

AI開発サロン



各種ご案内

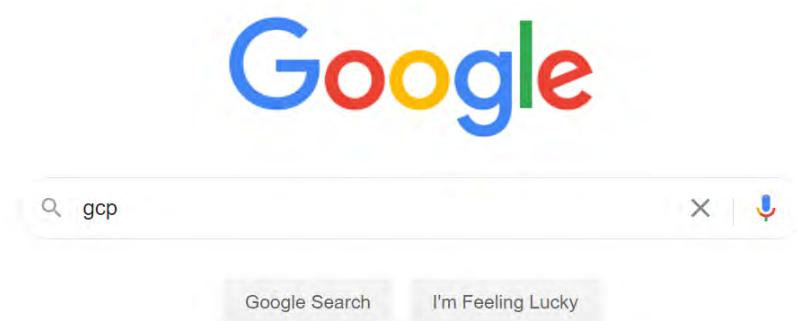
- 資料の閲覧ページについて
- Vimeoでのコメントについて
- アプリのダウンロードURLについて
 - CaptureTextアプリを使えばURLをコピーできる!!

先週の宿題

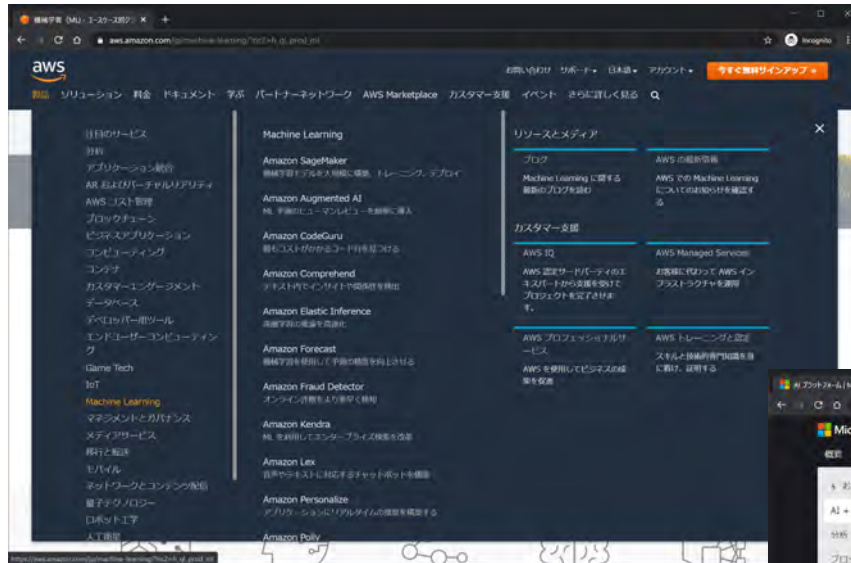
1. 利用できるAIのAPIにどのようなものがあるか調べよう。
2. どのようなアプリが開発できるか&便利か考えよう。

