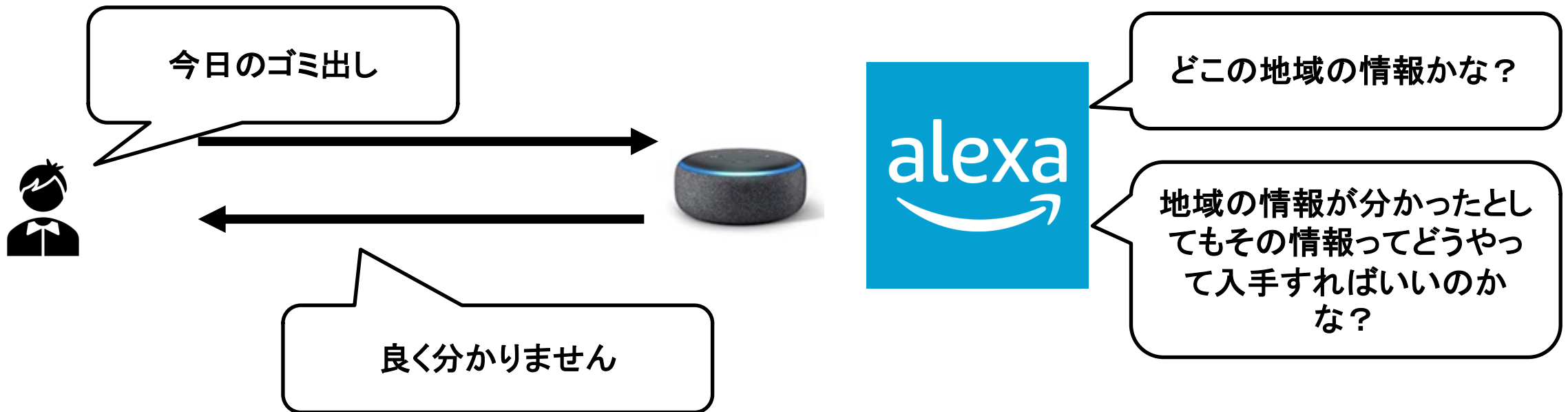


オリジナルAIを作ろう！

AI技術の基礎とプログラミング

Alexaについて

- Alexaには組み込まれている製品毎にできないことや分からない事もいっぱいあります



実際にコンソールを触ってみよう

・導入

Alexaには、みなさんが簡単にプログラミングをおこなって追加の機能を持たせる事ができます。

AlexaSkillと呼ばれていて、Alexa用の追加アプリケーションだと思えば分かり易いと思います。

※保護者の同意が必要ですが、アカウントの作成含め無料で使えるものです。

導入のイメージ

- Alexaスキルとは、AlexaがいろんなことをできるようにするためのAlexa専用のプログラム(アプリケーション)

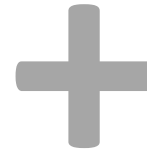


Alexaスキルの開発手順

対話モデル (音声UI)

バックエンドプログラム

発言の意図や
発言の揺れを
理解させるもの



人間の意図を
実行するもの



情報の提供

家電との連携



Alexa開発者コンソール



AWS Lambda

Alexaスキルの開発手順

対話モデル (音声UI)

このスキルは対話が必要なスキルではない為
発言の意図を
対話モデルの作成は不要
発言の振りを
理解させるもの

※「今日のゴミ出し」という発話で
スキルを呼び出せば良い為



Alexa開発者コンソール



バックエンドプログラム



ここだけ開発
してみよう！

人間の意図を
実行するもの

情報の提供
(ゴミ出しに関する情報)



AWS Lambda

学習内容の例

- ・コードを書き換えて「今日の給食スキルをつくってみよう！」

```
13
14 def handle(self, handler_input):
15     # Alexaの話す内容
16     speech = ""
17
18     # 今日の曜日を取得
19     japan_time = timezone(timedelta(hours=9), 'Asia/Tokyo')
20     weekday = datetime.now(tz=japan_time).isoweekday()
21
22     # 住んでいる地域のゴミの日に関するオープンデータを参照して、曜日に応じた自分の地域のゴミ種別を返す
23     # https://opendata.city.sagamihara.kanagawa.jp/dataset/gomisyusyu/resource/a0299f10-8ddd-49a1-b5ce-d44eedae540
24     if weekday == 1:
25         speech = "月曜日は容器包装プラスチックです。"
26     elif weekday == 2:
27         speech = "火曜日にはごみ収集はありません。"
28     elif weekday == 3:
29         speech = "水曜日は不燃ごみです。"
30     elif weekday == 4:
31         speech = "木曜日は缶・ビン・ペットボトルです。"
32     elif weekday == 5:
33         speech = "金曜日は資源ごみです"
34     elif weekday == 6:
35         speech = "土曜日のごみは一般ごみと乾電池です。"
36     elif weekday == 7:
37         speech = "日曜日にはごみ収集はありません。"
```

