python™ プログラミング勉強会

公式サイトから資料を
DLできます。
パスワード: pk_py1214
puzzleknot.org

puzzleknot
Tohoku University Programming Club

バックスラッシュと¥マーク

- Windowsでは、一般の文字入力ではバックスラッシュ\は¥マークとして表示されます
- PowerPointくんは区別してくれないので資料のコードは残念ながら¥記号になっています(コピペする際はブログの記事からやるのがおすすめです)

2

バックスラッシュと¥マーク

- なんで？
  http://juangotoh.hatenablog.com/entry/2016/06/13/104139

3

enumerate関数

- for文でlistなどを回す際、同時に”何周目であるか” を知りたい場合があります
- enumerateを使うと、楽に取得できます

4

enumerate関数

- a = [‘i’, ‘j’, ‘k’, ‘l’]
  ならば
  enumerate(a)は
  [(0, i), (1, j), (2, k), (3, l)] を返します
- [高度]enumerateはrangeと同じiterableなオブジェクトなので、そのままprintしてもよくわかりません

5

enumerate関数

- enumerate(a)は
  [(0, i), (1, j), (2, k), (3, l)]
- どう使うか？

```
for i, data in enumerate(a):
    print(i, data)
```

- for文はこんな感じで2つの数を受け取ることができます

6

enumerate関数

- これはアンパック代入と呼ばれるものの応用です
- a, b = [0, 1] のように、左辺に複数の変数、右辺にlistなどを書くと各要素が対応する左辺の変数に順番に代入されます(これを**アンパック代入**といいます)
- このとき、 aは0, bは1ですね

7

enumerate関数

- これはアンパック代入と呼ばれるものの応用です
- enumerate(a)は1周目に(0, a[0])のtupleを返すので、iには0が、dataにはa[0](つまり'i') がアンパック代入されます

8

enumerate関数

- 他にも、2つのlistを同時に回したい場合はzip関数で同じようなことができます
- https://python.civic-apps.com/zip-enumerate/

9

datetime モジュール

- 時間に応じて異なる処理をさせたり、時間の計算を行う際はdatetimeモジュールが有効です
- 公式チュートリアル https://docs.python.jp/3/library/datetime.html

10

datetime モジュール

```
import datetime

# 現在時刻の取得
print(datetime.datetime.today())

# 指定されたフォーマットに応じて文字列の表す時刻を取得
# フォーマットの指定のしかた
# https://docs.python.jp/3/library/datetime.html#strftime-strptime-behavior
string = "2017年01月01日00時00分00秒"
new_years_day = datetime.datetime.strptime(string, "%Y年%m月%d日%H時%M分%S秒")
print(new_years_day)
```

- 日付と時刻の組み合わせを扱うには datetime型です

11

datetime モジュール

```
import datetime

# 現在時刻の取得
print(datetime.datetime.today())

# 指定されたフォーマットに応じて文字列の表す時刻を取得
# フォーマットの指定のしかた
# https://docs.python.jp/3/library/datetime.html#strftime-strptime-behavior
string = "2017年01月01日00時00分00秒"
new_years_day = datetime.datetime.strptime(string, "%Y年%m月%d日%H時%M分%S秒")
print(new_years_day)
```

output例:
2017-12-17 23:19:53.824779
2017-01-01 00:00:00

12

datetime モジュール

```
elapsed_time = datetime.datetime.today() - new_years_day
print(type(elapsed_time)) # 時刻の差はtimedelta型で表されます
print("今年が始まってから{0}日が経ちました".format(elapsed_time.days))
```

- datetime型の差はtimedelta型で表されます
- timedelta型はdays, seconds, microsecondsという変数のアトリビュートを持っています
- そのため、.daysとつけるとそれは日付の差を表す値になります

13

datetime モジュール

```
elapsed_time = datetime.datetime.today() - new_years_day
print(type(elapsed_time)) # 時刻の差はtimedelta型で表されます
print("今年が始まってから{0}日が経ちました".format(elapsed_time.days))
```

output例:
<class 'datetime.timedelta'>
今年が始まってから350日が経ちました

14

datetime モジュール

- [高度]
  サーバー上で実行させる際は、
  タイムゾーンの違いに気をつけて
  ください
- タイムゾーン情報を持つ datetime 型
  オブジェクトは”aware”といわれ、
  持たない”naive”な datetime 型
  オブジェクトと区別されます
  （計算は両方ともawareかnaiveな
  場合に限る）

15

AWS Lambdaを用いたbotの自動化

- [高度]
  Twitterのbotは、やはり手放しで勝手
  に実行してほしいものです
- 決まった時間につぶやかせるように
  することは可能ですが(cronで検索)、
  PCを起動したままにしなければなり
  ません

16

AWS Lambdaを用いたbotの自動化

- [高度]
  リプに反応したりするbotなら別ですが、
  天気予報などのツイートのために常に
  PCを起動しっぱなしにしたり、
  サーバーを借りるのはもったいない
  気がします