GCPにあるAIのAPIを見てみよう

<https://cloud.google.com/products#ai-and-machine-learning>

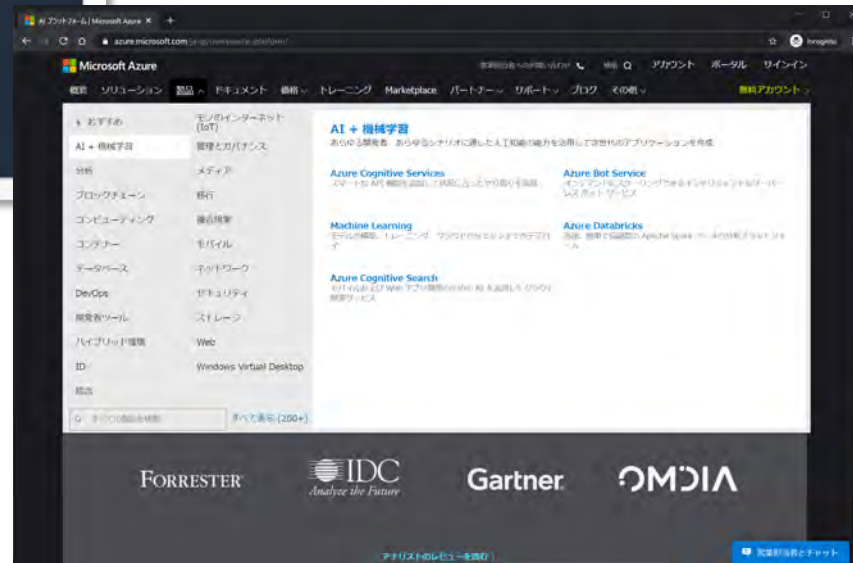


AWS・Azureも見てみよう



<https://aws.amazon.com/jp/machine-learning/>

<https://azure.microsoft.com/ja-jp/overview/ai-platform/>



<https://kentei.ai/salon/>

作れるアプリを考える

写真に撮ったものを読み上げてくれるアプリ

画像を写真に撮る



Vision API OCR

画像に写っている文字を認識して、テキストに変換してくれる。

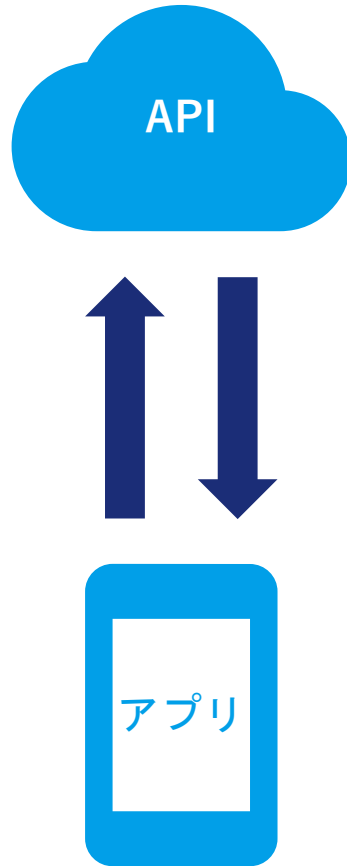


Text-To-Speech

テキストを読み上げてくれる。

APIを部品として組み合わせを考えると
アプリのアイデアができる

APIとは



アプリ(プログラム)を作るときに、
簡単に機能を追加できるようにしたもの。
≡ アプリを構成する部品

- クラウドと連携するもの/しないもの
- AIを使うもの/使わないもの

→ “部品”だから、いろいろなものがある

本日の目次

- **GCPのAI APIサービスを使う**
 - ・ サービスでできることを詳しく知る
 - ・ 料金・制限の調べ方
 - ・ APIを使いこなすための資料
- **フロントエンド開発とバックエンド開発**
 - ・ アプリ開発の作業を知る
 - ・ 開発に共通な知識の理解
- **CaptureTextアプリのコードを見てみよう**
 - ・ Windowsアプリの開発
 - ・ CaptureTextアプリの動作

AIアプリ開発に必要なスキル

企画	構築技術	マネタイズ
課題ドリブンのアイデア	AIのAPI	買い切り
市場調査	学習済みモデルの活用	サブスクリプション
UX設計	深層学習モデルの開発	フリーミアム
サービス設計	UI設計のパターン	アプリ内課金
プロジェクト管理	クライアントサイド開発	広告
	サーバーサイド開発	リリース方法
	DB	配信プラットフォーム
	開発言語/フレームワーク	
	IDE	

GCPのAI APIサービスを使う

- サービスでできることを詳しく知る
 - APIの提供する機能で自分が求めていることができるか確認する。
 - 自分のプログラムに組み込む方法が掲載されているか確認する。
- 料金・制限を調べる
 - APIを使用するときにかかる料金体系を理解する。
- APIを使いこなす
 - APIを使ったプログラムを作るときの細かな指定を調べる

サービスでできることを詳しく知る

1. 使いたいサービスを探す

- <https://cloud.google.com/products#ai-and-machine-learning>

2. 各サービスのドキュメントを探す

- <https://cloud.google.com/docs>

3. 「クイックスタート」で使い方の概要を見る

- <https://cloud.google.com/vision/docs/quickstart-client-libraries>

4. 「入門ガイド」でAPIでできることとプログラムの概要を知る

- <https://cloud.google.com/vision/docs/how-to>

料金・制限を調べる

- 「リソース」を見る
 - <https://cloud.google.com/vision/docs/resources>
 - 「料金」・「割り当てと上限」を主にチェック
- 「割り当てと上限」について
 - APIで用意されている機能の制限事項がまとめられている
 - 自分の作りたいことがきちんと行えるか確認
- クラウドでよくある料金体系
 - 回数: 1回の使用あたり...
 - 使った時間: 1時間あたり...
 - データ量: 転送したデータ量が1GBあたり...

詳しいAPIの使い方を調べる

- 「リファレンス」でプログラムを書く時の使い方を知る
 - <https://cloud.google.com/vision/docs/apis>
- 「リファレンス」とは
 - APIでできることの全情報が書かれたもの
 - APIの使い方辞典のような扱い
- よく調べること
 - 「入門ガイド」に書かれている内容が自分のやりたいことと少し違うとき
 - 自分のやりたいことを行うプログラムのベストな書き方を探すとき

