

APIを使ってみよう

ゴール：

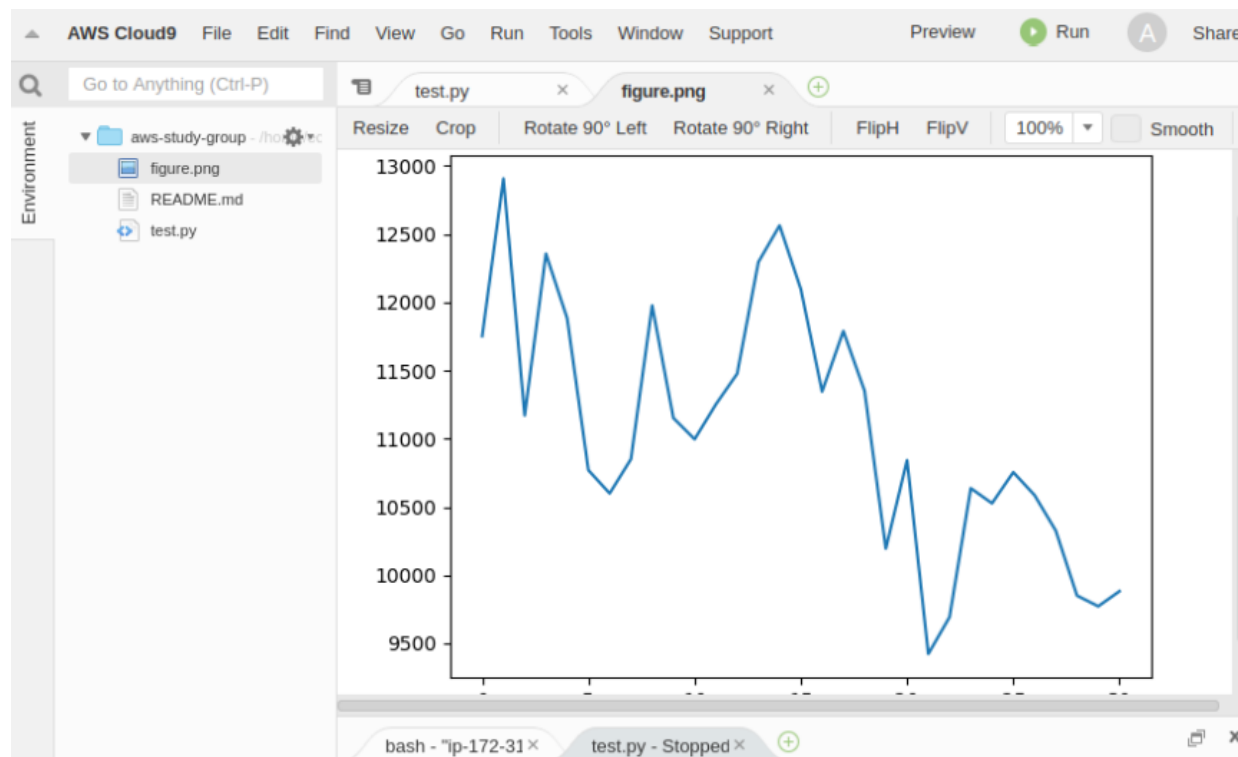
- ・ APIを使うことができるようになる

内容：

- ・ Cloud9 を使ってAWS上に開発環境を構築する(20分)
- ・ Pythonとは(5分)
- ・ Pythonの基本(35分)
- ・ APIとは(5分)
- ・ APIを使ってみる(25分)

今日のゴール

- Pythonを使ってBitcoinの価格を取得する
- 取得した結果から過去1か月間の値動きをグラフにしてみる

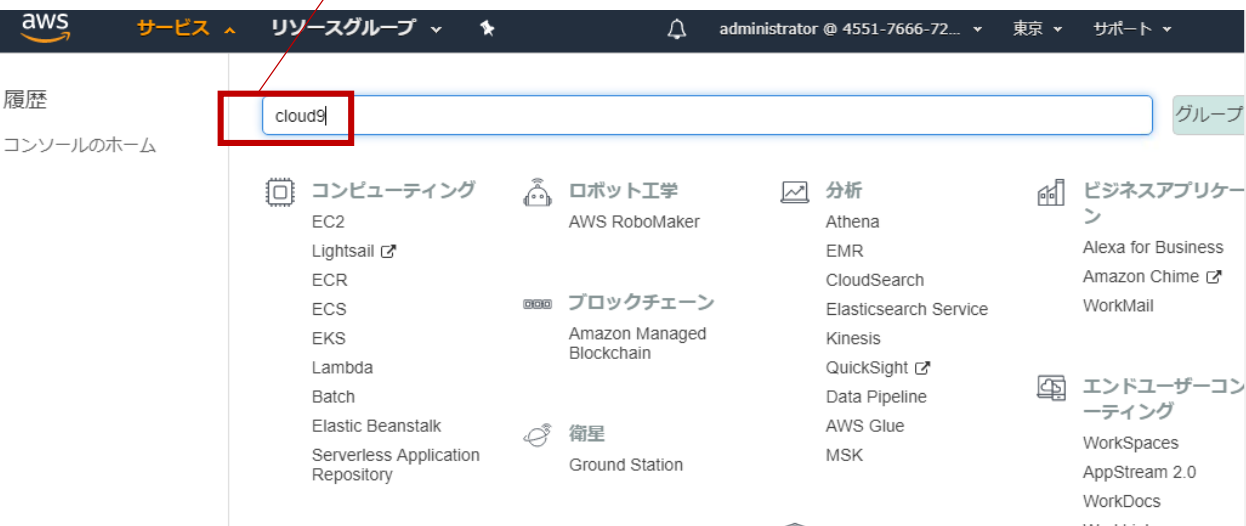


クラウドの主要なサービス

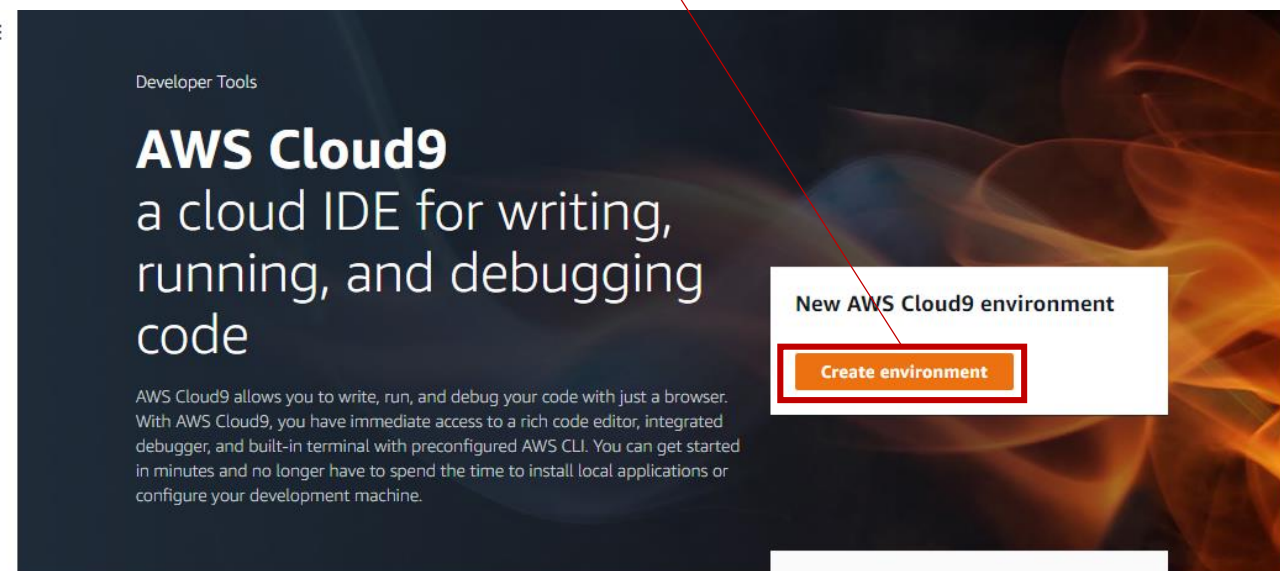


Cloud9 を使ってAWS上に開発環境を構築する

サービスよりCloud9と入力



Create Environmentをクリック



特に設定は変えない

AWS Cloud9 > Environments > Create environment

Step 1
Name environment

Step 2
Configure settings

Step 3
Review

Name environment

Environment name and description

Name
The name needs to be unique per user. You can update it at any time in your environment settings.

test-cloud9

Limit: 60 characters

Description - Optional
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.

test-cloud9

Limit: 200 characters

Cancel **Next step**

Environment type [Info](#)
Choose between creating a new EC2 instance for your new environment or connecting directly to your server over SSH.

☒ **Create a new instance for environment (EC2)**
Launch a new instance in this region to run your new environment.

☐ **Connect and run in remote server (SSH)**
Display instructions to connect remotely over SSH and run your new environment.

Instance type

☒ **t2.micro (1 GiB RAM + 1 vCPU)**
Free-tier eligible. Ideal for educational users and exploration.

☐ **t2.small (2 GiB RAM + 1 vCPU)**
Recommended for small-sized web projects.

☐ **m4.large (8 GiB RAM + 2 vCPU)**
Recommended for production and general-purpose development.

