・ 45分

注意点

- ※ 初心者向けですが、
わかりにくい箇所があるかもしれません。
- ※ 結構ツメツメです
- ※ この勉強会は、完全に理解するまでが目標です

完全に理解したとは



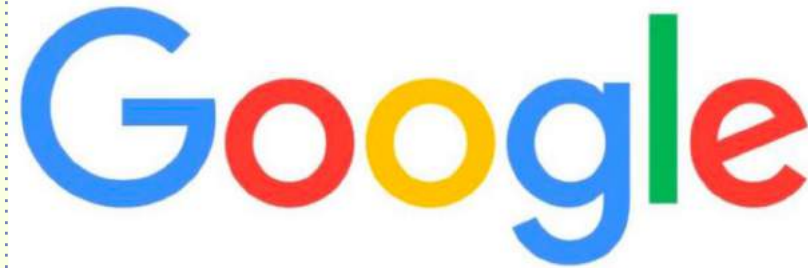
ではでは、はじめていきます

Step.1 環境構築 と プログラミング周辺話

プログラミングをする上で

◀ 5 ▶

困ったらググること



先生に聞こう

プログラミングをする上で

< 5 >

忘れるので継続すること

だいたい同じ、体力をシウモウする
消耗！！

例



パソコンばかりやっていると
漢字がかけなくなる

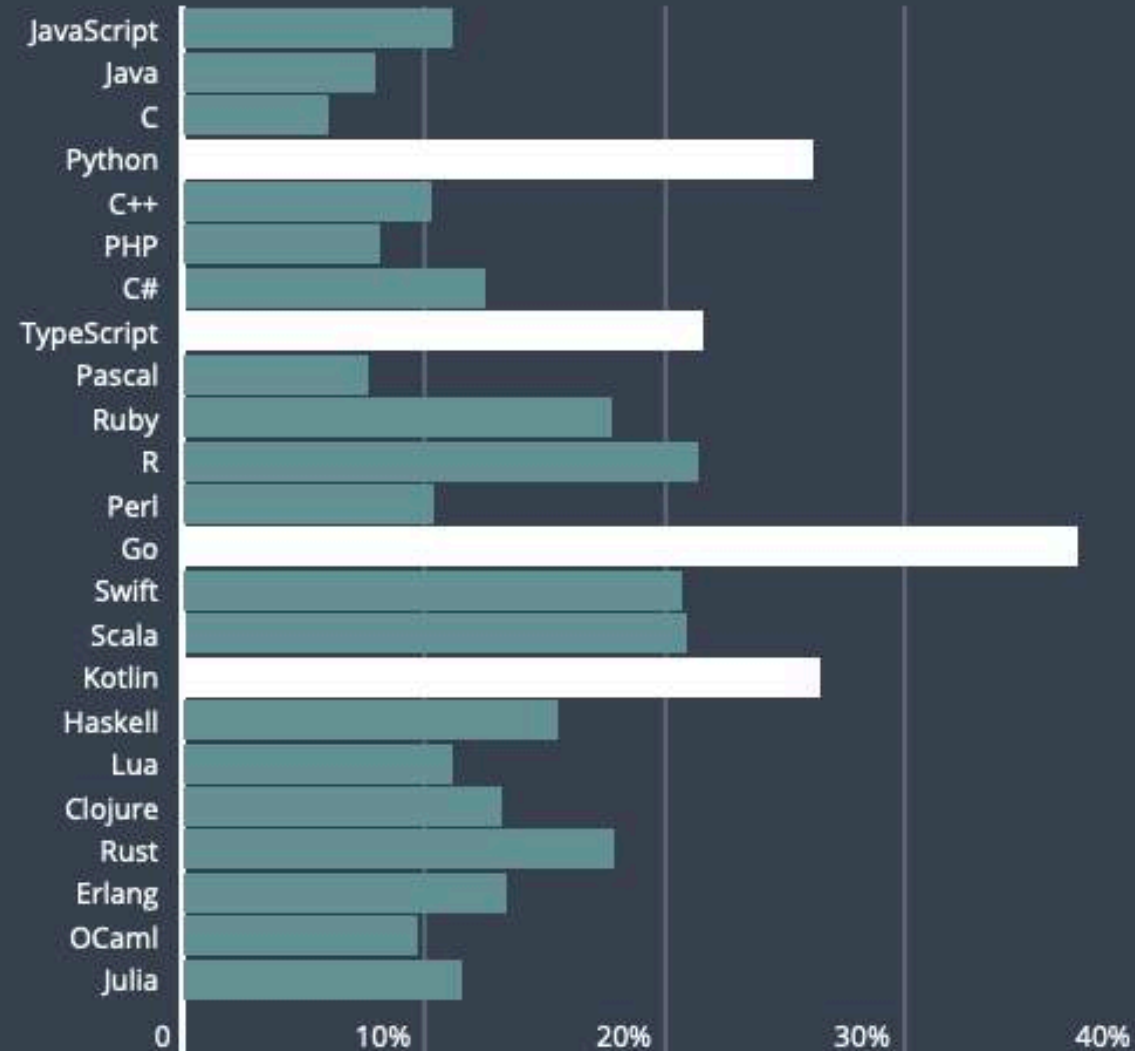
Python とは

- 言語

求人検索エンジン「スタンバイ」プログラミング言語別年収ランキング2018
(提示年収の中央値ベスト10)

順位	言語	年収中央値 (万円)	最大提示年収 (万円)	求人数 (件)
1	Go	600	1,600	2,202
2	Scala	600	1,300	1,489
3	Python	575.1	1,499	9,344
4	Kotlin	575	1,200	961
5	TypeScript	575	1,200	667
6	R	574.8	1,000	220
7	Ruby	550	1,200	11,676
8	Swift	550	1,200	3,353
9	Perl	525	1,200	4,509
10	C	525	1,000	9,347

