

プログラミング概論

第13回 2023年12月13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

Weather dashboard

Home / Weather dashboard

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


自分のAPIキーを確認する

home.openweathermap.org/api_keys

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

New Products Services API keys Billing plans Payments Block logs My orders My profile Ask a question

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Status	Actions
[Redacted Key]	Default	Active	 

Create key

API key name

この文字列をコピーして利用する

Product Collections

- Current and Forecast APIs
- Historical Weather Data
- Weather Maps
- Weather Dashboard
- Widgets

Subscription

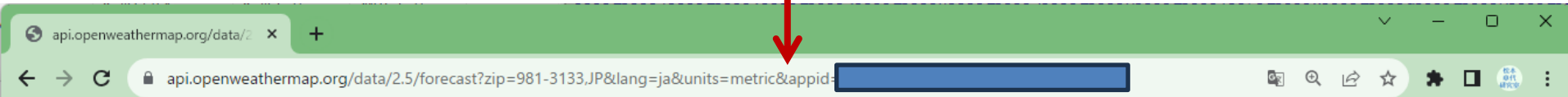
- How to start
- Pricing
- Subscribe for free
- FAQ

Company

OpenWeather is a team of IT experts and data scientists that has been practising deep weather data science since 2014. For each point on the globe, OpenWeather provides historical, current and forecasted weather data via light-speed APIs. Headquarters in London, UK.

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>



JSONというデータ形式

```
{
  "cod": "200",
  "message": 0,
  "cnt": 40,
  "list": [
    {
      "dt": 1702134000,
      "main": {
        "temp": 11.55,
        "feels_like": 10.75,
        "temp_min": 11.55,
        "temp_max": 11.55,
        "pressure": 1019,
        "sea_level": 1019,
        "grnd_level": 1014,
        "humidity": 76,
        "temp_kf": 0
      },
      "weather": [
        {
          "id": 800,
          "main": "Clear",
          "description": "晴天",
          "icon": "01n"
        }
      ],
      "clouds": {
        "all": 1
      },
      "wind": {
        "speed": 2.67,
        "deg": 242,
        "gust": 5.79
      },
      "visibility": 10000,
      "pop": 0.14,
      "sys": {
        "pod": "n"
      },
      "dt_txt": "2023-12-09 15:00:00"
    },
    {
      "dt": 1702144800,
      "main": {
        "temp": 11.44,
        "feels_like": 10.68,
        "temp_min": 11.23,
        "temp_max": 11.44,
        "pressure": 1019,
        "sea_level": 1019,
        "grnd_level": 1014,
        "humidity": 78,
        "temp_kf": 0.21
      },
      "weather": [
        {
          "id": 800,
          "main": "Clear",
          "description": "晴天",
          "icon": "01n"
        }
      ],
      "clouds": {
        "all": 3
      },
      "wind": {
        "speed": 2.27,
        "deg": 240,
        "gust": 3.37
      },
      "visibility": 10000,
      "pop": 0.14,
      "sys": {
        "pod": "n"
      },
      "dt_txt": "2023-12-09 18:00:00"
    },
    {
      "dt": 1702155600,
      "main": {
        "temp": 11.65,
        "feels_like": 10.96,
        "temp_min": 11.65,
        "temp_max": 11.7,
        "pressure": 1020,
        "sea_level": 1020,
        "grnd_level": 1015,
        "humidity": 80,
        "temp_kf": -0.05
      },
      "weather": [
        {
          "id": 802,
          "main": "Clouds",
          "description": "雲",
          "icon": "03n"
        }
      ],
      "clouds": {
        "all": 41
      },
      "wind": {
        "speed": 1.68,
        "deg": 300,
        "gust": 2.51
      },
      "visibility": 10000,
      "pop": 0,
      "sys": {
        "pod": "n"
      },
      "dt_txt": "2023-12-09 21:00:00"
    },
    {
      "dt": 1702166400,
      "main": {
        "temp": 12.03,
        "feels_like": 11.27,
        "temp_min": 12.03,
        "temp_max": 12.03,
        "pressure": 1022,
        "sea_level": 1022,
        "grnd_level": 1017,
        "humidity": 76,
        "temp_kf": 0
      },
      "weather": [
        {
          "id": 803,
          "main": "Clouds",
          "description": "曇りがち",
          "icon": "04d"
        }
      ],
      "clouds": {
        "all": 77
      },
      "wind": {
        "speed": 3.23,
        "deg": 325,
        "gust": 5.45
      },
      "visibility": 10000,
      "pop": 0,
      "sys": {
        "pod": "d"
      },
      "dt_txt": "2023-12-10 00:00:00"
    },
    {
      "dt": 1702177200,
      "main": {
        "temp": 11.47,
        "feels_like": 10.47,
        "temp_min": 11.47,
        "temp_max": 11.47,
        "pressure": 1022,
        "sea_level": 1022,
        "grnd_level": 1018,
        "humidity": 69,
        "temp_kf": 0
      },
      "weather": [
        {
          "id": 804,
          "main": "Clouds",
          "description": "厚い雲",
          "icon": "04d"
        }
      ],
      "clouds": {
        "all": 93
      },
      "wind": {