☐ **Other instance type**
Select an instance type.

t2.nano

Platform

☒ **Amazon Linux**

☐ **Ubuntu Server 18.04 LTS**

Cost-saving setting
Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default)

IAM role
AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

► **Network settings (advanced)**

Cancel Previous step **Next step**

Create Environmentをクリック

Name
test-cloud9

Description
test-cloud9

Environment type
EC2

Instance type
t2.micro

Subnet

Platform
Amazon Linux

Cost-saving settings
After 30 minutes (default)

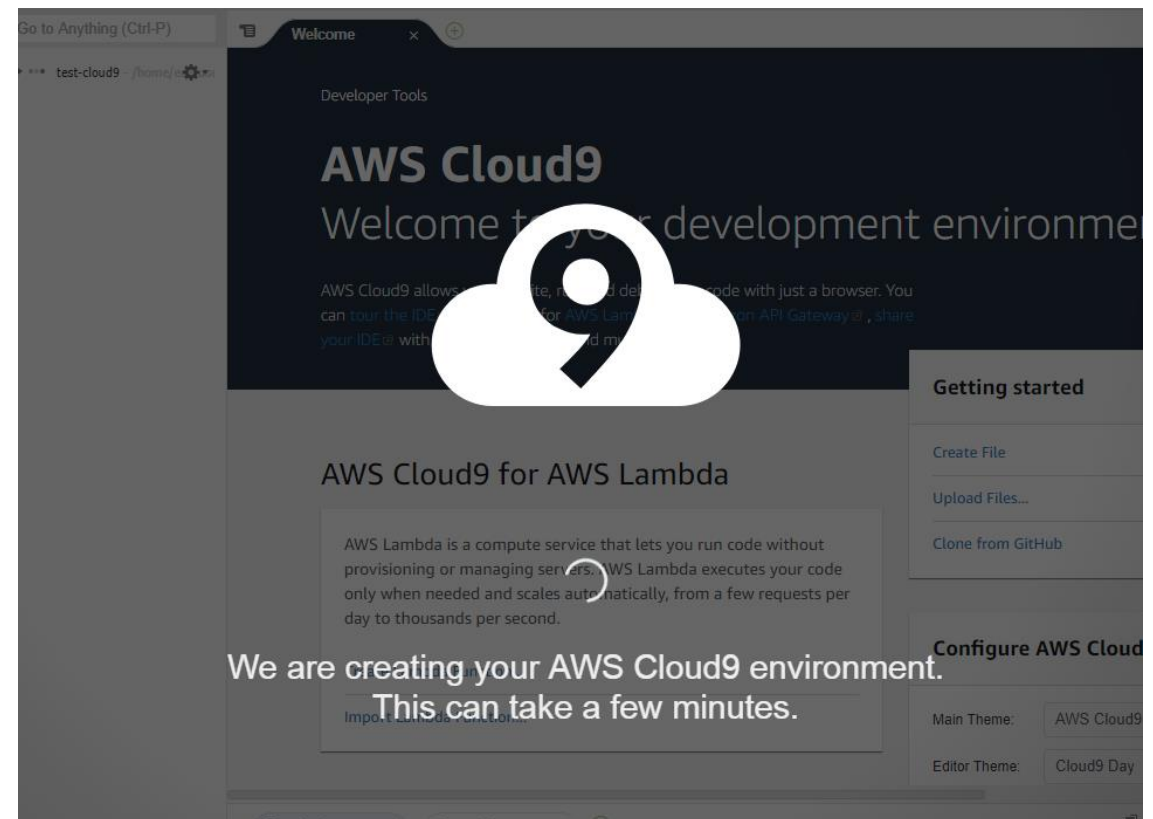
IAM role
AWSServiceRoleForAWSCloud9 (generated)

We recommend the following best practices for using your AWS Cloud9 environment

- Use **source control and backup** your environment frequently. AWS Cloud9 does not perform automatic backups.
- Perform regular **updates of software** on your environment. AWS Cloud9 does not perform automatic updates on your behalf.
- **Turn on AWS CloudTrail in your AWS account** to track activity in your environment. [Learn more](#)
- Only share your environment with **trusted users**. Sharing your environment may put your AWS access credentials at risk. [Learn more](#)

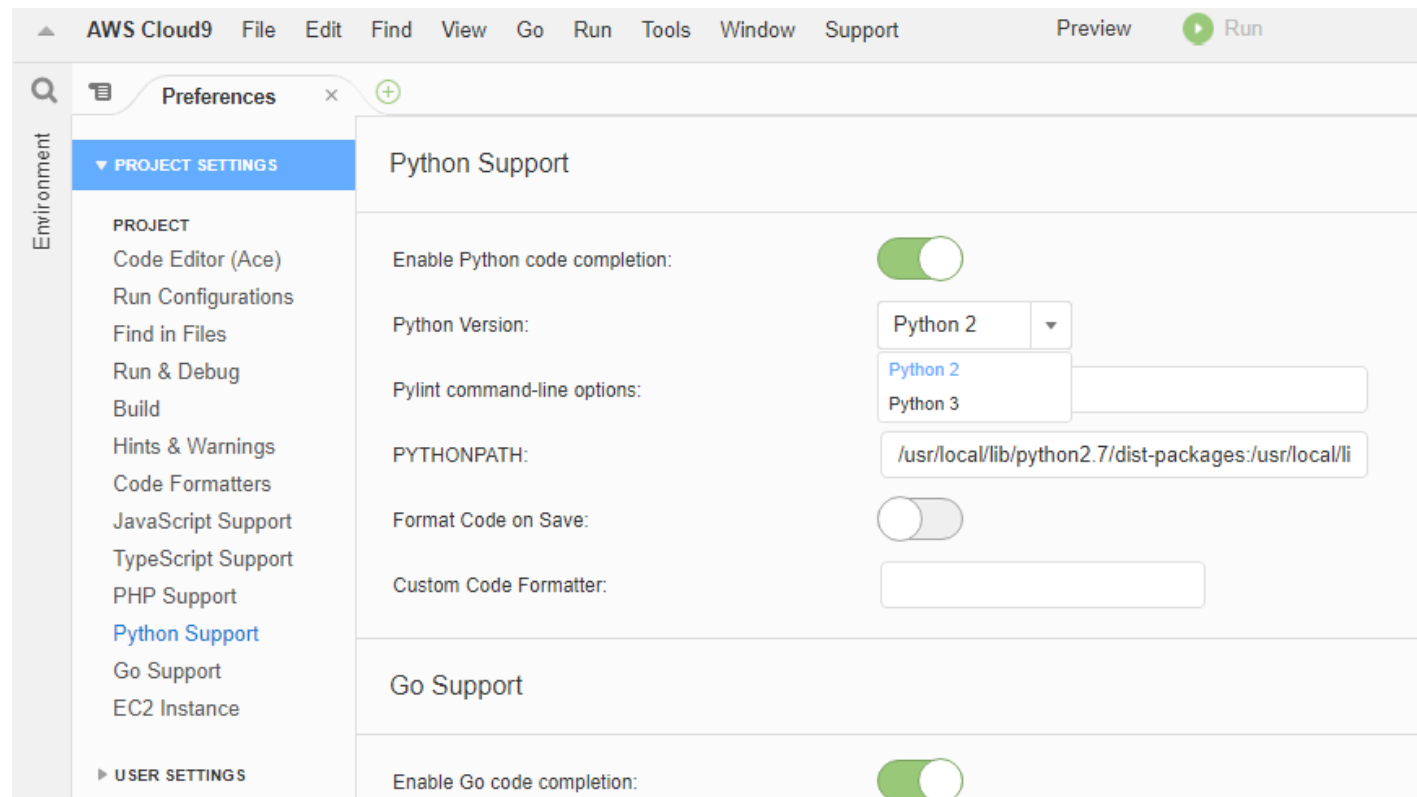
Cancel Previous step **Create environment**

開発環境の作成が始まる



Pythonの環境設定

[AWS Cloud9] – [Preferences] – [Python Support]
の順に選択して、「Python Version」を**Python3**に
変更する



Pythonの環境設定

Cloud9下の「bash～」と表示されている部分に以下のコマンドを順番に入力してみよう

```
python --version  
python3 --version  
pip --version
```



The screenshot shows a terminal window titled "Getting started" with a tab labeled "bash - 'ip-172-31' x Immediate x (+)". The terminal output is as follows:


```
administrator:~/environment $ python --version  
Python 2.7.16  
administrator:~/environment $ python3 --version  
Python 3.6.8  
administrator:~/environment $ pip --version  
pip 9.0.3 from /usr/lib/python2.7/dist-packages (python 2.7)  
administrator:~/environment $
```


Pythonの環境設定

Cloud9下の「bash～」と表示されている部分に以下のコマンドを順番に入力してみよう

```
vi ~/.bashrc
```

```
# ファイル編集後  
source ~/.bashrc
```



```
vim - "ip-172-31-... Immediate x (+)  
# .bashrc  
  
export PATH=$PATH:$HOME/.local/bin:$HOME/bin  
  
# load nvm  
export NVM_DIR="$HOME/.nvm"  
[ "$BASH_VERSION" ] && npm() {  
  # hack: avoid slow npm sanity check in nvm  
  if [ "$*" == "config get prefix" ]; then which node | sed "s/bin\/node\/";  
  else $(which npm) "$@"; fi  
}  
# [ -s "$NVM_DIR/nvm.sh" ] && . "$NVM_DIR/nvm.sh" # This loads nvm  
rvm_silence_path_mismatch_check_flag=1 # prevent rvm complaints that nvm is first in PATH  
unset npm # end hack  
  
# User specific aliases and functions  
alias python=python27  
  
# modifications needed only in interactive mode  
if [ "$PS1" != "" ]; then  
  # Set default editor for git  
  git config --global core.editor /usr/bin/nano
```

python27→python36に変更

Pythonの環境設定

以下のコマンドを入力

```
sudo update-alternatives --config python
```

```
administrator:~/environment $  
administrator:~/environment $ ls  
README.md  
administrator:~/environment $ sudo update-alternatives --config python  
  
There are 2 programs which provide 'python'.  
  
   Selection    Command  
-----  
*+ 1            /usr/bin/python2.7  
   2            /usr/bin/python3.6  
  
Enter to keep the current selection[+], or type selection number: 2
```

