プログラミング概論

第13回　2024年12月18日
App Inventorによる
Androidアプリ開発の実践
(7) WebAPIと連携したアプリの作成

今回の授業内容

- ウェブAPIとは
  - OpenWeatherMapのWebAPI
  - APIキー
  - ウェブブラウザで確認
  - UTCとJST
- 天気予報アプリの作成
- 完成したらレポートを書く

ウェブAPIとは

WebAPIとは

- WebAPIとは
  - Web上の他のサーバのサービスを利用できる（自分のアプリに組み込める）仕組み
    - Google API
      - 地図，カレンダー，Youtube，翻訳など
    - LINE Messaging API
      - チャットボットなど
    - 楽天API
      - 商品検索，ランキングなど
    - YahooAPI
      - ニュース，テキスト解析，求人，ショッピング，地図など
    - OpenWeatherMap
      - 世界中の天気情報

OpenWeatherMapのWebAPI

- OpenWeatherMapは世界中の天気情報を取得できる海外のオンラインサービス
- 指定した場所の天気や気温，湿度，気圧，風速などを取得できる
- 無料版と有料版があり，無料版でも「現在の天気」と「5日間（3時間ごと）の天気」の取得が可能
- OpenWeatherMapを利用するためには
  ① アカウントの作成
  ② キーの作成
  が必要

自分のAPIキーを確認する

openweathermap.org/weather-dashboard

OpenWeather | Weather in your city | Guide | API | Dashboard | Marketplace | Pricing | Maps | Our Initiatives | Partners | Blog | For Business | akiyo | Support

- My services
- My API keys
- My payments
- My profile
- Logout

Weather dashboard

Home / Weather dashbo

OpenWeather
Dashboard

A visual tool for working with weather data and timely tracking of dangerous phenomena

Try the Dashboard

The OpenWeather Dashboard is a lightweight and flexible visual tool for our customers who would like to be notified weather events to make informed decisions and plan actions based on the weather input.

We use cookies which are essential for the site to work. We also use non-essential cookies to help us improve our services. Any data collected is anonymised. You can allow all cookies or manage them individually.

Allow all | Manage cookies

Track trigger events in the dashboard or receive

Make timely decisions once trigger

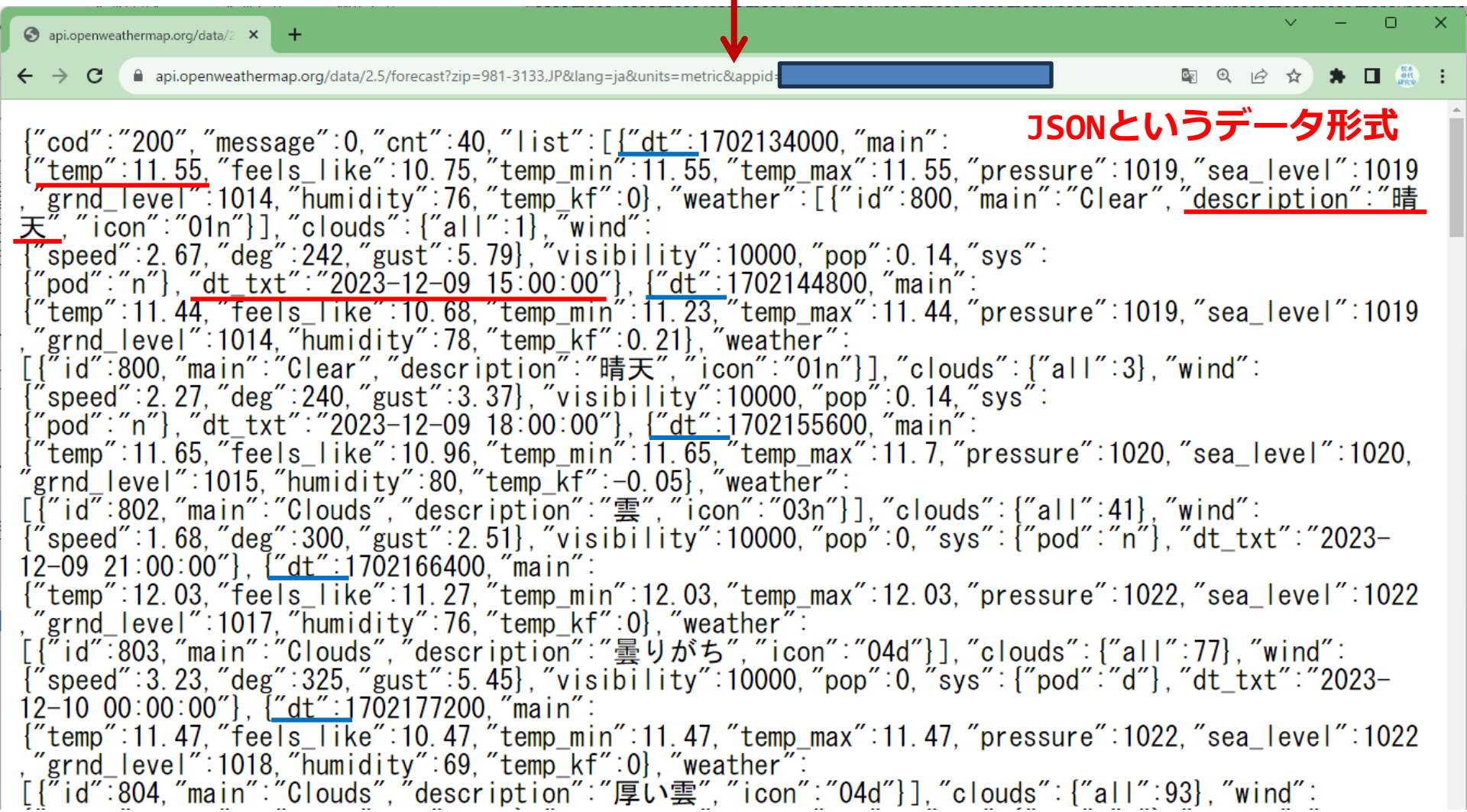
https://home.openweathermap.org/api_keys

