



1. 講師紹介
2. ロードマップ
3. android概要
4. 文字と画像の表示

講師紹介

1.講師紹介

竹野 学

東京都出身 35歳

- ・大学卒業後はフリーターとなり、手当たり次第にWEBのデザイン、プログラムを学ぶ。
- ・インターネット関連企業のベンチャー企業に就職し、エンジニアとして多数の開発に携わる。(WEBサービス、Androidアプリなど)
- ・フリーランスとして、WEBアプリ・スマホアプリの 設計、開発、プロジェクト管理、運用などに携わる。

<主な業務内容>

企業HPの他、通販、SNS、検索、動画配信、オンライン保険
デジタルチューナーなど

<得意な言語>

- ・ php
- ・ java
- ・ javascript
- ・ HTML5
- ・ 今や亡きFlash ActionScript

<最近取り組んでいることは>

- ・ Swift

<趣味>

音楽、旅

ロードマップ

2.ロードマップ

1回目

- ・ Android Studioの設定
- ・ 文字と画像を表示する(Layout・プログラム両方から)
- ・ 関数を作成し、関数で処理した値をTextViewに表示する。
(四捨五入値の表示する関数、10の倍数値の表示する関数、平均値の表示する関数、敬称「様」をつける関数を作成し、その値をTextViewに表示する。)
- ・ 本日の日時（東京の時間、ニューヨークの時間など）を出力する関数を作成し、その日時をTexViewに表示する。
- ・ より複雑な画面レイアウトを組む。

2回目

- ・ Intentにより、画面遷移を実装。
- ・ クラス、オブジェクトについて。（クラスの作成とクラスのオブジェクト化）。
- ・ クラスの継承。
- ・ 前回作成した数値処理関数と日時処理関数をクラス化し、データをTextViewに表示する。

3回目

- ・ listviewを使いリストを表示する。
- ・ インターフェイス、イベントリスナーの説明

4回目

- ・ RecyclerViewを使い、リスト表示をカスタマイズする。

5回目

- ・ volleyを使って、画像ロード、jsonのロードをする。
- ・ ロードしたデータRecyclerViewに表示する。
- ・ リロード機能、Paging機能。
- ・ 各リストの詳細画面へ遷移する。

6回目

- ・ クラウド型バックエンドサービスのMBaaS提供するAPIサービスを利用して、クラウドAPIとの連携するを実装する。

7回目

- ・ その他 =>メンバーからリクエストが多い事をやる。（例)viewpager/tablAYOUTを利用する。

8回目

- ・ その他 =>メンバーからリクエストが多い事をやる。（例) facebookAPIを利用してfacebookと連動させる

android概要

- activityについてプログラムファイル（java）とレイアウトファイル（xml）について
- androidプロジェクトのファイル構成について
- android Studioの使い方の概要

JAVA JDKについて

Java JDK

1.Java標準の関数群(J2SE/J2EE)