Pythonの環境設定

以下のコマンドを入力。

最後に最初と同じコマンドを入力してpythonのバージョンを確認

```
sudo pip install --upgrade pip  
hash -r  
pip install requests --user  
  
python --version  
pip --version
```

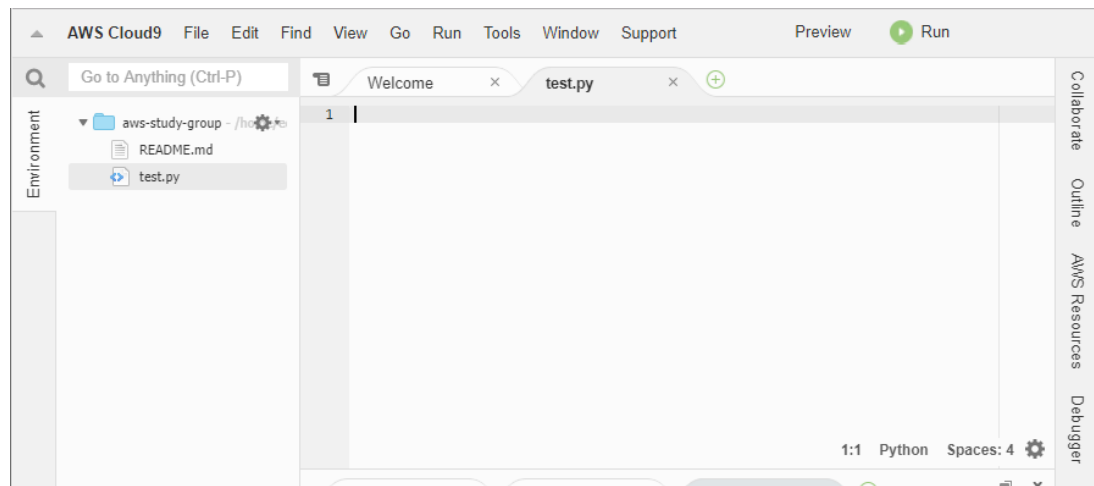
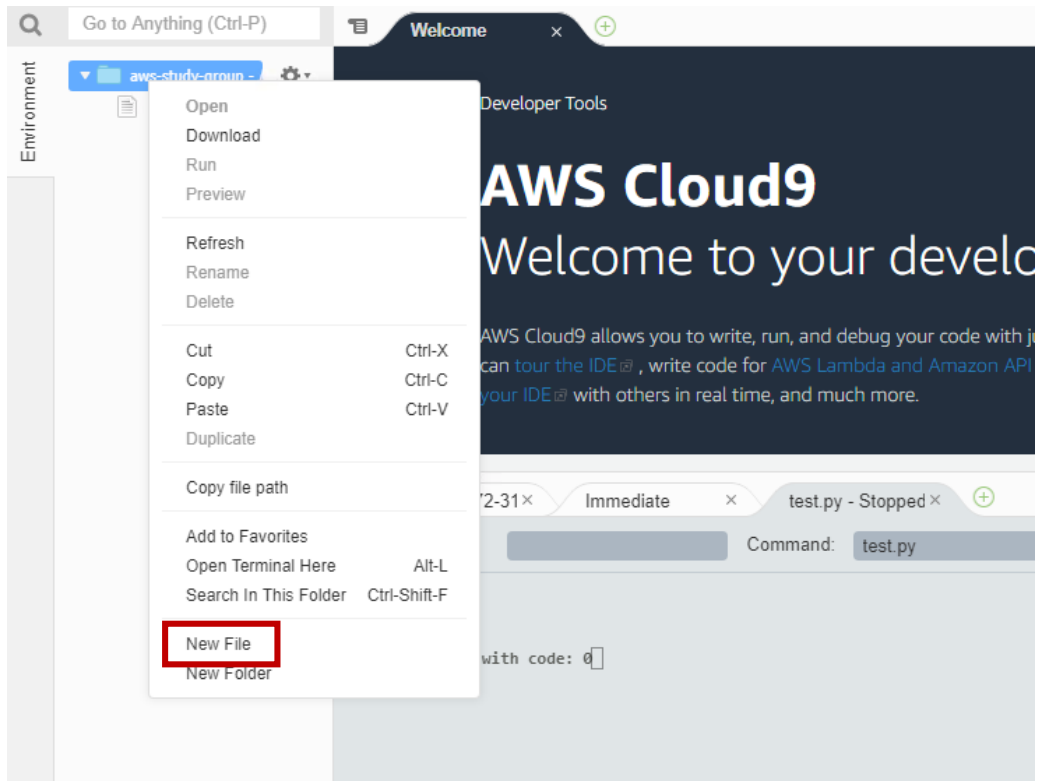
```
wrapc 1.11.2  
administrator:~/environment $ python --version  
Python 3.6.8  
administrator:~/environment $ pip --version  
pip 19.2.1 from /usr/local/lib/python3.6/site-packages/pip (python 3.6)  
administrator:~/environment $ █
```