```

3時間おきに5日分のデータ = $24 \div 3 \times 5 = 40$

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

```
call Clock1 .FormatDateTime
instant
pattern "yyyy/MM/dd hh:mm a, z" 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

The screenshot shows the MIT App Inventor web interface. At the top, there's a navigation bar with 'Projects', 'Connect', 'Build', 'Settings', 'Help', 'My Projects', 'View Trash', 'Guide', 'Report an Issue', 'English', and 'akiyolab5@gmail.com'. Below this is the 'User Interface' panel on the left, listing various widgets like Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, and WebViewer. A red arrow points from the 'Projects' tab to the 'User Interface' panel. Another red arrow points from the 'TextBox' widget to a text box on the phone screen. Several other red arrows point from the 'Label' widget to labels on the phone screen. The phone screen displays a weather forecast app with a blue header '天気予報' and a text box with an 'OK' button. On the right, the 'Appearance' panel for the selected text box is visible, showing properties like BackgroundColor, FontBold, FontItalic, FontSize (14.0), FontTypeface (default...), Height (Automatic...), Width (Automatic...), Hint (984-0075), TextAlignment (left: 0), and TextColor (Default). A blue dashed arrow points from the 'Hint' field to the text box on the phone screen. At the bottom, there's a link for 'Privacy Policy and Terms of Use'.

TextBoxのHintにあらかじめ郵便番号を設定しておくが良い

Webを配置する

The screenshot shows the MIT App Inventor web interface. At the top, a blue banner contains the title "Webを配置する". Below it is a browser address bar with the URL "ai2.appinventor.mit.edu/#4679632164880384". The main interface is divided into several sections:

- Navigation Bar:** Includes "Projects", "Connect", "Build", "Settings", "Help", "My Projects", "View Trash", "Guide", "Report an Issue", "English", and the user email "akiyolab5@gmail.com".
- Component Palette (Left):** Lists various categories: Data Science, Sensors, Social, Storage, Connectivity, LEGO® MINDSTORMS®, Experimental, and Extension. Under "Connectivity", the "Web" component is highlighted with a red dot and a red arrow pointing to the design canvas.
- Design Canvas (Center):** Displays a mobile app design with a white screen and a black Android navigation bar. The screen contains a text input field, an "OK" button, and five text labels ("Text for Label1" through "Text for Label5"). A "Web1" component icon is visible in the top right corner of the design area.
- Properties Panel (Right):** Shows the configuration options for the selected "Web1" component, including "SaveResponse" (checkbox), "Timeout" (input field with "0"), and "Url" (input field). Below this are "Rename" and "Delete" buttons.
- Media Panel (Bottom Right):** Contains an "Upload File ..." button.
- Non-visible components (Bottom Center):** A section labeled "Non-visible components" containing a "Web1" component icon.

At the bottom of the page, there is a link for "Privacy Policy and Terms of Use".

Clockを配置する

The screenshot shows the MIT App Inventor web interface. At the top, a blue banner contains the title "Clockを配置する". Below it, the browser address bar shows the URL "ai2.appinventor.mit.edu/#4679632164880384". The main interface includes a top navigation bar with options like "Projects", "Connect", "Build", "Settings", "Help", "My Projects", "View Trash", "Guide", "Report an Issue", "English", and the user email "akiyolab5@gmail.com".

On the left side, there is a "Sensors" panel with a list of components. The "Clock" component is highlighted in green, and a red arrow points from it to the central design canvas. The design canvas shows a mobile phone screen with a white background and a black bottom navigation bar. A red arrow also points from the "Projects" dropdown menu to the canvas. The canvas contains a white rectangular component with a text input field and an "OK" button. Below the input field, there are five text labels: "Text for Label1", "Text for Label2", "Text for Label3", "Text for Label4", and "Text for Label5".

On the right side, there is a "Web1" component and a "Clock1" component. The "Clock1" component is highlighted in green. Below it are "Rename" and "Delete" buttons. Below the "Web1" component is a "Media" section with an "Upload File ..." button.

At the bottom of the design canvas, there is a "Non-visible components" section with icons for "Web1" and "Clock1". The "Clock1" icon is highlighted with a green box.

At the bottom of the page, there is a link for "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。

initialize global a to create empty list

initialize global b to create empty list

initialize global c to create empty list

initialize global dt to create empty list

initialize global l to create empty list