Languages developers want to learn in 2019



出典： HackerRank 7万人の開発者 アンケート

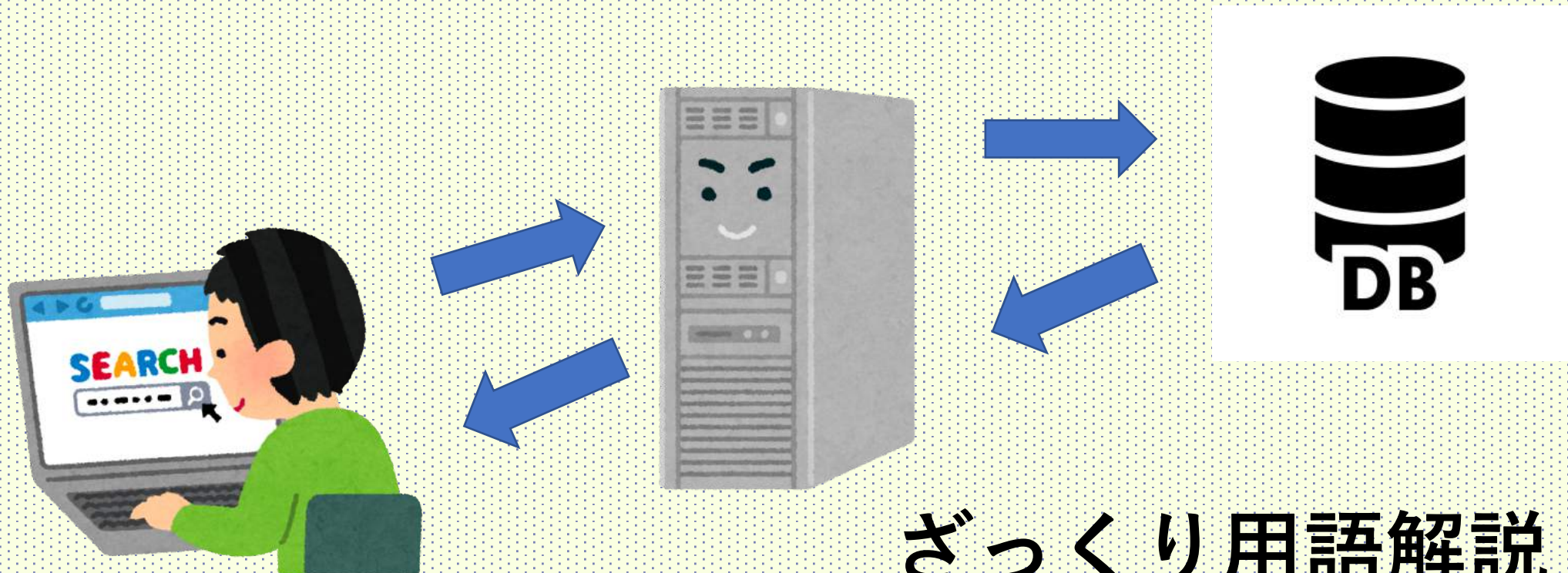
Python とは

- ・ 言語としては低速 高級言語 (遅くて 理解しやすい)
- ・ 機械学習(俗にいうAI), データ分析,
Webアプリケーション(バックエンド多め)
- ・ 非標準のフレームワーク を利用するのが多い
 - ・ 動作を早く
 - ・ 楽
 - ・ 機能を知っている必要がある(知らなければググる)

Webアプリケーション

<https://tech-camp.in/note/technology/45702/>

上記のリンクのようなWebページだとしたら、
大雑把に言うと下のような図



ざっくり用語解説

フレームワーク

またの名をライブラリ, モジュール 小屋を作りたい

イメージ：他の人が作った道具



ハンマー



釘



板

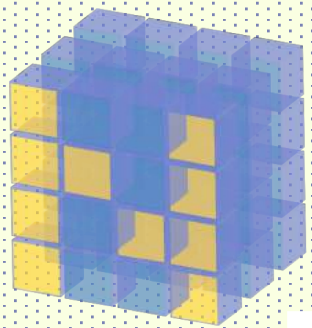
ざっくり用語解説

フレームワーク

< 5 >

またの名をライブラリ

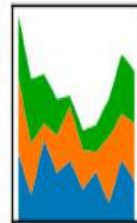
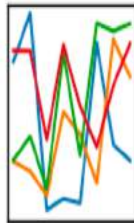
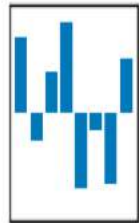
Webアプリを作ろう