早速使ってみよう

1. 右クリック→「New File」を選択してファイルを作成

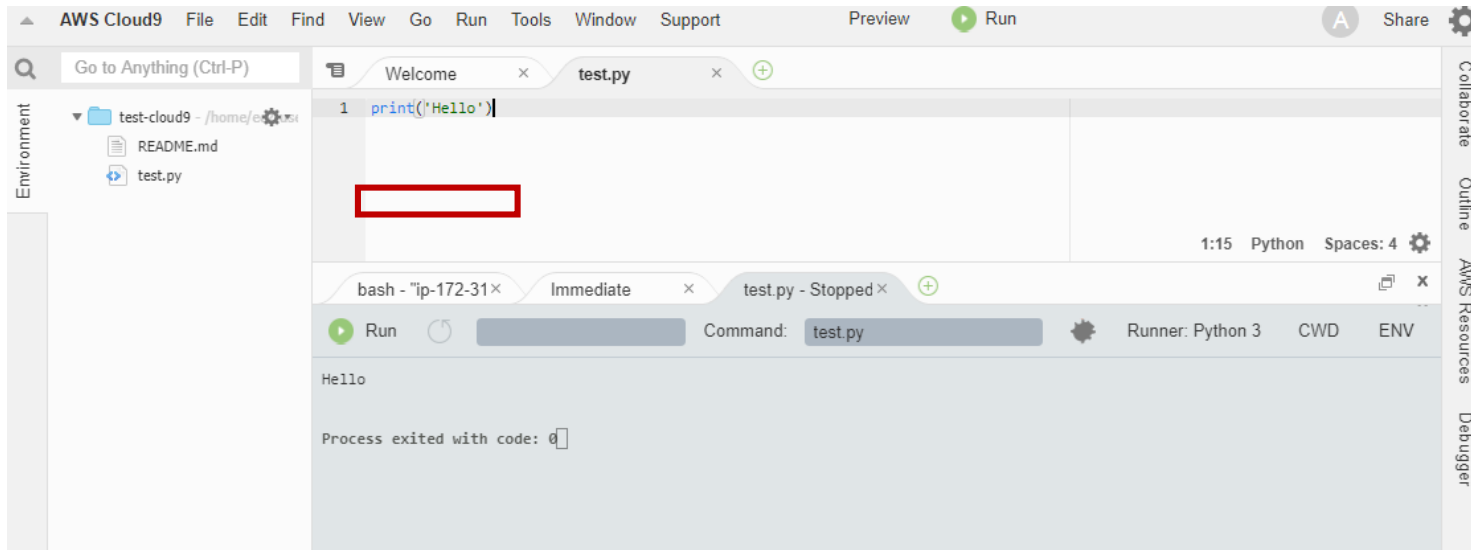
※ファイルの末尾は必ず「.py」

2. 1で作成したファイルをダブルクリックしてファイルを開く



3. コードを入力後、右上の[Run]を押すとPythonのプログラムが実行される

```
print('hello')
```

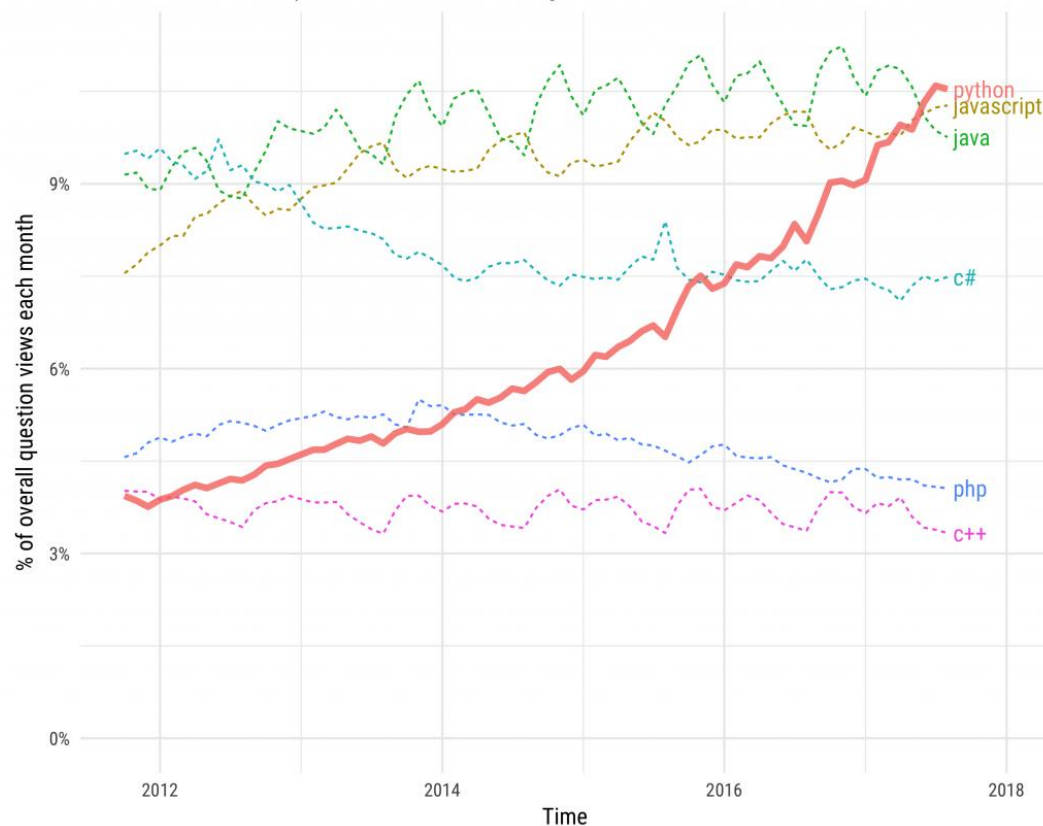


これでPythonを使う準備は完了

- ・近年人気が上昇しているプログラミング言語
- ・ライブラリが豊富で色々なことが簡単にできる
- ・AIの作成においては一番人気の言語

Growth of major programming languages

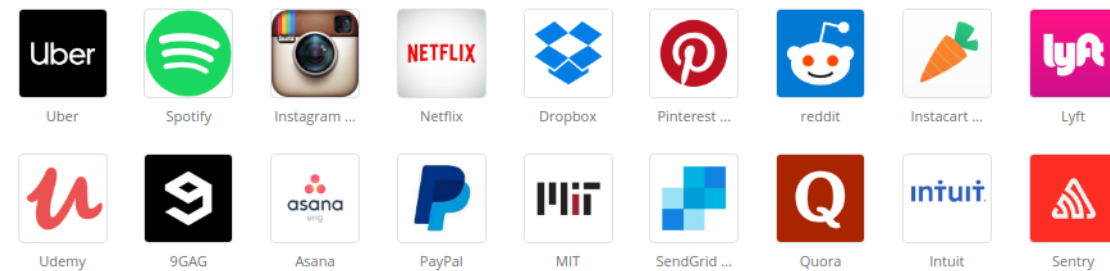
Based on Stack Overflow question views in World Bank high-income countries



Who uses Python?

COMPANIES

2797 companies use Python in their tech stacks, including Uber, Spotify, and Instagram.



プログラミング言語の種別

- ・プログラムが動く場所で以下の2つに大別できる



JavaScript, Swift等
(HTML, CSSも)



Python, Java, COBOL等

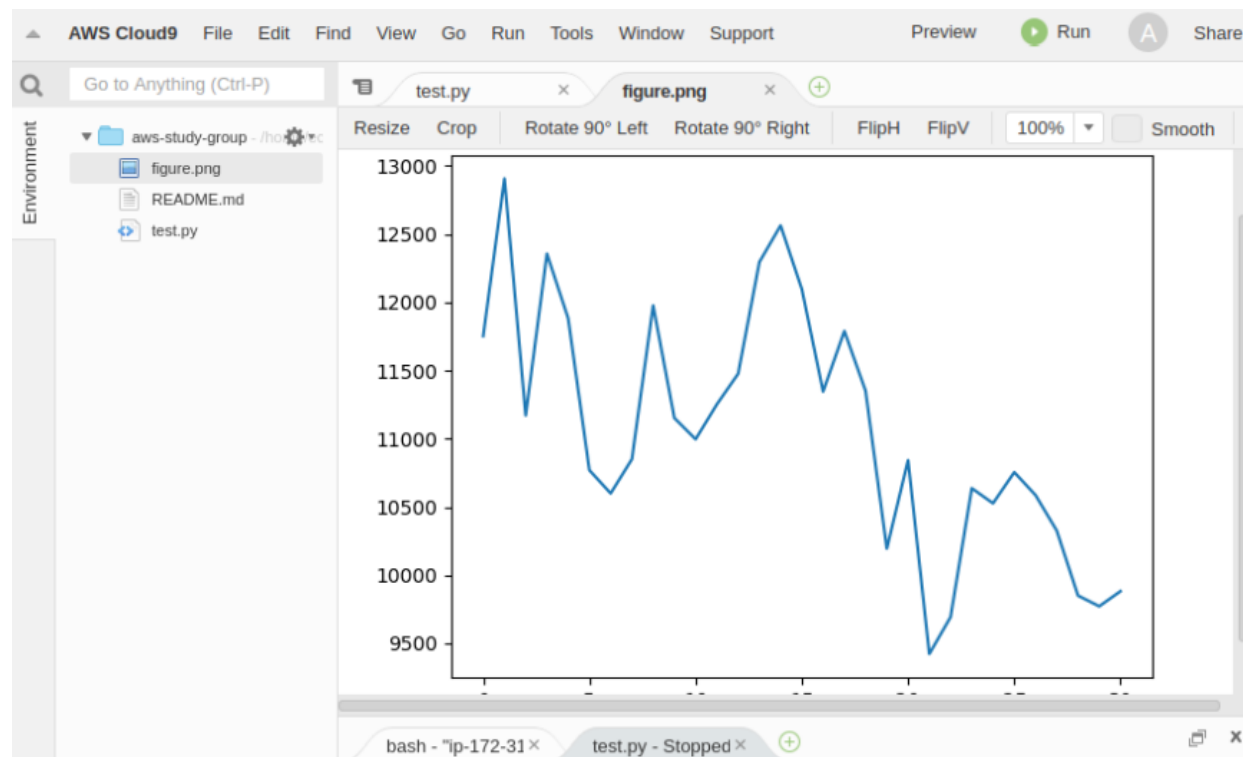
今日のゴールの再確認

グラフを作成するには以下の3ステップが必要

①APIを呼び出す

②①で取得したデータを処理してグラフ描画用の**データを作る**

③②で作成したデータを使って**グラフを作る**



APIとは？

ひとことでいうと、ある機能やデータにアクセスするための方法（どうやって使うか）を定めたもの

三菱UFJ国際投信 投信APIの例

こうやってアクセスすると

こういうデータが取得できる

3. ファンド情報 Web API 仕様

1). 最新ファンド情報（全ファンド）

全ファンドの最新情報を返します

Request

GET

/fund_information_all_latest

サンプル URI

GET /fund_information_all_latest

Parameters

パラメータ	データ型	指定有無	値	サンプル
fund_information_all_latest	String	必須	固定値	固定値

Response

HTTP/1.1 200

Header

Content-Type: application/json

Body

```
{
  "@odata.context":
"http://developer.am.mufig.jp/api/api.rsc/$metadata#fund_information_all_latest",
  "value": [
    {...},
    {...},
    {...},
    {
      nav_min_3y": "15282",
      "netassets_change_cmp_prev_day": "-0.98",
      "risk_min_full": "1.30",
      "nav_min_1y_dt": "20161109",
      "risk_min_1m": "2.55",
      "percentage_change_1m": "4.71",
      "percentage_change_max_full": "123.82",
```

①APIを呼び出す

APIとは？ 具体例1

以下の例ではAPIを通じてYoutubeのデータを利用できる

Youtube Data API

 YouTube > Data API

[ホーム](#) [ガイド](#) [リファレンス](#) [サンプル](#) [サポート](#)

Overview

- ▶ アクティビティ (Activities)
- ▶ Captions
- ▶ チャンネルバナー (ChannelBanners)
- ▶ チャンネル (Channels)
- ▶ ChannelSections
- ▶ Comments
- ▶ CommentThreads
- ▶ ガイドカテゴリ (GuideCategories)
- ▶ 再生リストの項目 (PlaylistItems)
- ▶ 再生リスト (Playlists)
- ▶ 検索 (Search)
- ▶ サブカテゴリ (Subscriptions)
- ▶ サムネイル (Thumbnails)
- ▶ VideoAbuseReportReasons
- ▶ 動画カテゴリ (VideoCategories)
- ▶ 動画 (Videos)
- ▶ ウォーターマーク (Watermarks)
- ▶ 標準クエリパラメータ (英語版)
- ▶ エラー

activities

activity リソースには、特定のチャンネルまたはユーザーが、YouTube で行った操作に関する情報が格納されています。アクティビティ フィードに報告される操作には、動画の評価、動画の共有、お気に入りへの動画の追加、動画へのコメントの投稿、動画のアップロードなどがあります。各 **activity** リソースは、操作の種類、操作に関連付けられたチャンネル、および操作に関連付けられたリソース（評価またはアップロードされた動画など）を識別します。