水曜日は不燃ごみ



水曜日はカレー
に書き換える

書き換える

学習内容の例

- ・コードを書き換えて「今日の給食スキルをつくってみよう！」

←スキル一覧 今日のごみ出し日 ビルド コードエディタ テスト 公開 認定 レポート

New File New Folder Delete Rename DynamoDB Database S3 Storage CloudWatch Logs Usage Integrate Download Skill Import Code Offline Tools Docs

Skill Code

- lambda
- lambda_function.py
- requirements.txt
- utils.py

```
lambda_function.py x
/ from ask_sdk_core.dispatch_components import AbstractRequestHandler, AbstractExceptionHandler
8
9 class ResponseHandler(AbstractRequestHandler):
10     """返答を返すための処理"""
11     def can_handle(self, handler_input):
12         return True
13
14     def handle(self, handler_input):
15         # Alexaの話す内容
16         speech = ""
17
18         # 今日の曜日を取得
19         japan_time = timezone(timedelta(hours=9), 'Asia/Tokyo')
20         weekday = datetime.now(tz=japan_time).isoweekday()
21
22         # 住んでいる地域のゴミの日にに関するオープンデータを参照して、曜日に応じた自分の地域のゴミ種別を返す
23         # https://opendata.city.sagamihara.kanagawa.jp/dataset/gomisisyuyu/resource/a0299f10-8ddd-49a1-b5ce-d44eeedae540
24         if weekday == 1:
25             speech = "月曜日は容器包装プラスチックです。"
26         elif weekday == 2:
27             speech = "火曜日はごみ収集はありません。"
28         elif weekday == 3:
29             speech = "水曜日はカレーです"
30         elif weekday == 4:
31             speech = "木曜日は缶・ビン・ペットボトルです。"
32         elif weekday == 5:
33             speech = "金曜日は資源ごみです"
34         elif weekday == 6:
35             speech = "土曜日のごみは一般ごみと乾電池です。"
36         elif weekday == 7:
37             speech = "日曜日にはごみ収集はありません。"
```

最終デプロイ日時

保存 デプロイ

書き換えたら
「保存」を押して
「デプロイ」を押す
(10~20秒ほどかかります)

学習内容の例

- ・コードを書き換えて「今日の給食スキルをつくってみよう！」

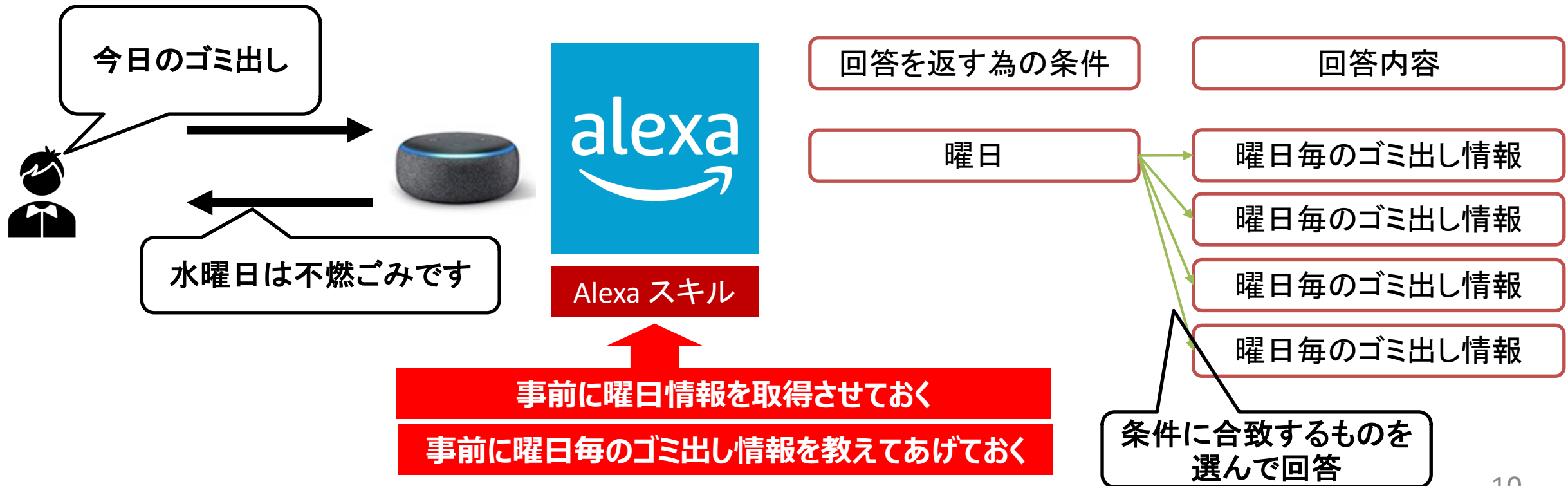
The screenshot shows the Alexa Developer Console interface. At the top, there's a search bar for developer help. Below it, a navigation bar contains tabs for 'スキル一覧' (Skill List), '今日の給食' (Today's Lunch), 'ビルド' (Build), 'コードエディタ' (Code Editor), 'テスト' (Test), '公開' (Publish), '認定' (Certify), and 'レポート' (Report). The main content area is divided into several sections. On the left, there's a 'Alexaシミュレータ' (Alexa Simulator) with a language dropdown set to '日本語' (Japanese) and a microphone icon. A simulated user input is shown as a blue bubble: '今日の給食' (Today's lunch). The skill's response is shown as a blue bubble: '水曜日はカレーです' (Wednesday is curry). Below the simulation, there's a 'パーソナライズのテスト' (Test Personalization) section. At the bottom, there are two code editors: 'JSON入力 1' (JSON Input 1) and 'JSON出力 1' (JSON Output 1), displaying the JSON data for the request and response respectively. The footer shows 'Japanese (日本語)' and a 'フィードバック' (Feedback) button.