自分のAPIキーを確認する

ここの文字列をコピーして利用する

ウェブブラウザでAPIの出力を確認

https://api.openweathermap.org/data/2.5/forecast?zip=981-3133,JP&lang=ja&units=metric&appid=自分のAPIキー



3時間おきに5日分のデータ＝24÷3×5＝40

UTC（協定世界時）とJST（日本標準時）

- dt_txtの日時はUTCなので，日本ではこれより9時間進んだ時刻になる

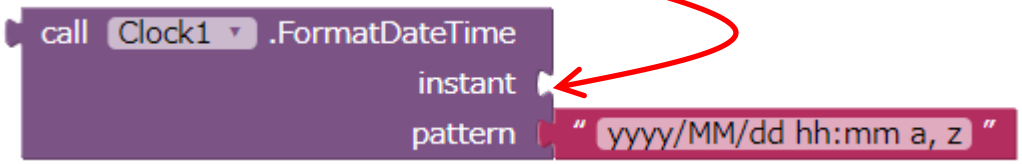
```
UTC                        JST
2023-12-12 15:00:00 -> 2023-12-13 00:00:00
2023-12-13 00:00:00 -> 2023-12-13 09:00:00
2023-12-13 15:00:00 -> 2023-12-14 00:00:00
```

- 実はdtの数字は1970年1月1日0時0分0秒からの経過秒数なのでこれを利用してJSTを求める

```
{"dt":1702447200,・・・,"dt_txt":"2023-12-13 06:00:00"},
```