このリソースの詳細については、対応する [リソース表現](#)と[プロパティ](#)のリストを参照してください。

メソッド	HTTP リクエスト	説明
https://www.googleapis.com/youtube/v3 に対する相対 URI		
list	GET /activities	リクエスト条件に一致するチャンネル アクティビティ イベントのリストを返します。たとえば、特定のチャンネルと関連付けられたイベント、ユーザーの登録チャンネルや Google+ フレンドに関連付けられたイベント、各ユーザーにカスタマイズされた YouTube トップページ フィードなどを取得できます。
insert	POST /activities	特定のチャンネルに対するお知らせメッセージを投稿します（リクエストを送信するユーザーは、チャンネルの代理で行動する権限を与えられている必要があります）。

注: **activity** リソースには、動画の評価やお気に入りへの動画の追加といった操作に関する情報が含まれますが、これらの **activity** リソースを生成するには別の API メソッドを使う必要があります。たとえば動画の評価は API の [videos.rate\(\)](#) メソッド、お気に入りへの動画の追加は [playlistItems.insert\(\)](#) メソッドを使います。

①APIを呼び出す

APIとは？具体例2

以下の例ではAPIを通じて「画像のテキスト化」という機能を利用できる

Google Vision API

AIと機械学習プロダクト

お問い合わせ

VISION AI

概要

Vision API のデモ

利点

特長

導入事例

ユースケース

料金


リソース

トレーニング

使ってみる

Try the API

FacesObjectsLabelsWebTextPropertiesSafe Search



page4.png

+Page 1

+Block 1

COMMUNITY

+Block 2

選択肢について検討する Chris Berry氏（手前）と Andy Glover氏。Netflixは自社のデータを使用して、購入または制作すべきコンテンツを決定しています。

+Block 3

JAVA IN ACTION

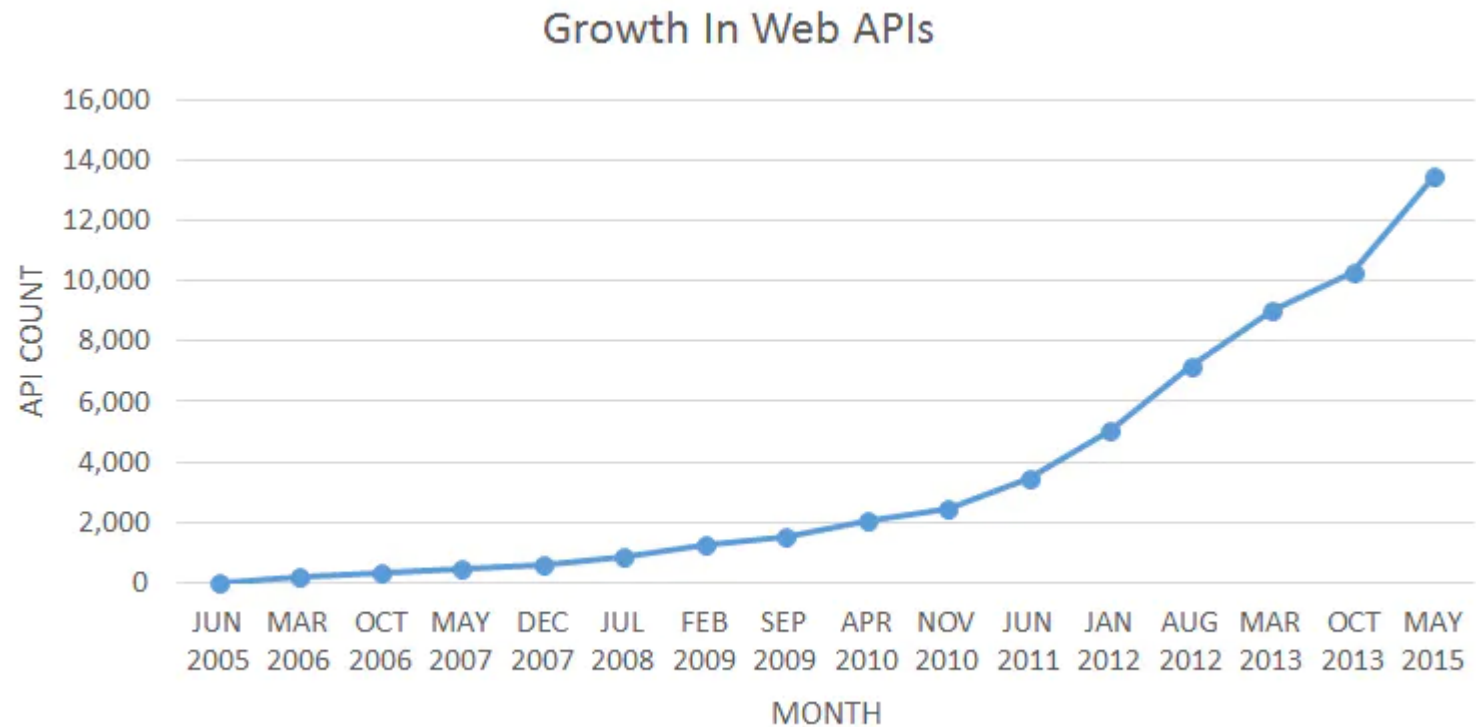
Show JSON ▼

RESET

NEW FILE

APIとは？

- 世の中でいわれているAPIはほとんどがWebAPI
- WebAPIとはWebの仕組みを使うことで自由に呼び出せるAPIのこと
- 以下の図の通りWeb APIの数は増え続けている



まずはブラウザでAPIを使ってみよう

以下にサンプルとしてすぐに使えるAPIを用意しました

- ・ Bitcoinの現在値を取得できるAPI

<https://www.coindesk.com/api>

- ・ ランダムで猫の画像のURLを取得できるAPI

<https://random.cat/view/1039>

- ・ ランダムで犬の画像のURLを取得できるAPI

<https://dog.ceo/>

- ・ 架空のユーザの情報(名前、メール、顔写真等)を取得できるAPI

<https://randomuser.me/>

①APIを呼び出す

ブラウザでAPIを使ってみよう

Googleを開くときもAPIを呼ぶときもやっていることは全く同じGET Request
違うのはサーバ側で返されるデータの目的

	<u>Google.co.jp</u>	<u>Cat API</u>
形態	Webページ	Web API
Request	GET / HTTP/1.1 Host: www.google.co.jp	GET /meow HTTP/1.1 Host:aws.random.cat
Response	HTML形式 人が見るためのデータ	JSON形式 機械が処理するためのデータ

APIがあると何がうれしい？



データや機能がほしくなったときにどこからでも使うことができる

PythonでAPIを呼ぶ

以下のコードを実行すればAPIの呼び出しができます。
まずは実行してみましょう。

```
import requests

URL =
'https://api.coindesk.com/v1/bpi/currentprice.json'

response = requests.get(URL)
print(response.text)
```


変数と値

```
URL = 'https://api.coindesk.com/v1/bpi/currentprice.json'
```

変数(Variable)

値(Value)

- ・ URLという変数を定義して、変数にURLの文字列を代入(Assign)している
- ・ Pythonでは変数の定義時にデータの型を指定する必要がない
- ・ `type()`でデータの型を調べられる

今日登場するデータの型

具体例

文字列
(str)

name = 'John'

数値
(int)

number = 15

辞書
(dictionary)

data = {'name': 'John'}

リスト
(list)

data = [1, 2, 3, 4, 5]

関数 (Function)

```
response = requests.get(URL)
```

- requestsというモジュールの`get()`という関数(Function)を呼び出すことでAPIコールを行っている
- 関数(Function)とは何らかの機能呼び出せるようにしたもの
- 関数で重要な点は中身でどんなことをやっているのか知らなくても使えること

関数 (Function)

```
def add(x, y):  
    return x + y  
  
a = 5  
b = 10  
result = add(a, b)  
print(result) # 15
```

引数(Argument)

… 関数に渡す値

返り値(Return Value)

… 関数から反ってくる値



①APIを呼び出す

APIで取得した値の確認

```
print(response.text)
```

- `.text`でresponseの文字列(str)にアクセスできる
- `type()`を使ってデータの型を調べてみよう

演習問題1

Bitcoinの過去1か月間の価格を取得できるAPIを呼んで、データをCloud9上に表示してみよう。

URLは以下です。

<https://api.coindesk.com/v1/bpi/historical/close.json>

①APIを呼び出す

(参考) Pythonにおけるライブラリ

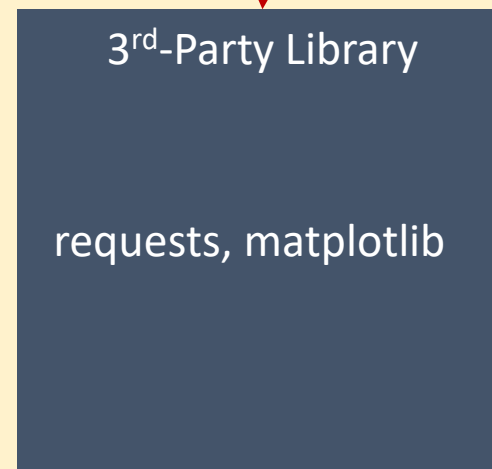
Pythonの世界



何もせずに使えるもの

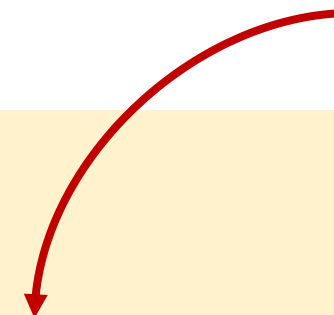


Importが必要なもの



Installが必要なもの

Download
& Install



APIで取得した値の処理

Bitcoinの現在値を取得するAPI(<https://api.coindesk.com/v1/bpi/currentprice.json>)
を例にとって、APIで取得したデータにどうアクセスするかを学んでいく。