テストしてみましょう

今日の給食が完成しました！

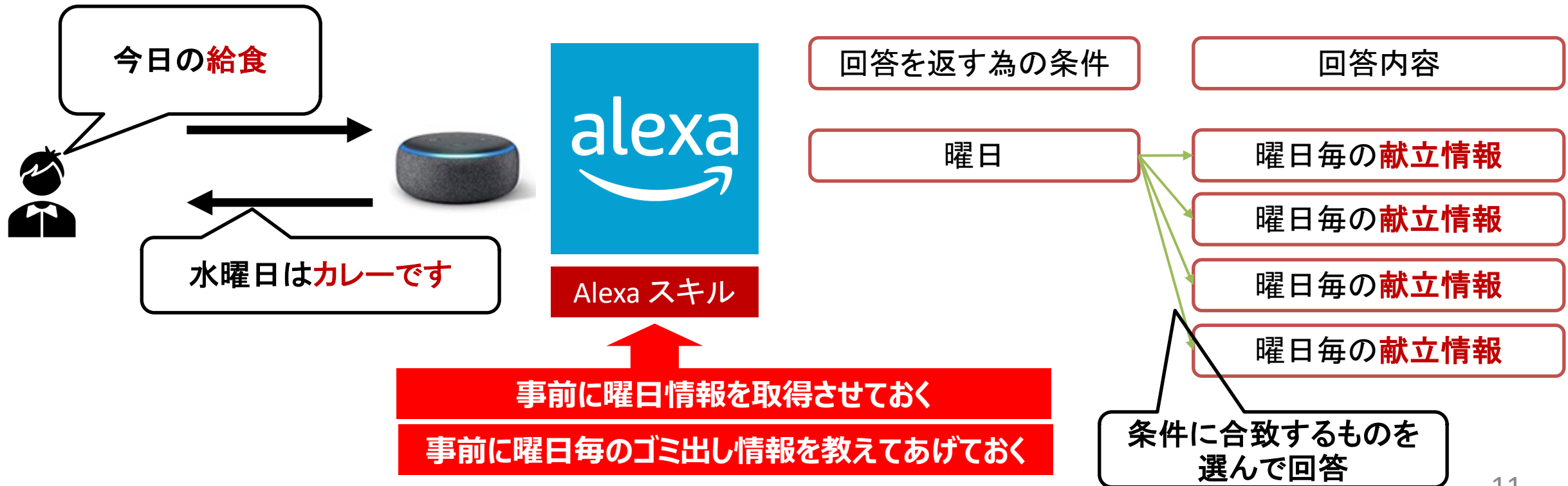
おさらい

- ・プログラミングって難しそうに見えますが形を理解してしまえば少し変化を加えることで違った機能を持たせる事ができます



おさらい

- ・プログラミングって難しそうに見えますが形を理解してしまえば少し変化を加えることで違った機能を持たせる事ができます



色々考えてみよう！

- ・プログラミングって難しそうに見えますが形を理解してしまえば少し変化を加えることで違った機能を持たせる事ができます