```
to msec x
result join get x "000"
```

ミリ秒に変換する処理

```
to getResultTextOne startText endText baseJsonText
result select list item list split text select list item list split text get baseJsonText
index 1 at get endText index 2 at get startText
```

1つのデータから必要な部分を抽出する処理

```
to getResultTextAll startText sepText baseJsonText
result split text select list item list split text get baseJsonText
index 2 at get startText at get sepText
```

全体から1つ1つのデータに切り分ける処理

```
when Button1 .Click
do set Web1 .Url to join " https://api.openweathermap.org/data/2.5/forecast? "
" zip=981-3133,JP "
" &units=metric "
" &lang=ja "
" &appid= 自分のAPIキー "
```

ボタンをクリックしたらWebAPIを呼び出す

```
when Web1 .GotText
url responseCode responseType responseContent
do
set global dt to call getResultTextAll
startText "list:[ "
sepText "{dt}"
baseJsonText get responseContent
```

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)

```
for each number from 1
to 4
by 1
```

変数numberが1から4になるまで4回繰り返す

```
do
insert list item list
index
item
```

変数a[number]に日時のデータを入れる

```
call Clock1 .FormatDateTime
instant call msec
x call getResultTextOne
startText ":"
endText ","
baseJsonText select list item list
index get global dt
+ get number + 1
pattern "yyyy/MM/dd hh:mm a, z"
```

「:」から「,」までを抽出

```
insert list item list
index
item
```

変数b[number]に天気の詳細データを入れる

```
call getResultTextOne
startText "description:"
endText ""
baseJsonText select list item list
index get global dt
+ get number + 1
```

「"description:"」から「"」までを抽出

```
insert list item list
index
item
```

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

```
join
select list item list
index get global a
+ get number
select list item list
index get global b
+ get number
```

```
set Label1 .Text to select list item list
index 1
set Label2 .Text to select list item list
index 2
set Label3 .Text to select list item list
index 3
set Label4 .Text to select list item list
index 4
```

APIから得られたデータの処理方法