USドルの現在価格が欲しい

```
{
  "time": {
    "updated": "Jul 29, 2019 08:09:00 UTC",
    "updatedISO": "2019-07-29T08:09:00+00:00",
    "updateduk": "Jul 29, 2019 at 09:09 BST"
  },
  "disclaimer": "This data was produced from the CoinDesk Bitcoin Price Index (USD). Non-USD currency data converted using hourly conversion rate from openexchangerates.org",
  "chartName": "Bitcoin",
  "bpi": {
    "USD": {
      "code": "USD",
      "symbol": "$",
      "rate": "9,604.7050",
      "description": "United States Dollar",
      "rate_float": 9604.705,
      "GBP": {
        "code": "GBP",
        "symbol": "£",
        "rate": "7,785.0936",
        "description": "British Pound Sterling",
        "rate_float": 7785.0936
      },
      "EUR": {
        "code": "EUR",
        "symbol": "€",
        "rate": "8,637.1462",
        "description": "Euro",
        "rate_float": 8637.1462
      }
    }
  }
}
```


APIで取得できるデータ形式

- APIで返される値の形式は**JSON(JavaScript Object Notation)**がデファクトスタンダード
- 以前はXML形式が一般的だったが、今は衰退

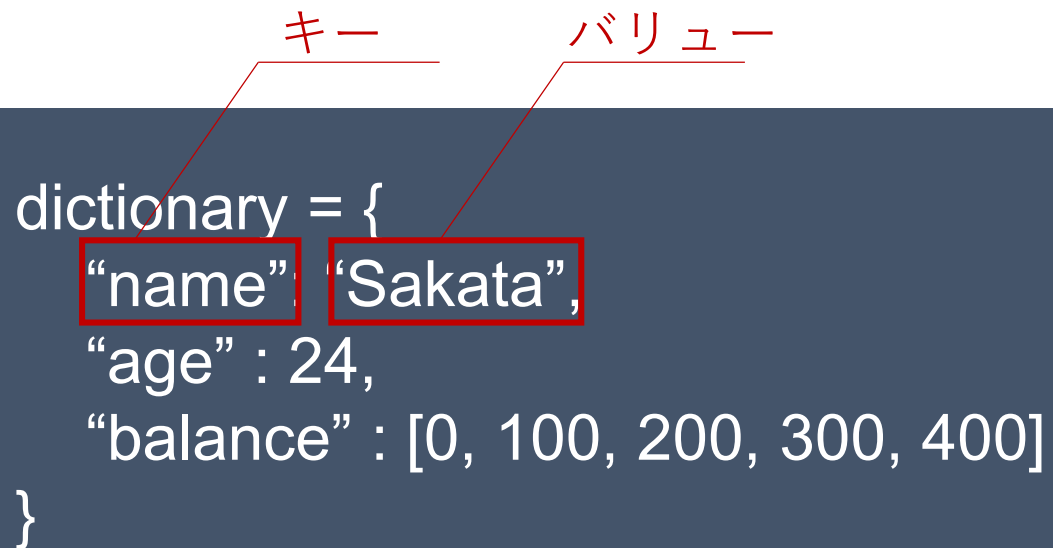
Google Trendで見るAPIのデータ形式の人気推移



JSONとは

データを表現するための記法（決まり事）。
JSONという共通の記法を使うことで、言語に関わらずデータのやり取りができるようになる。

JSONではキーとバリューでデータを表す。
また、数値や配列などを組み合わせることで複雑なデータを表現できる。

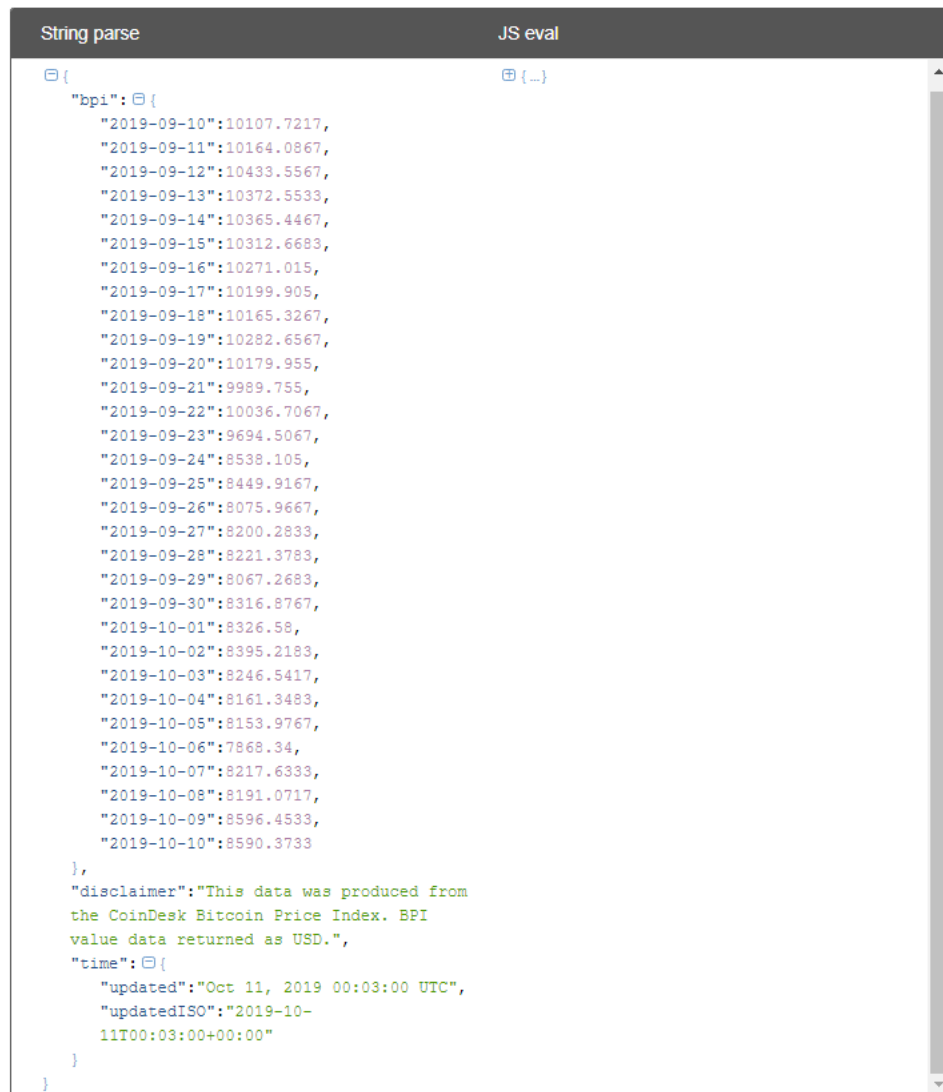


```
dictionary = {  
  "name": "Sakata",  
  "age": 24,  
  "balance": [0, 100, 200, 300, 400]  
}
```

The diagram illustrates the structure of a JSON object. It shows a dictionary with three key-value pairs. The first pair is "name": "Sakata". A red line points from the label "キー" (Key) to the string "name", and another red line points from the label "バリュー" (Value) to the string "Sakata". The other two pairs are "age": 24 and "balance": [0, 100, 200, 300, 400].

今回取得したデータの形式を確認

JSONデータは構造が複雑でわかりづらかったりするので、
JSON Parserを使って構造を理解しよう(Googleで「json parser」で検索)



The screenshot shows a JSON parser interface with two tabs: "String parse" and "JS eval". The "String parse" tab is active, displaying a JSON object. The "JS eval" tab shows a simplified representation of the object. The JSON object contains a "bpi" array of price data points, a "disclaimer" string, and a "time" object with "updated" and "updatedISO" fields.

```
{
  "bpi": [
    {
      "2019-09-10": 10107.7217,
      "2019-09-11": 10164.0867,
      "2019-09-12": 10433.5567,
      "2019-09-13": 10372.5533,
      "2019-09-14": 10365.4467,
      "2019-09-15": 10312.6683,
      "2019-09-16": 10271.015,
      "2019-09-17": 10199.905,
      "2019-09-18": 10165.3267,
      "2019-09-19": 10282.6567,
      "2019-09-20": 10179.955,
      "2019-09-21": 9989.755,
      "2019-09-22": 10036.7067,
      "2019-09-23": 9694.5067,
      "2019-09-24": 8538.105,
      "2019-09-25": 8449.9167,
      "2019-09-26": 8075.9667,
      "2019-09-27": 8200.2833,
      "2019-09-28": 8221.3783,
      "2019-09-29": 8067.2683,
      "2019-09-30": 8316.8767,
      "2019-10-01": 8326.58,
      "2019-10-02": 8395.2183,
      "2019-10-03": 8246.5417,
      "2019-10-04": 8161.3483,
      "2019-10-05": 8153.9767,
      "2019-10-06": 7868.34,
      "2019-10-07": 8217.6333,
      "2019-10-08": 8191.0717,
      "2019-10-09": 8596.4533,
      "2019-10-10": 8590.3733
    }
  ],
  "disclaimer": "This data was produced from the CoinDesk Bitcoin Price Index. BPI value data returned as USD.",
  "time": {
    "updated": "Oct 11, 2019 00:03:00 UTC",
    "updatedISO": "2019-10-11T00:03:00+00:00"
  }
}
```

PythonでどうやってJSONの値にアクセスするか

- ・ JSONはPythonの中では**単なる文字列(str)**として表現される
- ・ 文字列のままでは必要な情報にアクセスできないので、**変換が必要**
- ・ PythonではJSONを**Dictionary**に変換することで値へのアクセスが可能になる