- 毎日決まりきった時間につぶやかせたい場合は、Amazon Web Services(AWS)のLambdaを利用するのがおすすめです

17

AWS Lambdaを用いたbotの自動化

- [高度]
  軽く使い方を解説します
- AWSアカウントを作成し、ルート
  ユーザー ログイン
  https://aws.amazon.com/jp/
- 支払情報でクレジットカードを登録しておきます
  (ない人はVプリカなどを使いましょう)

18

AWS Lambdaを用いたbotの自動化

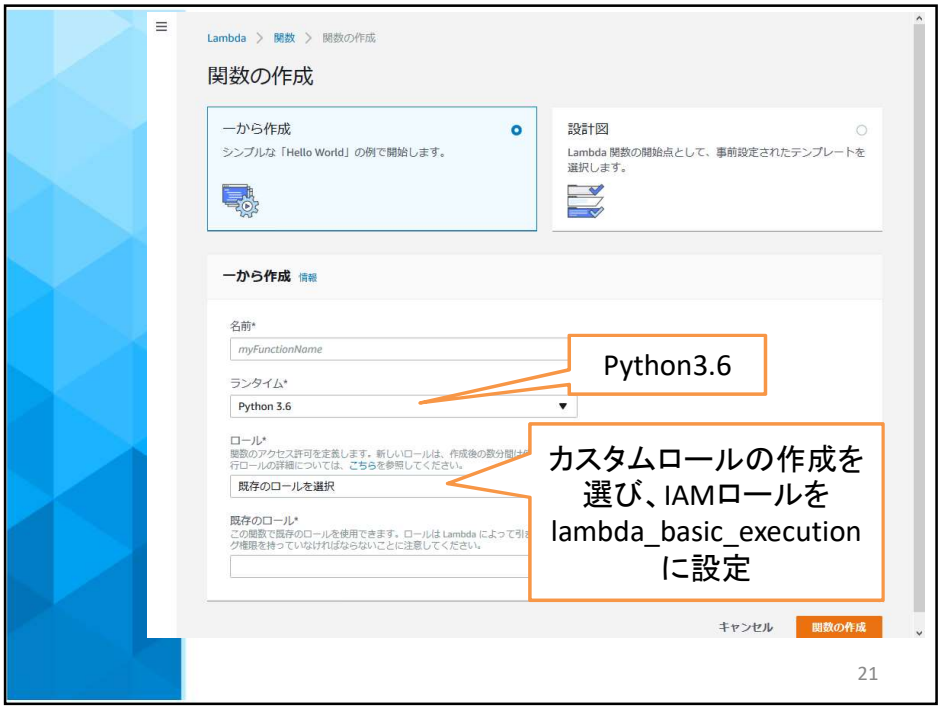
- [高度]
  ルートユーザーは不正アクセスされると余裕で数千万円を悪用されかねないので、2段階認証を有効化することを強く推奨します
- https://qiita.com/macoshita/items/06e2a2474bc8eca1a67d