```
when Web1 .GotText
  url responseCode responseType responseContent
do
  set global dt to call getResultTextAll
    startText "list:[ "
    sepText {"dt" "
    baseJsonText get responseContent
```

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

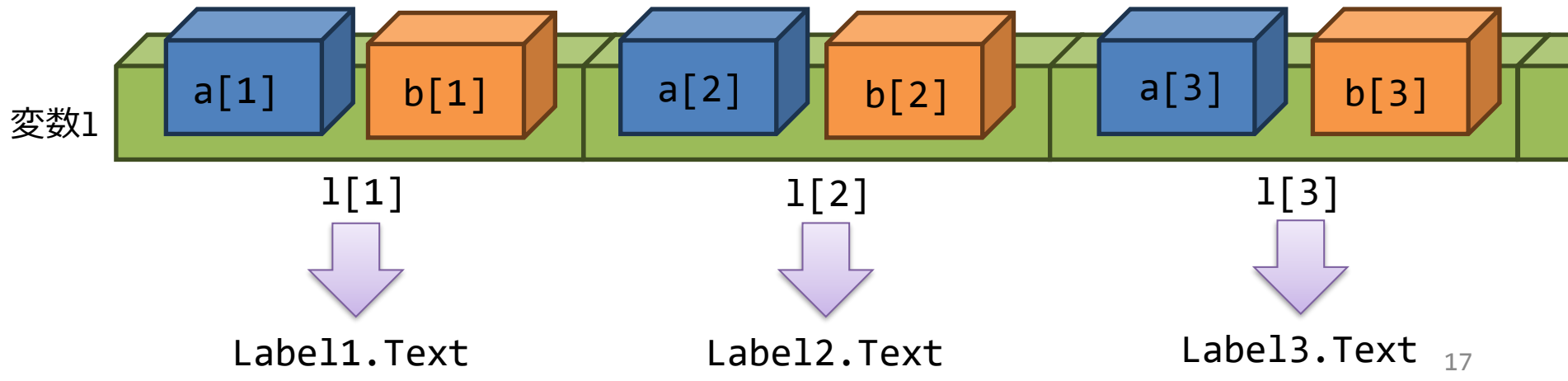
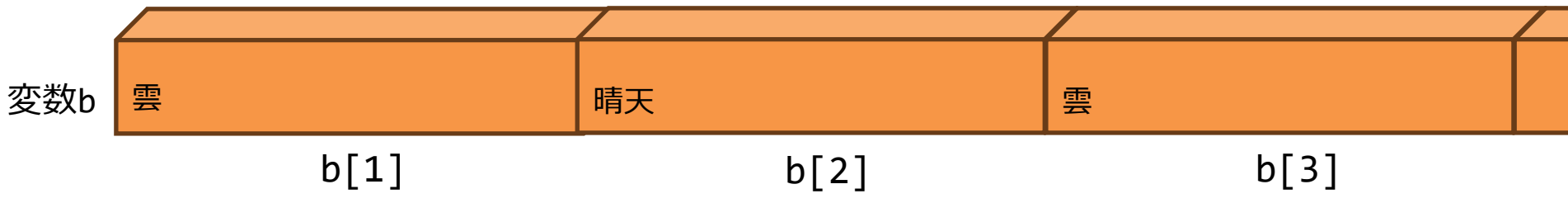
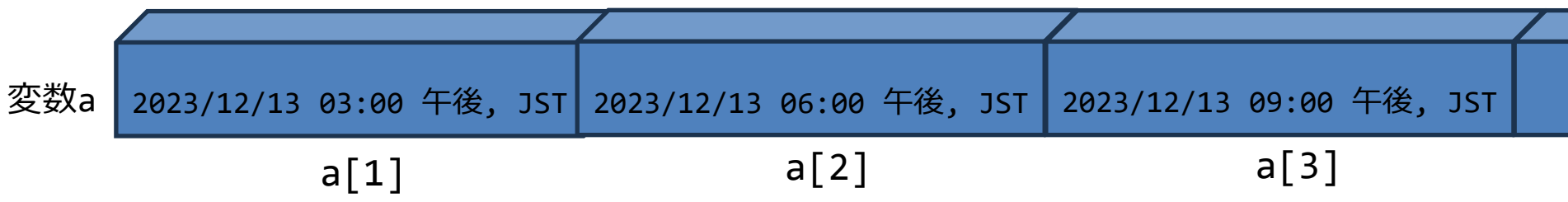
```
{"cod":"200","message":0,"cnt":40,"list":[{"dt":1701961200,"main":...
```

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

- Label15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
天気の上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

拡張後

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

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

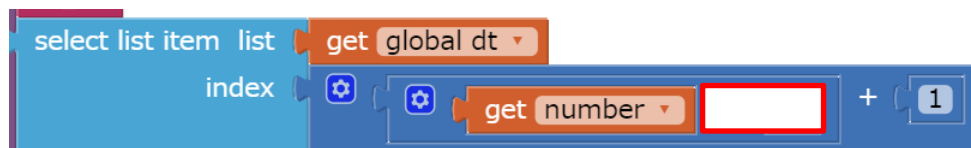


ここまでではひととおりやってみよう
できたらあとは自由に工夫する

例：

- 6つ以上のデータが出るようにする
- 天気／気温以外の情報も出るようにする
- 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

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9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ー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