JSONからDictionaryへの変換

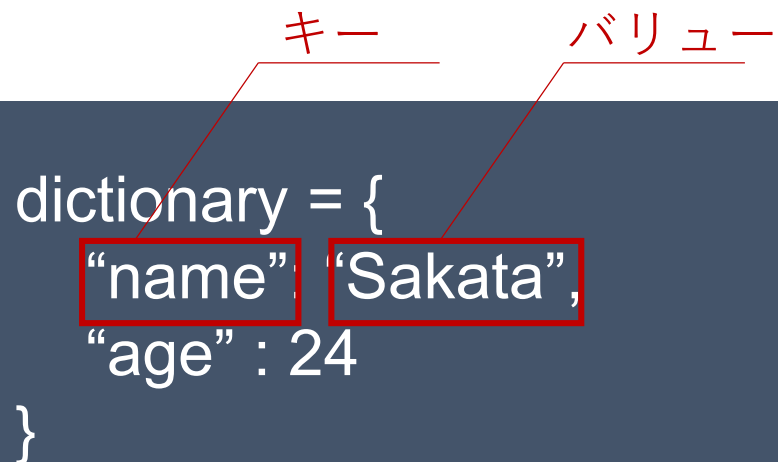
文字列(str)をアクセス可能なデータに変換するには、`json`というライブラリを使います。

```
import json
```

```
data = json.loads(response.text)  
type(data)
```

Dictionaryとは

Dictionaryとはキーとバリューでデータを表す形式。
JSONとデータの表し方がほぼ一緒。



```
dictionary = {  
    "name": "Sakata",  
    "age": 24  
}
```

The diagram illustrates a Python dictionary. The key "name" is highlighted with a red box and labeled "キー" (Key) with a red line pointing to it. The value "Sakata" is also highlighted with a red box and labeled "バリュー" (Value) with a red line pointing to it.

キーを指定することでバリューを取得できる

```
>>> dictionary['name']  
Sakata
```

ネストしたDictionary

バリューにはPythonのどんなデータも使える。
そのため、以下の例のようにバリューにDictionaryを使うこともできる。

```
sample_dict= {  
    "person": {  
        "name": "Sakata",  
        "age": 24  
    },  
    "is_valid": True  
}
```

Listとは

Listとは複数の要素を保存することができるコンテナのようなデータ型。要素を順番で持つので、時系列データを保存する場合に適している。

```
sample_list = ['Sakata', 'Imai']
```

データにアクセスするにはインデックスを使う

```
>>> sample_list[0]  
Sakata  
>>> sample_list[1]  
Imai
```


Listに要素を追加

Listには`append()`というメソッドがあり、
これを使うと指定した要素をListの末尾に追加できます。

```
>>> sample_list = []  
>>> sample_list.append(10)  
>>> sample_list  
[10]  
  
>>> sample_list.append(20)  
>>> sample_list  
[10, 20]
```

繰り返しの実行

Pythonで繰り返し処理を実行するにはfor文かwhile文のどちらかを使います。

```
>>> for i in range(5):  
    print(i)  
0  
1  
2  
3  
4
```

Listに対する繰り返しの実行

Listに対してfor文を実行すると要素の数だけループが実行されます。

```
>>> sample_list = ['Sakata', 'Imai']  
>>> for data in sample_list:  
>>>     print(data)  
'Sakata'  
'Imai'
```

Dictionaryに対する繰り返しの実行

Dictionaryに対してfor文を実行するとキーの数だけループが実行されます。

```
>>> sample_dict = {  
    "name": "Sakata",  
    "age" : 24  
}  
>>> for data in sample_dict:  
>>>     print(data)  
Sakata  
Imai
```

Dictionaryに対する繰り返しの実行2

Dictionaryに対して.items()を使うとキーとバリューにアクセスできます。

```
>>> sample_dict = {  
    "name": "Sakata",  
    "age": 24  
}  
>>> for key, value in sample_dict.items():  
>>>     print(key, value)  
name Sakata  
age Imai
```

演習問題2

演習問題1で取得したAPIの結果から、時系列のBitcoin価格を格納したListを作成しよう

↓ 結果のイメージ

```
[11882.5083, 10769.4217, 10599.3267, 10850.4, 11976.6317, 11151.9183,  
10996.4267, 11252.4533, 11476.4083, 12294.6567, 12563.215, 12096.6533,  
11345.7717, 11789.3367, 11350.6283, 10194.21, 10842.4517, 9422.4517,  
9692.7067, 10636.91, 10526.3917, 10754.29, 10586.2433, 10325.825, 9849.82,  
9771.5667, 9882.1, 9844.5583, 9466.2717, 9527.7717, 9501.0317]
```

グラフ作成用ライブラリの使用

グラフを作成するためのライブラリにはいくつか種類がありますが、今回は**matplotlib**というものをつかいます。
まずは以下のコマンドをコンソールに入力してライブラリをインストールします。

```
pip install matplotlib --user
```

matplotlibの使い方

折れ線グラフであれば以下のコードで簡単に作れる。
※`plt.plot()`で折れ線グラフのx軸とy軸の値を指定

```
import matplotlib.pyplot as plt

x = range(5)
y = [10, 20, 50, 30, 20]

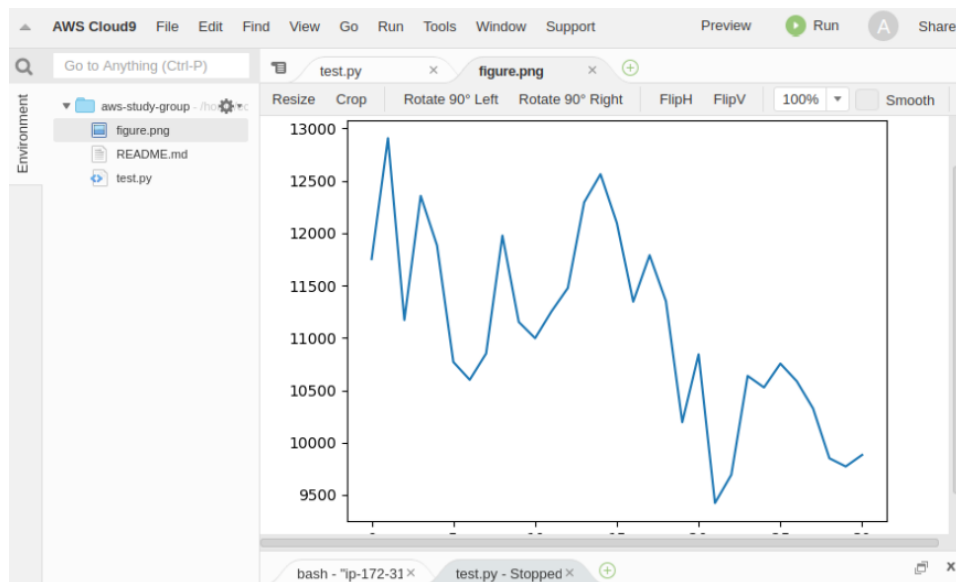
plt.plot(x, y)
plt.savefig('sample.png')
```


③グラフの作成

演習問題3(最後)

演習問題2で作成したBitcoin価格のListからグラフを作成してみよう

↓ 結果のイメージ



勉強会のまとめ

- ・ APIがどんなもので、どんなメリットがあるか
- ・ Pythonという言語の概要や基本的な文法
- ・ PythonでAPIを呼んで、データを処理する方法

Pythonの活用例



ライブラリ
(Flask)



Webサーバ
(API)



ライブラリ
(BeautifulSoup)



スクレイピング



ライブラリ
(Numpy)



金融の計算



ライブラリ
(Scikit-learn)



AI(機械学習)

参考資料

- ・ AWSクラウド活用資料集



日本担当チームへお問い合わせ サポート 日本語 ▼ アカウント ▼ [コンソールにサインイン](#)

製品 ソリューション 料金 ドキュメント 学習 パートナー AWS Marketplace その他 🔍

AWS クラウドサービス活用資料集トップ

アマゾン ウェブ サービス (AWS) は安全なクラウドサービスプラットフォームで、ビジネスのスケールと成長をサポートする処理能力、データベースストレージ、およびその他多種多様な機能を提供します。お客様は必要なサービスを選択し、必要な分だけご利用いただけます。それらを活用するために役立つ日本語資料、動画コンテンツを多数ご提供しております。(本サイトは主に、AWS Webinar で使用した資料およびオンデマンドセミナー情報を掲載しています。)

[AWS Webinar お申込 »](#)[AWS 初心者向け »](#)[業種・ソリューション別資料 »](#)[サービス別資料 »](#)



日本担当チームへお問い合わせ サポート 日本語 ▼ アカウント ▼ [コンソールにサインイン](#)

製品 ソリューション 料金 ドキュメント 学習 パートナー AWS Marketplace その他 🔍

Compute

Service Name	Published Date	Title				
Amazon EC2	2019/03/05	Amazon EC2	SlideShare	PDF	Youtube	
Amazon EC2	2016/04/20	Amazon EC2 - Windows	SlideShare	PDF		
Amazon EC2	2016/11/22	HPC分野でのAWS活用	SlideShare	PDF		
Amazon EC2	2017/11/10	AWSのコスト最適化/リザーブドインスタンス	SlideShare	PDF		
Amazon EC2	2019/03/06	Amazon EC2スポットインスタンス	SlideShare	PDF	Youtube	
Amazon EC2 Auto Scaling	2017/03/15	Auto Scaling	SlideShare	PDF		

AWSクラウド活用資料集トップ

<https://aws.amazon.com/jp/aws-jp-introduction/>

EC2

https://d1.awsstatic.com/webinars/jp/pdf/services/20190305_AWS-Blackbelt-EC2.pdf

Cloud9

https://d1.awsstatic.com/webinars/jp/pdf/services/20180613_AWS-BlackBelt-Introducing-AWS-Cloud9.pdf

ところで、Cloud9を使うとき何が起きている？

EC2を見てみると

何かができている！

The screenshot displays the AWS Management Console's EC2 dashboard. On the left, a navigation menu lists various EC2-related features, with 'インスタンス' (Instances) highlighted. The main panel shows a table of EC2 instances. A single instance is listed and highlighted with a red border. The instance is named 'aws-cloud9-test-cloud9-6aaee34bbeeb41b...', has an ID of 'i-0c61b009b2caabc25', is of type 't2.micro', located in 'ap-northeast-1a', and is in a 'running' state. The console also shows buttons for 'インスタンスの作成' (Create Instance), '接続' (Connect), and 'アクション' (Actions).