1000倍して単位をミリ秒に直す

1702447200000

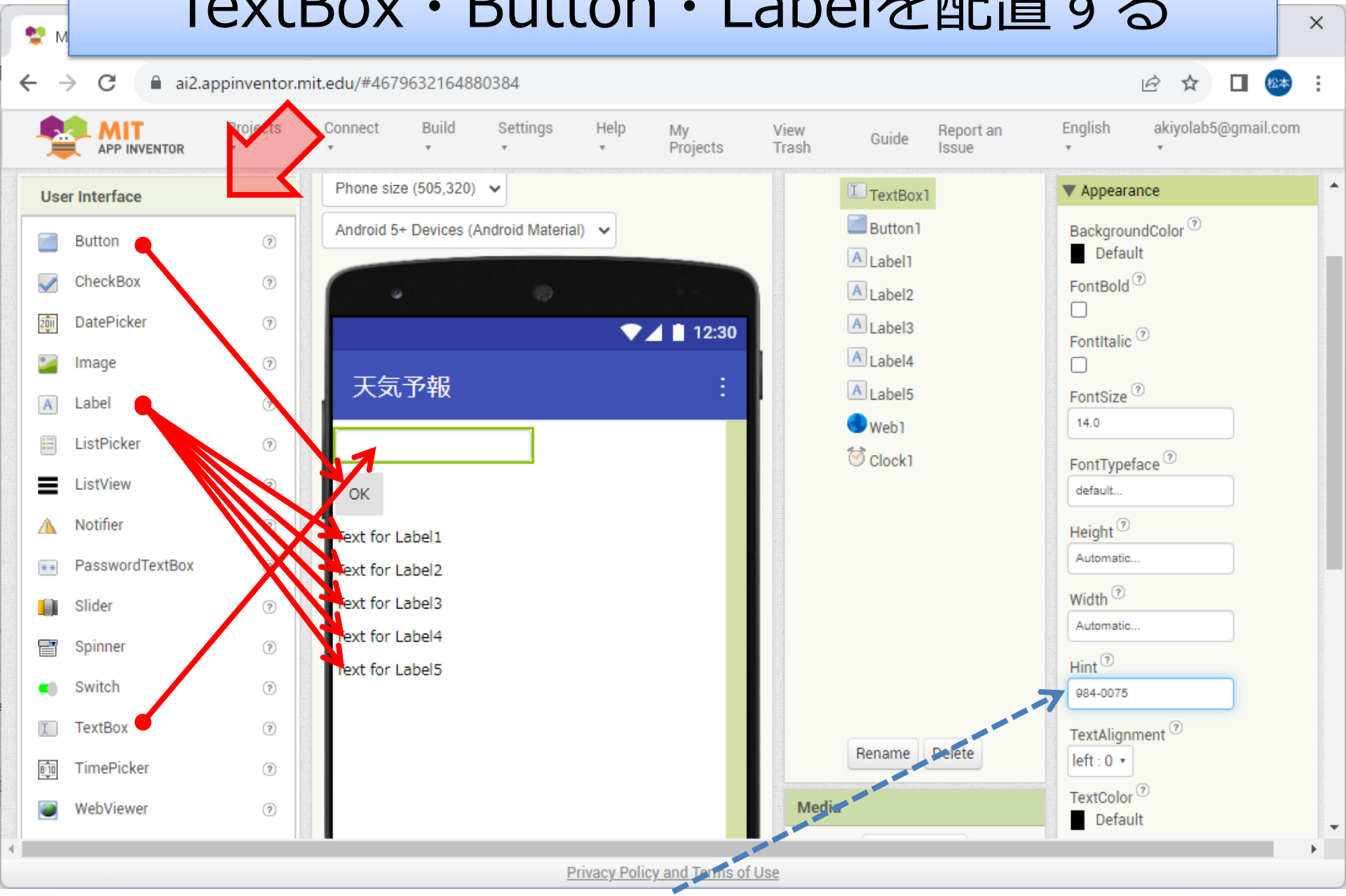


2023/12/13 03:00 午後, JST

その値をこのブロックに与えるとJSTに変換してくれる

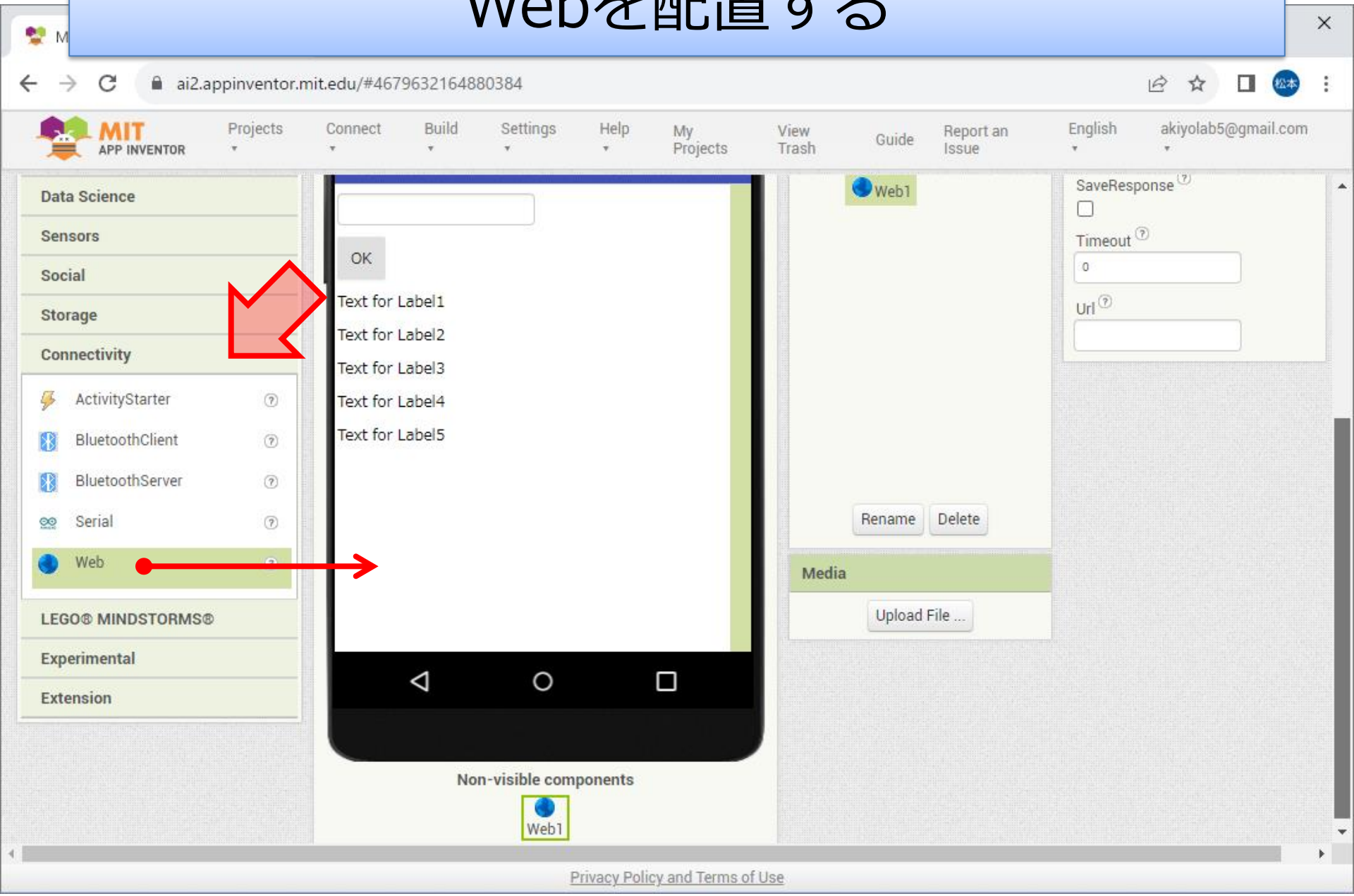
天気予報アプリの作成

TextBox・Button・Labelを配置する

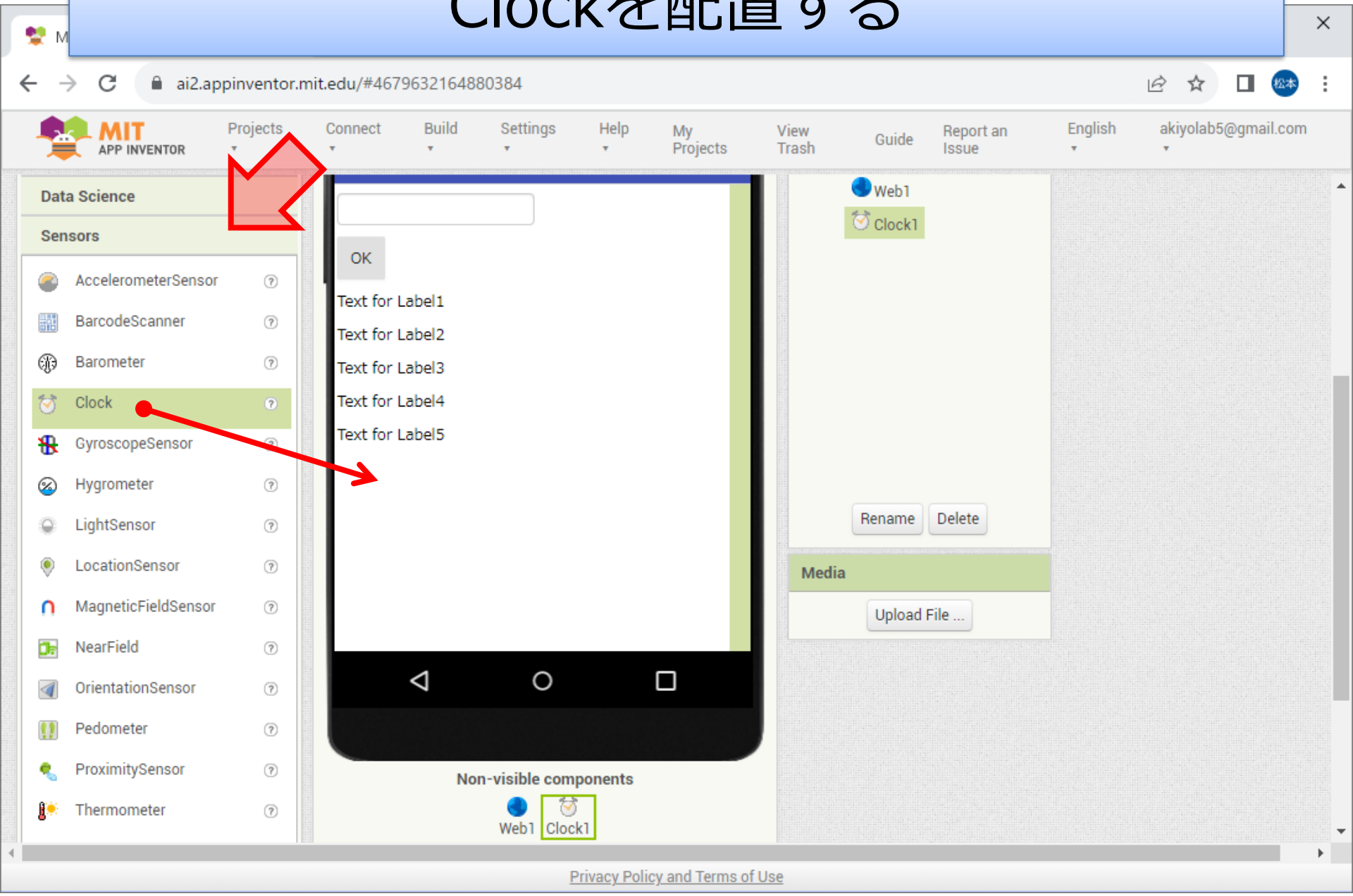


TextBoxのHintにあらかじめ郵便番号を設定しておくと良い

Webを配置する

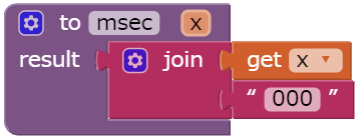

ai2.appinventor.mit.edu/#4679632164880384
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
Data Science
Sensors
Social
Storage
Connectivity
ActivityStarter
BluetoothClient
BluetoothServer
Serial
Web
LEGO® MINDSTORMS®
Experimental
Extension
OK
Text for Label1
Text for Label2
Text for Label3
Text for Label4
Text for Label5
Non-visible components
Web1
Web1
Rename
Delete
Media
Upload File ...
SaveResponse
Timeout
0
Url
Privacy Policy and Terms of Use

Clockを配置する


ai2.appinventor.mit.edu/#4679632164880384
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
akiyolab5@gmail.com
Data Science
Sensors
AccelerometerSensor
BarcodeScanner
Barometer
Clock
GyroscopeSensor
Hygrometer
LightSensor
LocationSensor
MagneticFieldSensor
NearField
OrientationSensor
Pedometer
ProximitySensor
Thermometer
OK
Text for Label1
Text for Label2
Text for Label3
Text for Label4
Text for Label5
Non-visible components
Web1
Clock1
Web1
Clock1
Rename
Delete
Media
Upload File ...
Privacy Policy and Terms of Use