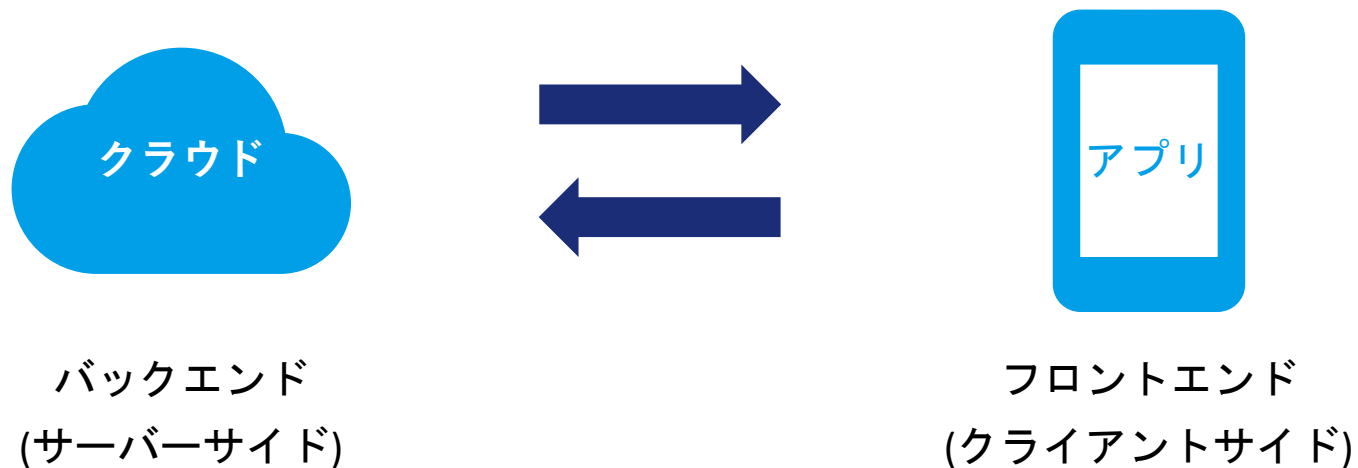
クイズ

Q

実際にAPIを使用するプログラムを書いている。APIの機能で自分の使い方にピッタリ合わないところがあった。どこを見て修正すると良いか？

フロントエンド・バックエンドの開発

- アプリを開発するには
 - ・ ユーザーに見える画面: フロントエンド
 - ・ クラウドでのデータの保存・処理: バックエンド
 - ・ フロントエンド開発はアプリ開発に必須
- フロントエンド開発
 - ・ フロントエンド開発は作るアプリによって型が決まっている
 - ・ フロントエンド開発に必要な知識を理解する
- バックエンド開発
 - ・ バックエンド開発は自由度が高い
 - ・ バックエンド開発に必要な知識を理解する



アプリを開発するには

- **フロントエンド開発は必須**
 - ・ フロントエンド: ユーザーに見える画面のこと ≡ アプリ自体
- **アプリの内容によってバックエンド開発が必要**
 - ・ データの保存(データベース)・複雑で重い処理(AIなど)

フロントエンド開発

- フロントエンド開発は型が決まっている

1. 画面レイアウトを作る
2. アクション(ボタンが押されたなど)に紐づける動作のプログラムを書く
3. アプリを提供できる形式に変換して、配布・公開する

- フロントエンド開発に必要な知識

- レイアウトファイルの書き方: HTML / XML など
- 動作を記述するプログラミング言語の習得: Javascript / Kotlin / C# など
- 開発環境の使い方: VS Code / Android Studio / Xcode / Visual Studio など
- アプリの公開方法: アプリストアの知識・公開方法 など
- + 開発の基礎知識・作りたいアプリ特有の知識

プラットフォームごとに必要な知識

プラットフォーム	レイアウト	言語	開発環境	公開
Web	HTML/CSS	Javascript	各種エディタ	各種Webサーバー
Android	XML (Layout Editor)	Kotlin	Android Studio	Play Store
iOS/Mac	XML (Storyboard)	Swift	Xcode	App Store
Windows	XML (XAMLデザイナー)	C#	Visual Studio	Microsoft Store

GCPのドキュメントにいろいろな言語のサンプルがあるのは
各プラットフォームに組み込むため

バックエンド開発

- アプリ(クライアントサイド/フロントエンド)でできないこと
 - アプリを使うユーザーでデータを共有する・検索する
 - 重い処理を行う(動画の変換・AIを使った分析 など)
 - 機密性のある処理 (自社開発の新技术 など)
- バックエンド開発の知識 → 自由度が高い≒難しい
 - 開発に用いる言語: Javascript(Node.js) / Python / Ruby / Java / C言語 など
 - 公開方法: サーバー / Docker / Kubernetes / 各種クラウドサービス など
 - + 開発の基礎知識・作りたいアプリ特有の知識
- 言語の選択
 - アプリに必要な機能を作りやすい言語を選ぶ (AI = Python など)

開発の基礎知識・作りたいアプリ特有のこと

● 開発の基礎・共通知識

- 通信の仕組み: HTTP / REST / ネットワーク(IP・DNS)
- データの形式: テキスト(JSON / CSV) / バイナリ(画像・動画・音声)
- 開発ツール: Bash (コマンドライン・シェル) / Git

● 作りたいアプリ特有のこと = 「リファレンス」の見方

- フレームワーク/ライブラリ: 多種多様
- クラウドサービス: 多種多様
- データベース: SQL / NoSQL

クイズ

Q

FacebookのようなSNSでユーザーの投稿を検索する機能は、バックエンド / フロントエンドのどちらで行われるか?