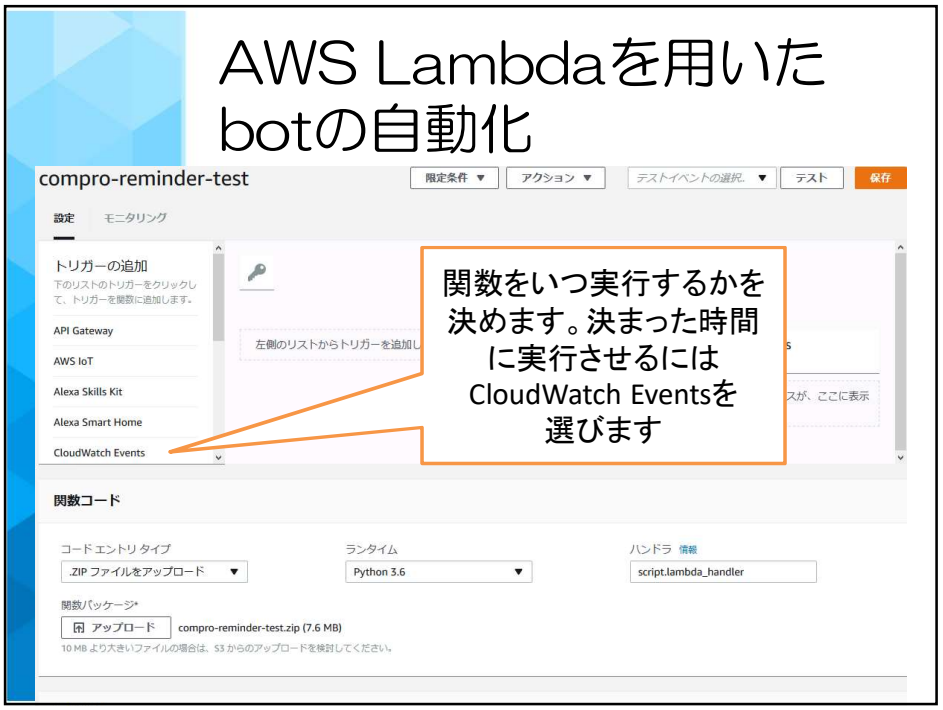
19

AWS Lambdaを用いたbotの自動化

- [高度]
  コンピューティングからLambdaを選択します
- 関数の作成をポチる

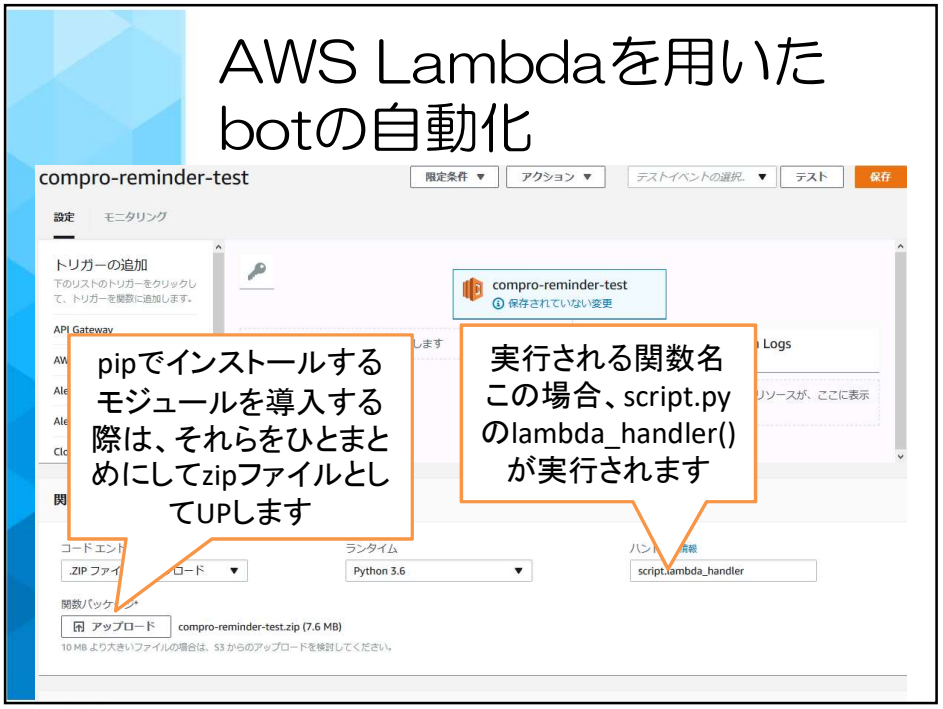
20


Lambda > 関数 > 関数の作成
関数の作成
一から作成
シンプルな「Hello World」の例で開始します。
設計図
Lambda 関数の開始点として、事前設定されたテンプレートを選択します。
一から作成 情報
名前*
myFunctionName
ランタイム*
Python 3.6
ロール*
既存のロールを選択
既存のロール*
キャンセル
関数の作成
Python3.6
カスタムロールの作成を選び、IAMロールをlambda_basic_executionに設定
21


AWS Lambdaを用いたbotの自動化
compro-reminder-test
限定条件
アクション
テストイベントの選択.
テスト
保存
設定
モニタリング
トリガーの追加
下のリストのトリガーをクリックして、トリガーを関数に追加します。
API Gateway
AWS IoT
Alexa Skills Kit
Alexa Smart Home
CloudWatch Events
関数をいつ実行するかを決めます。決まった時間に実行させるにはCloudWatch Eventsを選びます
関数コード
コード エントリ タイプ
.ZIP ファイルをアップロード
ランタイム
Python 3.6
ハンドラ 情報
script.lambda_handler
関数パッケージ*
アップロード
compro-reminder-test.zip (7.6 MB)
10 MB より大きいファイルの場合は、S3 からのアップロードを検討してください。

AWS Lambdaを用いたbotの自動化

AWS Lambdaを用いたbotの自動化



AWS Lambdaを用いたbotの自動化

- [高度]
  pipを使いたい場合はこちらも参考にしてください
  https://qiita.com/Hironsan/items/0eb5578f3321c72637b4
- 時間がかかるスクリプトの場合は設定の下の方にある「タイムアウト」を伸ばしておきましょう

25

AWS Lambdaを用いたbotの自動化

- [高度]
  AWS Lambdaを用いたbotのスクリプト例です
  https://github.com/AokabiC/compro-reminder/blob/master/script.py
- アカウントは
  https://twitter.com/ComproReminder?lang=ja

26

やってみようの解答

- ブログに掲載しています
- http://puzzleknot.org/study_open/325/
- パスワード: pk_pycode

27

宣伝！

- アンケートにご協力ください

- https://goo.gl/forms/K8MgaVsEnsvM7mXk1

28