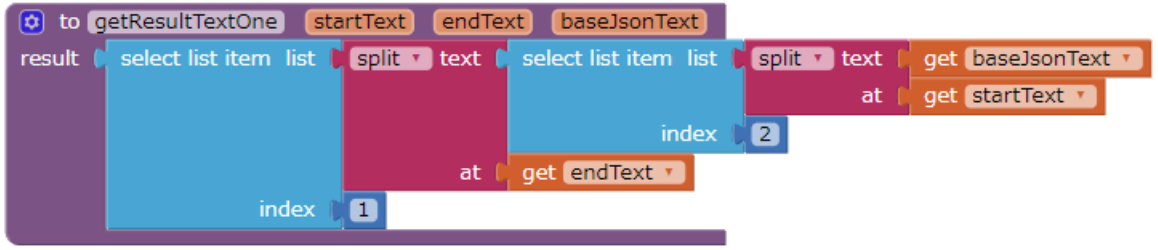
〒981-3133の日時と天気を4つ表示する

プロシージャを作成する.
プロシージャとは
複数の処理を意味を持った
ひとまとまりにしたもの.
意味のある処理をひとつに
まとめることにより,
プログラムがわかりやすく
なる.

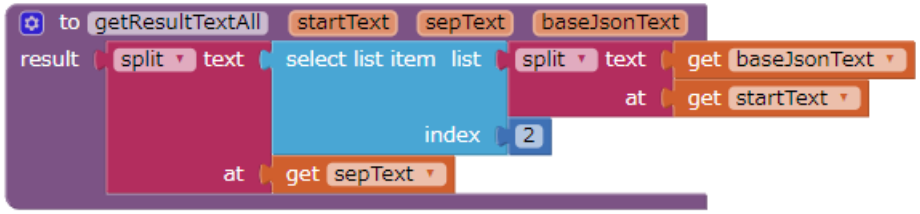


ミリ秒に変換する処理

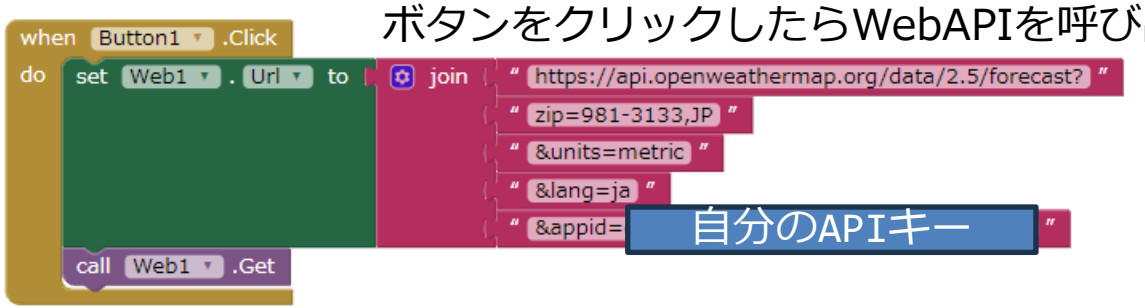
1つのデータから必要な部分を抽出する処理

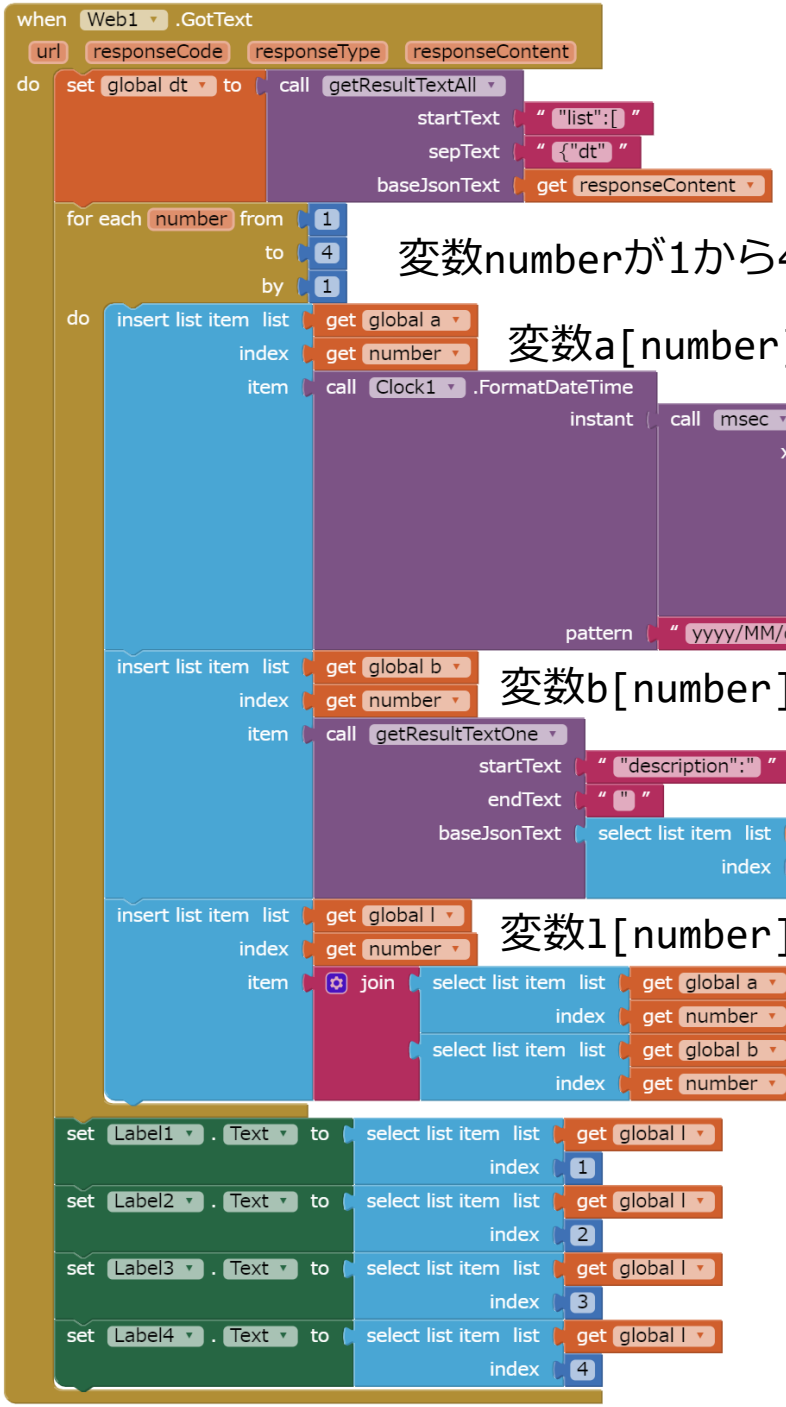


全体から1つ1つのデータに切り分ける処理



ボタンをクリックしたらWebAPIを呼び出す