Name	インスタンス ID	インスタンスタイプ	アベイラビリティゾーン	インスタンスの状態	ステータスチェック	アラーム
aws-cloud9-test-cloud9-6aaee34bbeeb41b...	i-0c61b009b2caabc25	t2.micro	ap-northeast-1a	running	2/2 のチェック...	なし

EC2とは

Amazon EC2とは

- 数分で起動し、1時間または秒単位の従量課金で利用可能なAWSクラウド上の仮想サーバー
- サーバーの追加・削除、マシンスペック変更も数分で可能
- 管理者権限(root / Administrator) で利用可能

EC2インスタンス

Guest 1

Guest 2

Guest n

Hypervisor

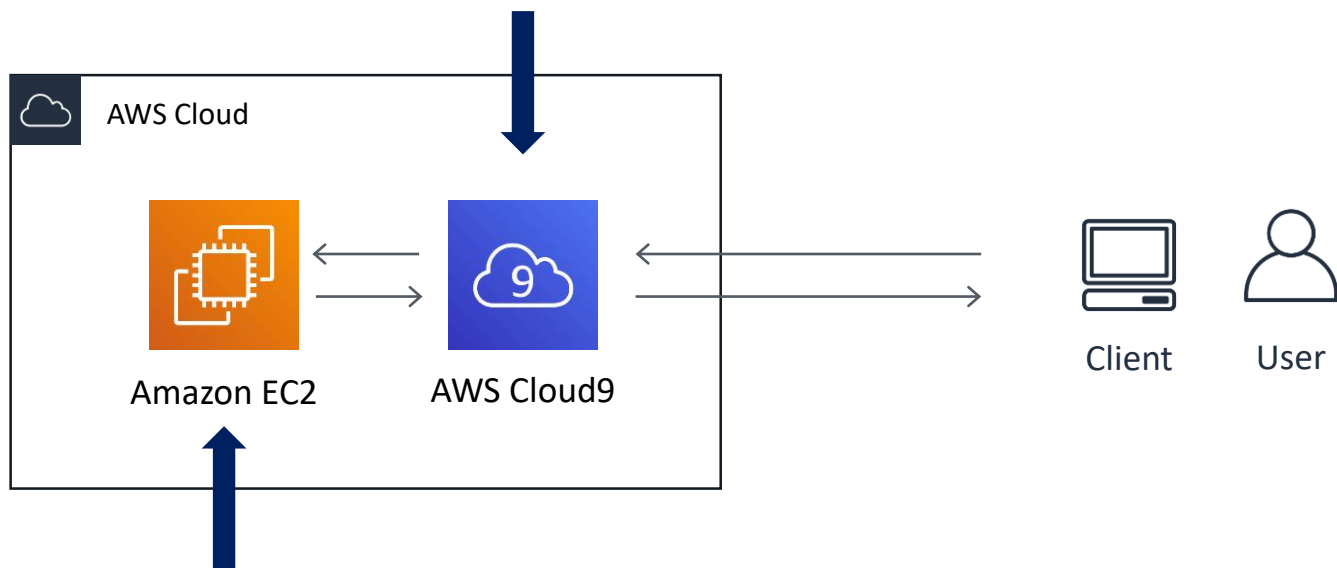
Host Server

EC2はElastic Compute Cloudの頭文字

つまり、EC2=AWS上に立てられるサーバ

Cloud9=AWS上のサーバ(EC2)を使って開発環境を作ってくれるツール

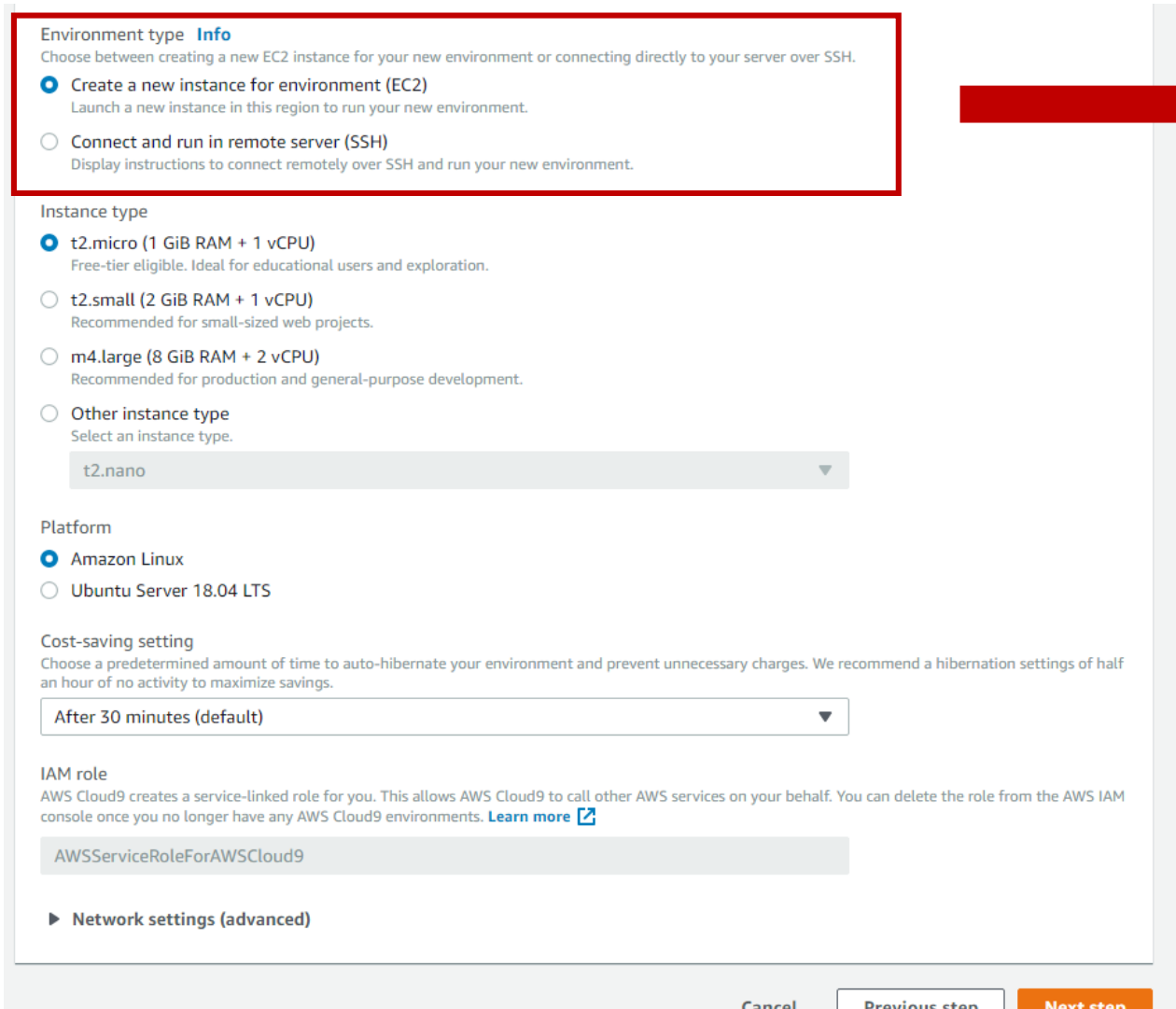
ブラウザでのアクセスを可能にしてくれる



実際のプログラムはここで動いている

実はサーバ(EC2)ができていたのは自分で選択したから

Cloud9作成時の画面



Environment type [Info](#)
Choose between creating a new EC2 instance for your new environment or connecting directly to your server over SSH.

- ☒ **Create a new instance for environment (EC2)**
Launch a new instance in this region to run your new environment.
- ☐ **Connect and run in remote server (SSH)**
Display instructions to connect remotely over SSH and run your new environment.

Instance type

- ☒ **t2.micro (1 GiB RAM + 1 vCPU)**
Free-tier eligible. Ideal for educational users and exploration.
- ☐ **t2.small (2 GiB RAM + 1 vCPU)**
Recommended for small-sized web projects.
- ☐ **m4.large (8 GiB RAM + 2 vCPU)**
Recommended for production and general-purpose development.
- ☐ **Other instance type**
Select an instance type.
t2.nano

Platform

- ☒ **Amazon Linux**
- ☐ **Ubuntu Server 18.04 LTS**

Cost-saving setting
Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default)

IAM role
AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

► **Network settings (advanced)**

Cancel Previous step Next step

ここで選択していたのは以下の二択

- ・新しくサーバ(EC2)を立てて開発環境にする
- ・既存のサーバを開発環境にする