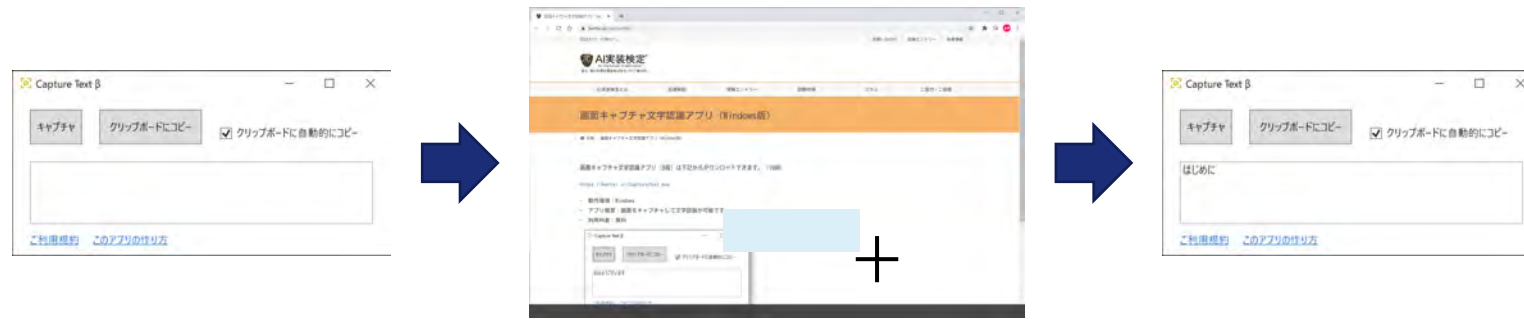
Windowsアプリの開発方法

- **開発方法の選択 … 歴史的経緯から3種類の開発方法がある**

1. UWP: 最新、Windows 10のみで動作、Microsoft Storeからインストール
2. WPF: 一般的なWindowsアプリケーションが作れる、Webサイトで配布
3. Windows フォーム: 古い開発方法 WPFを使う方が良い

- **開発環境の準備と開発の始め方**

- 詳細: <https://docs.microsoft.com/ja-jp/visualstudio/get-started/csharp/tutorial-wpf>



CaptureTextアプリの動作

1. 「キャプチャ」ボタンが押される → 画面選択機能が起動
2. キャプチャ範囲を選択 → キャプチャ範囲の画像取得 → OCR APIに認識を依頼
3. 認識結果がAPIから返ってくる → 貼り付けできる状態にする & 画面に表示

この動作がプログラムでどう表現されているか
実際のコードを見てみよう

クイズ

Q

OCRのAPIはどのタイミングで使われるか?

今日のまとめ

- **GCPのAI APIサービスを使う**

- サービスの探し方 → GCPのプロダクトページ
- 料金の調べ方 → 各サービスのドキュメント内「リソース」
- APIを使いこなす → 「クイックスタート」 → 「リファレンス」

- **フロントエンド開発とバックエンド開発**

- アプリ開発の作業を知る → フロントエンド/バックエンド開発のいろいろ
- 開発に共通な知識の理解 → いくつかの共通知識をしっかりと学ぼう

- **CaptureTextアプリのコードを見てみよう**

- Windowsアプリの開発方法 → 3つの開発方法があること
- 動作の仕組み → アプリの動きを各ステップに分解できるようになる

AIアプリ開発に必要なスキル

企画	構築技術	マネタイズ
課題ドリブンのアイデア	AIのAPI	買い切り
市場調査	学習済みモデルの活用	サブスクリプション
UX設計	深層学習モデルの開発	フリーミアム
サービス設計	UI設計のパターン	アプリ内課金
プロジェクト管理	クライアントサイド開発	広告
	サーバーサイド開発	リリース方法
	DB	配信プラットフォーム
	開発言語/フレームワーク	
	IDE	

宿題

- **考えたアプリの画面と動きを紙芝居で作ろう**
 - 画面にはどのようなボタンが必要?
 - ボタンを押したときの動作は?
 - APIはいつ呼び出す?

音声操作でドローンを飛ばそう



アンケート（ミニテスト付き）のお願い

<https://ws.formzu.net/fgen/S85111623/>