データを「{"dt"」で分割し変数dtに入れる
（次ページで詳しく解説）

変数numberが1から4になるまで4回繰り返す

変数a[number]に日時のデータを入れる

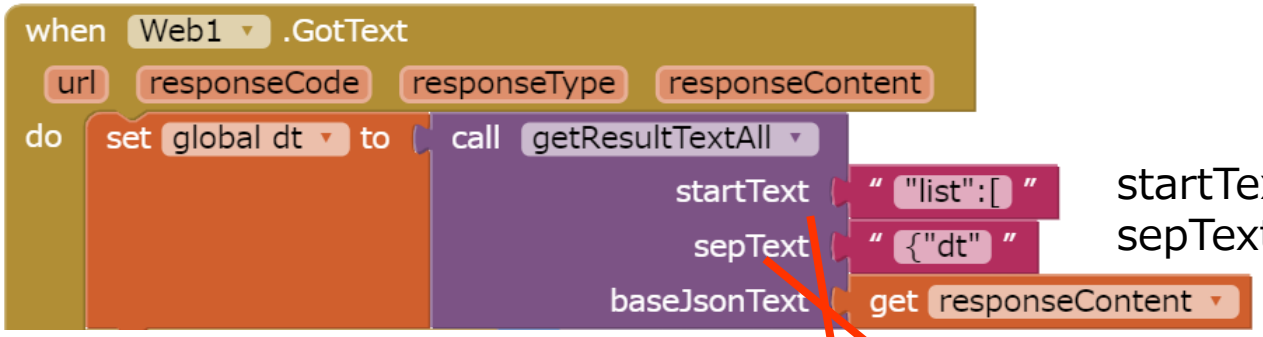
「:」から「,」までを抽出

変数b[number]に天気のデータを入れる

「"description":"」から「"」までを抽出

変数l[number]にa[number]とb[number]をくっつけて入れる

APIから得られたデータの処理方法



startTextは始まりの文字列
sepTextは区切りの文字列

WebAPIから返ってきた
データはresponseContent
に格納されている

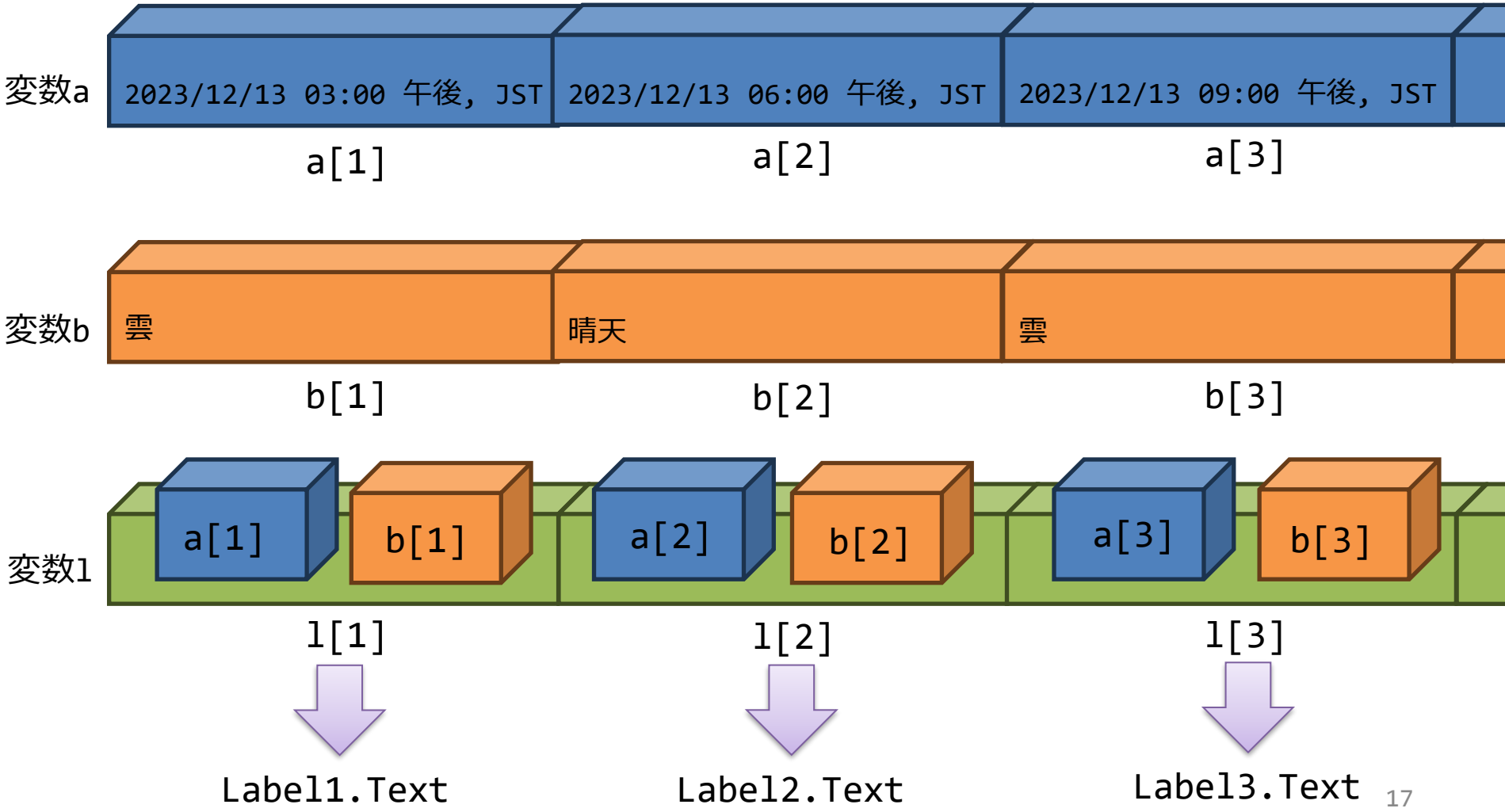
{“cod”:“200”,“message”:0,“cnt”:40,“list”:[{“dt”:1701961200,“main”:・・・

不要
（使用しない）

１つ目のデータはdt[2]に
２つ目のデータはdt[3]に
・・・と格納される

“list”:[より後ろで
{“dt”より前は
実質的には存在しないが
理論上dt[1]に格納される

```
変数dt[2] :1702447200,・・・,“description”:“雲”,・・・},
    dt[3] :1702458000,・・・,“description”:“晴天”,・・・},
    dt[4] :1702468800,・・・,“description”:“雲”,・・・},
                          :
                          :
```