2.コンパイラー

3.Java VM

Android

SDK

Java Servlet

SDK

スマホアプリ

Androidのバイナリファイル
.apk

WEBアプリ

サーブレットのバイナリファイル
.class

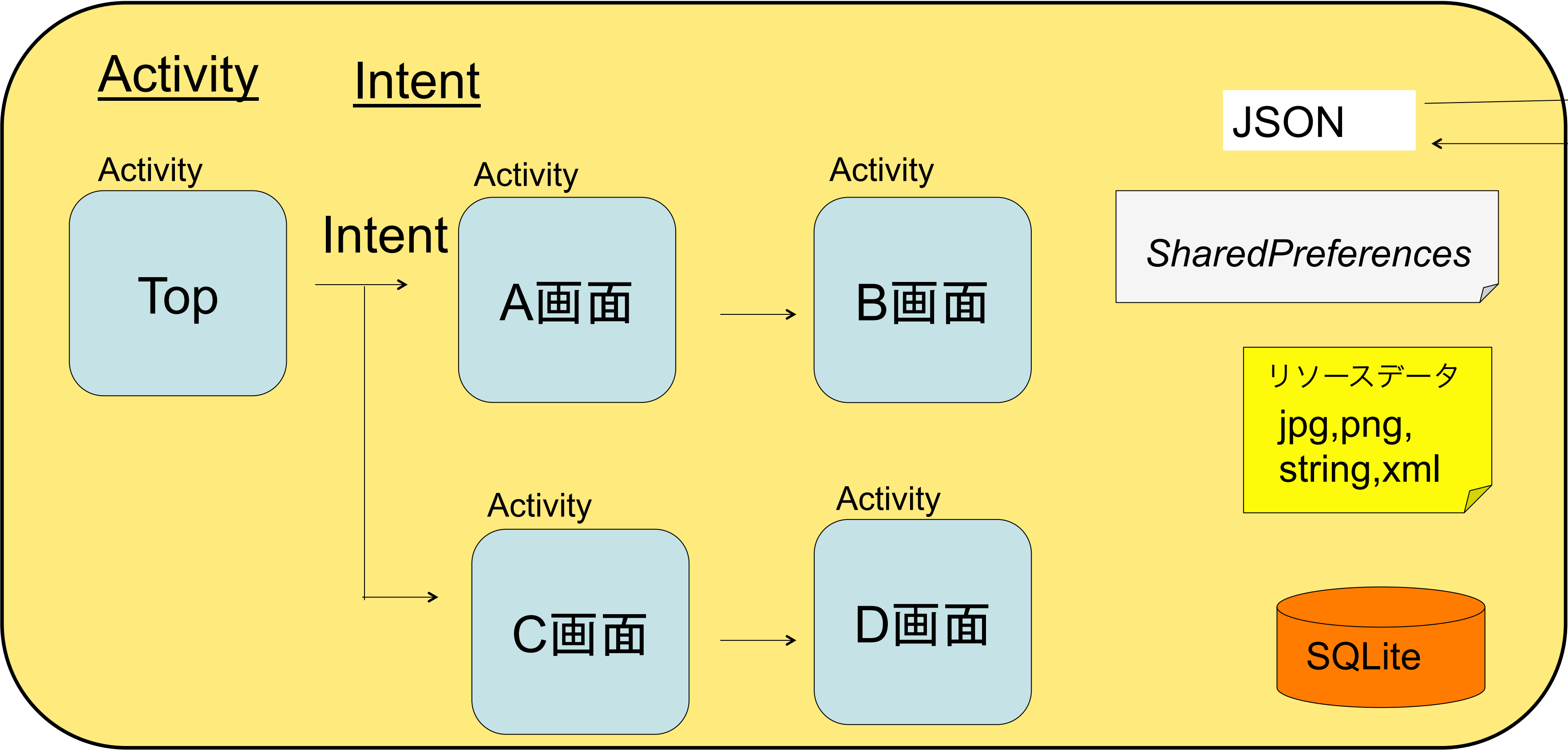
とりあえず

ACTIVITY

INTENT

RESOURCE

Android SDK（ローカル側）

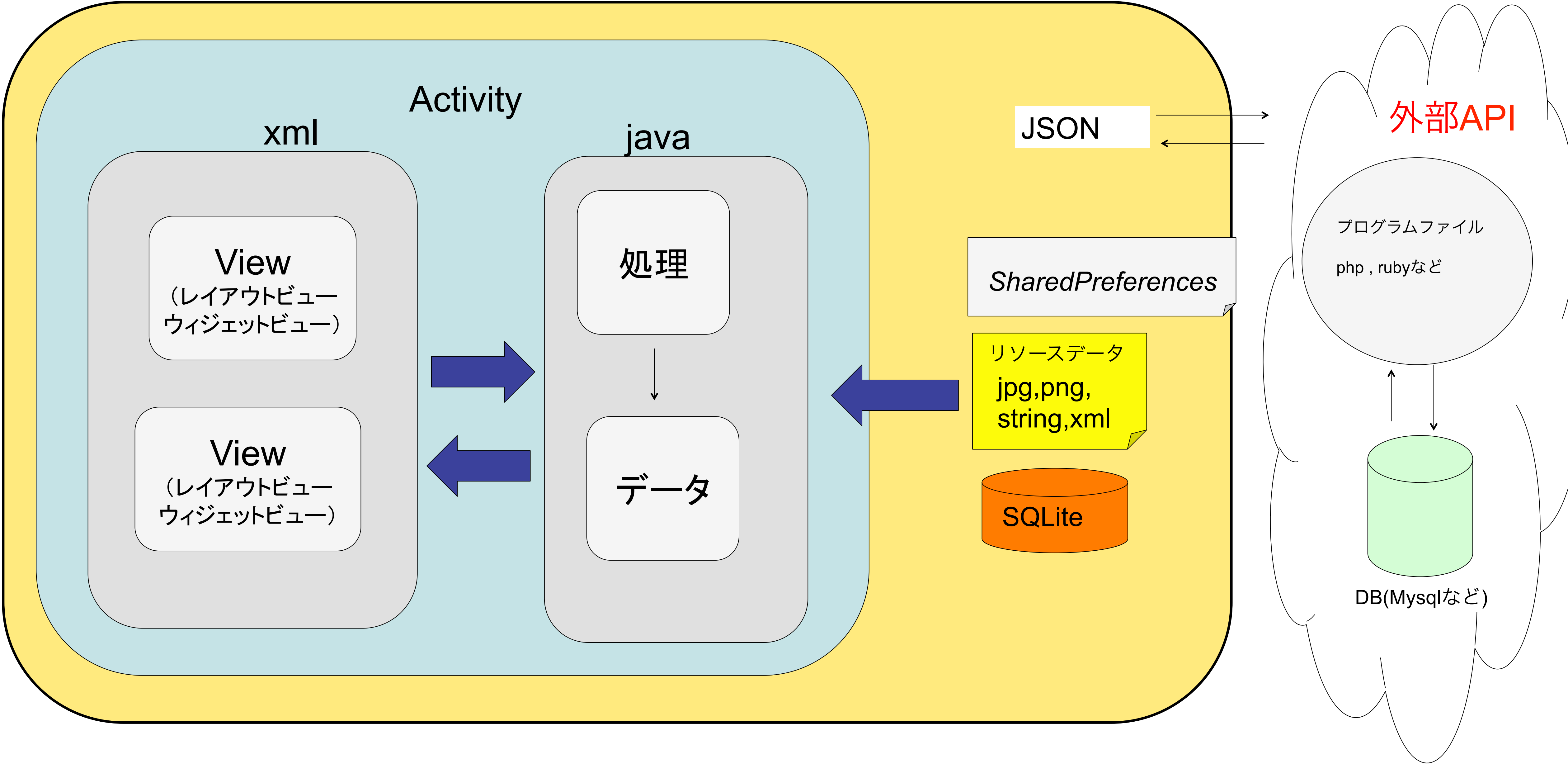


* **SharedPreferences**
アプリの設定データをデバイス内に保存するための仕組み

* **SQLite**
端末側に作られるデータベース

* **リソースデータ**
Android Projectのresフォルダに用意するファイル。予めアプリ内部に保持されるデータ

* **JSON**
データ種類の一つ。



画面を構成するView

コンテナView

RelativeLayoutとLinearLayoutについて
(htmlでいうdivみたいなもの。)

ウィジェットView

TextView,ImageView,Button,ListView

文字と画像の表示

とにかく、文字と画像を画面（Activity）に表示してみよう！
Layoutのテンプレートを操作する。

プログラムからも文字と画像を表示
したり変更したりする。

変数と変数のデータ型について。

変数

変数(**variable**)とは、プログラムのソースコード中で、データを一時的に記憶しておくための領域に固有の名前を付けたもの。データを入れる箱。

データ型

変数には格納するデータの種別を必ず定義しないといけない。
そのデータの種類の事をデータ型と言う。
つまり変数には、定義されたデータ型(データの種別)のデータしか格納出来ない。
まずは、基本の基本、**int**型変数、**String**型変数、配列変数をしっかり理解する。