NumPy

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



計算



Flask

web development,
one drop at a time

Json
datetime

必要な型

基盤

ざっくり用語解説

フレームワーク

またの名をライブラリ

- ・ 特徴とか知ってるのが強み
- ・ 道具をうまく使えると早く開発が進む
- ・ 道具を作る側もいる

ざっくり用語解説

環境構築とは

環境とは、物置のようなもの。フレームワークを入れる



Anaconda3



pyenv venv

環境構築とは

環境とは、物置のようなもの。

Anaconda3

- ・ 最初から色々入っている



- ・ フレームワークをインストールするときに関連するフレームワークをもインストールする

- ・ Windows 側のGUI でも動く

pyenv venv

- ・ シンプル

- ・ 必要なフレームワークは自分で入れる

- ・ CUIでの動作



環境構築とは

環境とは、物置のようなもの。

Anaconda3

- ・ 最初から色々入っている



- ・ フレームワークをインストールするときに関連するフレームワークをもインストールする

- ・ Windows 側のGUI でも動く

pyenv venv

- ・ シンプル

- ・ 必要なフレームワークは自分で入れる

- ・ CUIでの動作

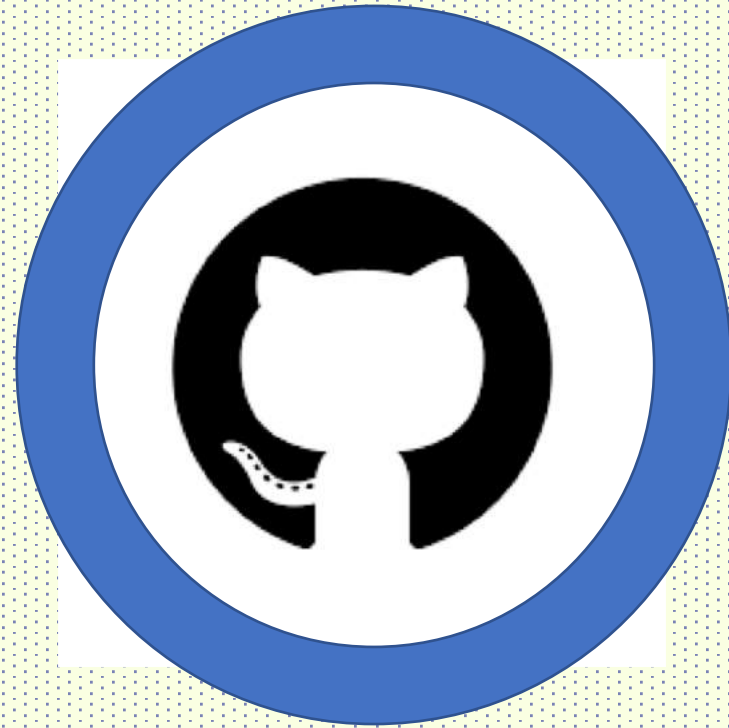
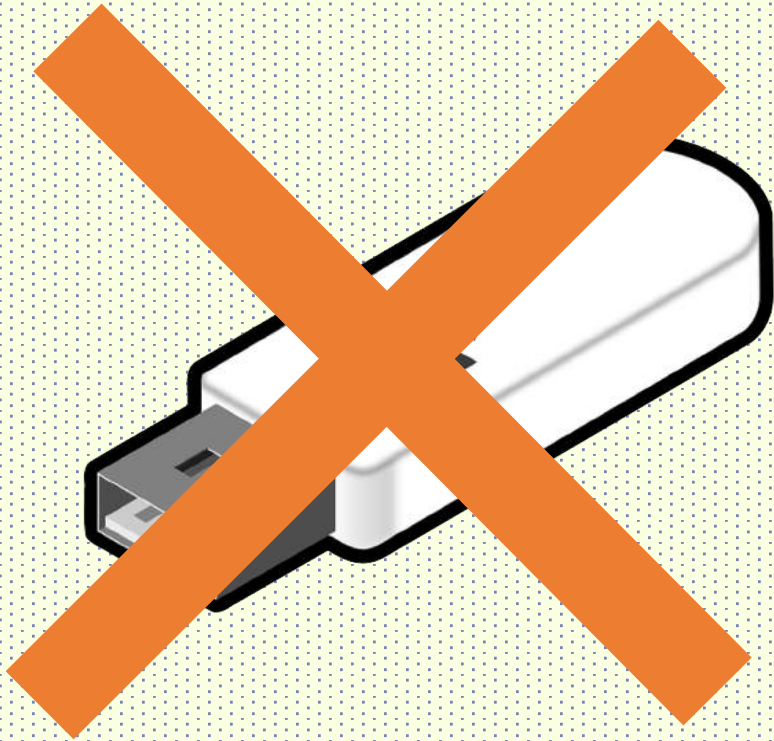


周辺知識

- ・ VCS (バージョン管理システム)・・・最悪ここまで
 - ・ Github
- ・ IDE(開発統合環境)
- ・ 分野別おすすめ本・フレームワーク
 - ・ データ分析
 - ・ 機械学習
 - ・ Webアプリケーション

Github

USBでソースコード管理するのはモダンではない。