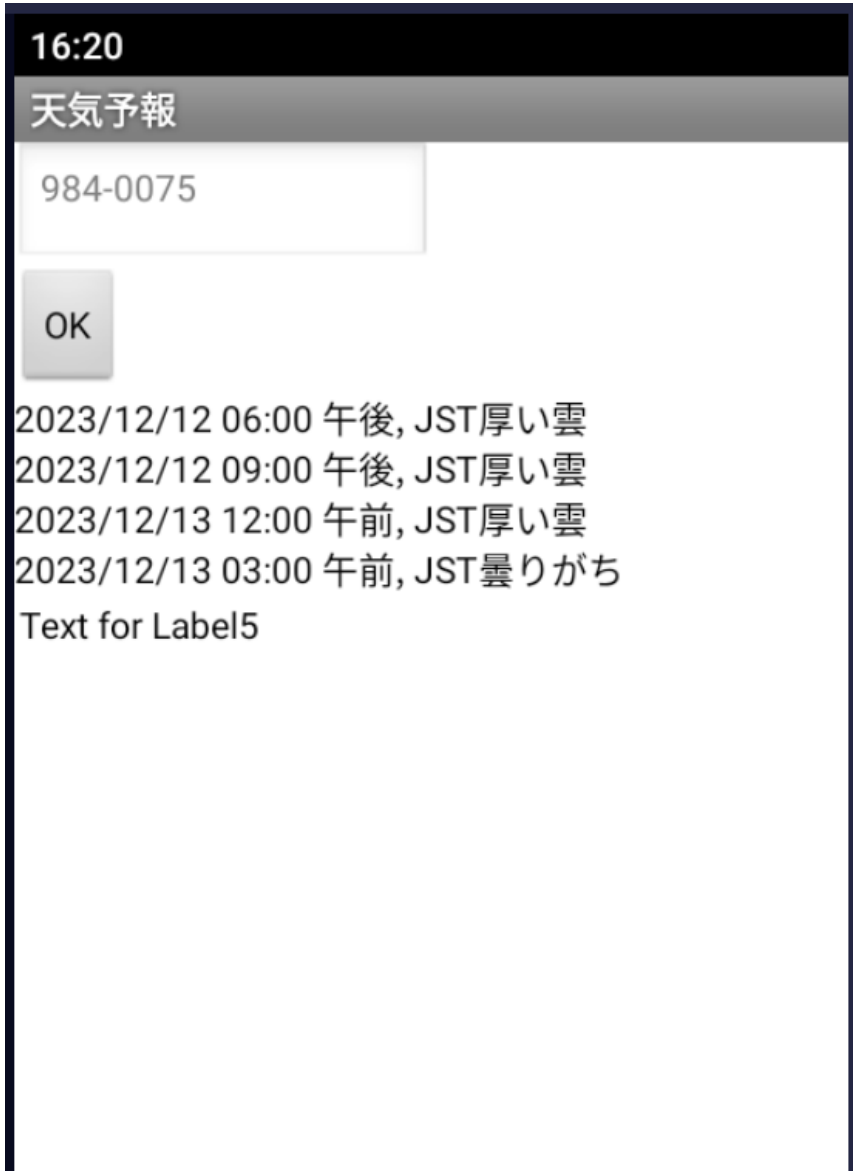


拡張する

- Labe15に次のデータが入るようにする
- 変数cに気温のデータ（Temp）を入れて天気の右に表示されるようにする
- テキストボックスに入力した郵便番号の場所の天気情報が出るようにする
- 日時・天気・気温の間にスペースを入れて見やすくする

実行画面

拡張前


16:20
天気予報
984-0075
OK
2023/12/12 06:00 午後, JST厚い雲
2023/12/12 09:00 午後, JST厚い雲
2023/12/13 12:00 午前, JST厚い雲
2023/12/13 03:00 午前, JST曇りがち
Text for Label5

拡張後

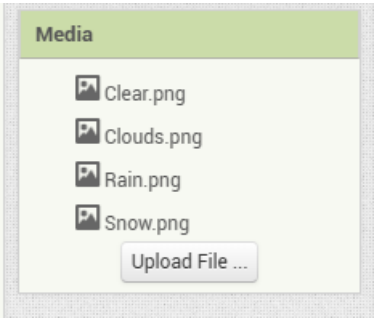
16:48
天気予報
984-0075
OK
2023/12/12 06:00 午後, JST 厚い雲 9.75
2023/12/12 09:00 午後, JST 厚い雲 8.84
2023/12/13 12:00 午前, JST 厚い雲 7.06
2023/12/13 03:00 午前, JST 曇りがち 4.27
2023/12/13 06:00 午前, JST 晴天 3.84

ここまではひととおりやってみよう
できたらあとは自由に工夫する
例：

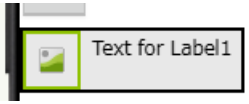
- 6つ以上のデータが出るようにする
- 日時・天気・気温以外の情報も出るようにする
  - APIの仕様を確認する
    - https://openweathermap.org/forecast5
- 自宅と五橋キャンパスの天気がいっぺんに出るようにする
- 3時間おきではなく6時間おきにする
  - ヒント

などなど

どうしても天気画像を出したい人へ向けたヒント



1. 晴・曇・雨・雪の画像を用意しファイル名を Clear・Clouds・Rain・Snowにする【重要】
2. それらをMediaにアップロードする
3. デザイナ画面でまずImage1を作って WidthやHeightを適当に設定する



4. APIから得られるデータの中にmainという項目があり，そこにClear/Clouds/Rain/Snowのいずれかが入っているので，それを抽出する
5. ".png"をくっつけてImage1.Pictureに入れる

完成したらレポートを書く

課題レポートについて

- レポートの書き方
  - Microsoft Wordで作成する
  - レポートの冒頭に，科目名・タイトル・所属・学生番号・氏名・提出年月日を入れる
  - 決められた構成で書く
  - サンプルをよく見てまねて書く
  - 提出方法・期限を守る
  - 困ったことがあったら期限前に担当教員に相談する → 1G2G: akiyo@mail.tohoku-gakuin.ac.jp
    3G4G: hiro_m@mail.tohoku-gakuin.ac.jp

課題レポートについて

- レポートの構成
  1. アプリの概要
     - 特徴・機能など詳細に書く。
  2. デザイナ画面
     - 各コンポーネントの説明も入れる。
  3. ブロックエディタ画面
     - 各変数の説明も入れる。
  4. 実行画面・操作方法
  5. 工夫した点
     - 「なし」はダメ。何か工夫すること。

第3回課題レポートについて

- タイトル
  - 天気予報アプリの制作
- 内容
  - 制作した天気予報アプリについてレポートを作成する
- 提出方法
  - 前回と同じ
- 提出期限
  - 1月14日（火）23:59

レポート作成上の注意

- 「コンポーネントの説明」「変数の説明」「操作説明」を書き漏らさないよう注意すること。
- ブロックエディタ画面のスクリーンショットはすべてのブロックが読めるように貼り付けること。
- 「工夫した点」を必ず挙げること。レイアウトの変更などデザイナ画面でおこなった工夫を含めても良いが，最低でも一つは「プログラム上の工夫（ブロックエディタ上の工夫）」があること。
- 「工夫した点」については，（プレイしなくても読んだだけで）意味が伝わるように特に丁寧に説明を書くこと。
- レポートは感想文ではないので感想を書いてはいけない。
  - 「～できてよかった」「大変だった」「難しかった」など