(基本データ型と参照型(オブジェクト型)があるが、
これはクラスとオブジェクトの仕組みを理解した段階で詳しく説明する。)

int型変数、String型変数

```
//String型 普通の文字列
String name = "Gs Proコース";
//String name = new String("Gs Proコース");とほぼ同じ！
でも、実はメモリ消費的に不効率！
//理由は、クラスとオブジェクト/データのメモリ管理の仕組みの段階で説明する。

//int型 整数
int age = 2;

//変数nameのデータ型は、Stringなので、数値である「2」を格納することは不可。コンパイルエラーが発生する。
name = 2;

//文字列(String型)に変換しているので、エラーは発生しない。
name = 2+"";
name = "2";

//基本的データ型のintのラッパークラスIntegerを利用し,Stringに変換している。
//(ちょっと難しいので、現段階では、分からなくてもOK!クラスとオブジェクトの理解をした段階以降でOK)
name = Integer.toString(age);
```

配列変数

//数値の配列変数について。

```
int[] ageArg = new int[2];  
ageArg [0] = 1;  
ageArg [1] = 2;  
Log.d(“hello_age”,“”+ageArg[1]);//ageArg[2]とするとエラーになる。
```

//テキスト（String）の配列について

```
String[] nameArg = new String[2];  
nameArg[0] = “石井”;  
nameArg[1] = “後藤”;  
Log.d(“hello_name”,“”+nameArg[1]);//nameArg[2]とするとエラーになる。
```

関数とは

```
/**
 * 自己紹介文を作成する
 */
public String myGreeting (String familyName,String firstname,int age) {
    String greeting = "こんにちは、私の名前は"+familyName+" "+firstname+"と申します。";
    greeting += "年齢は"+age+"歳です。";
    return greeting;
}
```

以下の数値処理をする関数を作成し、
その戻り値をそれぞれTextViewに表示する。

- ①四捨五入値の表示する関数
- ②10の倍数値の表示する関数、
- ③平均値の表示する関数、
- ④敬称「様」をつける関数

①四捨五入値の表示する関数

```
int value = (int) Math.round(number);
```

②10の倍数値の表示する関数、

```
int value = number * 10;
```

③平均値の表示する関数、

```
int total = 0;  
for ( int i = 0;i<intArg.length;i++) {  
    total += intArg[i];  
}  
int average = total / intArg.length;
```

④敬称「様」をつける関数

```
name += "様";
```

Or

```
name = name + "様";
```


**本日の日時（東京の時間、ニューヨークの時間など）
を出力する関数を作成し、その日時をTextViewに表示する。**

//現在の日時を出力するロジックを作ってみる。

```
String[] weekArg = new String[8];
weekArg[0] = "";
weekArg[1] = "日";
weekArg[2] = "月";
weekArg[3] = "火";
weekArg[4] = "水";
weekArg[5] = "木";
weekArg[6] = "金";
weekArg[7] = "土";

String tokyoTimezone= "Asia/Tokyo";
//String newYorkTimezone= "America/New_York";

DateFormatSymbols dfs = DateFormatSymbols.getInstance();
dfs.setWeekdays(weekArg);
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd(EEEE) HH:mm:ss",dfs);
TimeZone tz = TimeZone.getTimeZone(tokyoTimezone);
sdf.setTimeZone(tz);
String todayWeek = sdf.format(new Date());

//レイアウトファイルで定義したTextViewオブジェクトに現在の日時の文字列を入れる。
TextView dateView = (TextView) findViewById(R.id.date);
dateView.setText(todayWeek);
```

4.文字と画像の表示

より複雑な画面レイアウトを作成する。