Web に保存しよう

ってだけ。

メリット

- ・複数人でソースコードをいじれる
- ・活動のログを取れる
- ・公開しているフレームワークはこっちで見れる

デメリット

- ・ネットが無いとアクセスできない

Github

というわけで、
今回の資料はこちらで取れます。

https://github.com/bouch20/PyBegi_Okinawa22

IDE

簡単にいうと、万能エディター

- ・ 代表格 PyCharm (有能)
- ・ Atom (カスタマイズ必要)
- ・ Spyder (わかりません)



PyCharm

- ・ 環境・フレームワーク管理
- ・ VCS 管理
- ・ ディレクトリ管理
- ・ デプロイ
- ・ Docker起動・実行

PyCharmは学生は申請すると有料版を無料で使える

フレームワーク(標準外)

- データ分析
 - NumPy (高速演算ツール)
 - Pandas (データ処理)
 - Scipy (演算系)
 - mysql-connector (DBの知識も必要)
 - Matplotlib (可視化)
 - Jupyter Notebook (プラットフォーム)

フレームワーク(標準外)

- 機械学習
 - NumPy (高速演算ツール)
 - Pandas (データ処理)
 - Scipy (演算系)
 - mysql-connector (DBの知識も必要)
 - Matplotlib (可視化)
 - +
 - Scikit-learn PyTorch, Tensorflow, Chainer...etc
(機械学習ライブラリ)

フレームワーク(標準外)

- Web アプリケーション
 - NumPy (高速演算ツール)
 - Pandas (データ処理)
 - Flask, Django, Tornado, aiohttp(Web フレームワーク)
 - mysql-connector (DBの知識も必要)

他には機能によって追加する