

A faint, light blue world map is visible in the background of the slide, centered behind the text.

INTERNET ACADEMY

Institute of Web Design & Software Services

Spring Boot 1

インターネット・アカデミー

Spring Boot 1 目次

- 概要、環境構築
- 知識確認 (Java)
- 知識確認 (データベース)



概要・環境構築

概要・環境構築

- **Javaフレームワーク「Spring Boot」** とは
- テンプレートエンジン「Thymeleaf」 とは
- インストール (Eclipse)
- Spring Tool Suite
- ビルドツール「Gradle」 とは
- Eclipseの初期設定

Spring Bootとは

Webアプリケーションを開発するためのJavaのフレームワーク。Spring Frameworkをベースに作られている。

J2EEを用いた複雑な開発を避けるために、当初軽量なフレームワークを目的として作られた「Spring Framework」から、MVCパターンを実現する「Spring MVC」、そして開発で必要な設定を不要とする「Spring Boot」というトレンドの変遷がある



Spring Bootとは

Spring Framework

2004年に登場したフレームワーク。当時はDI(依存性注入)を実現する小さなフレームワークとしてスタート。今ではこれを土台として、いろいろなフレームワークが作られているため、「統合フレームワーク」と呼ばれるほど大規模になっている。

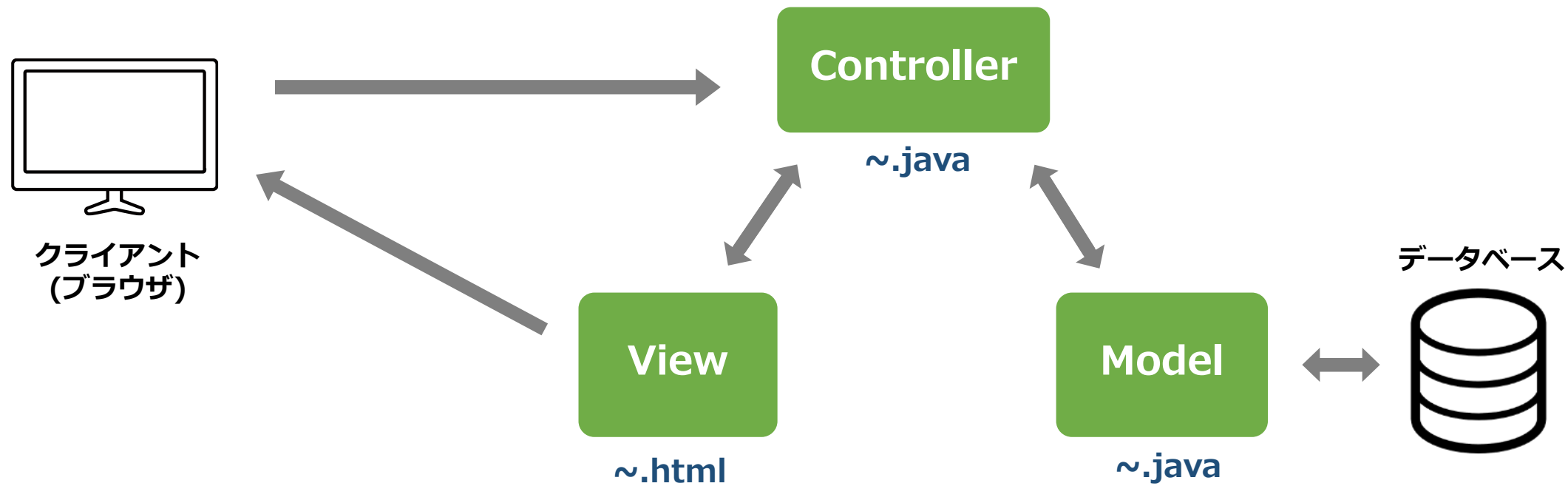
Spring MVC

Webアプリケーションに「MVCアーキテクチャ」による開発を導入するフレームワーク。Webアプリ全体の制御、処理から画面の定義を切り離し、データベースの種類ごとの差を吸収といった、多くの機能を実装した。

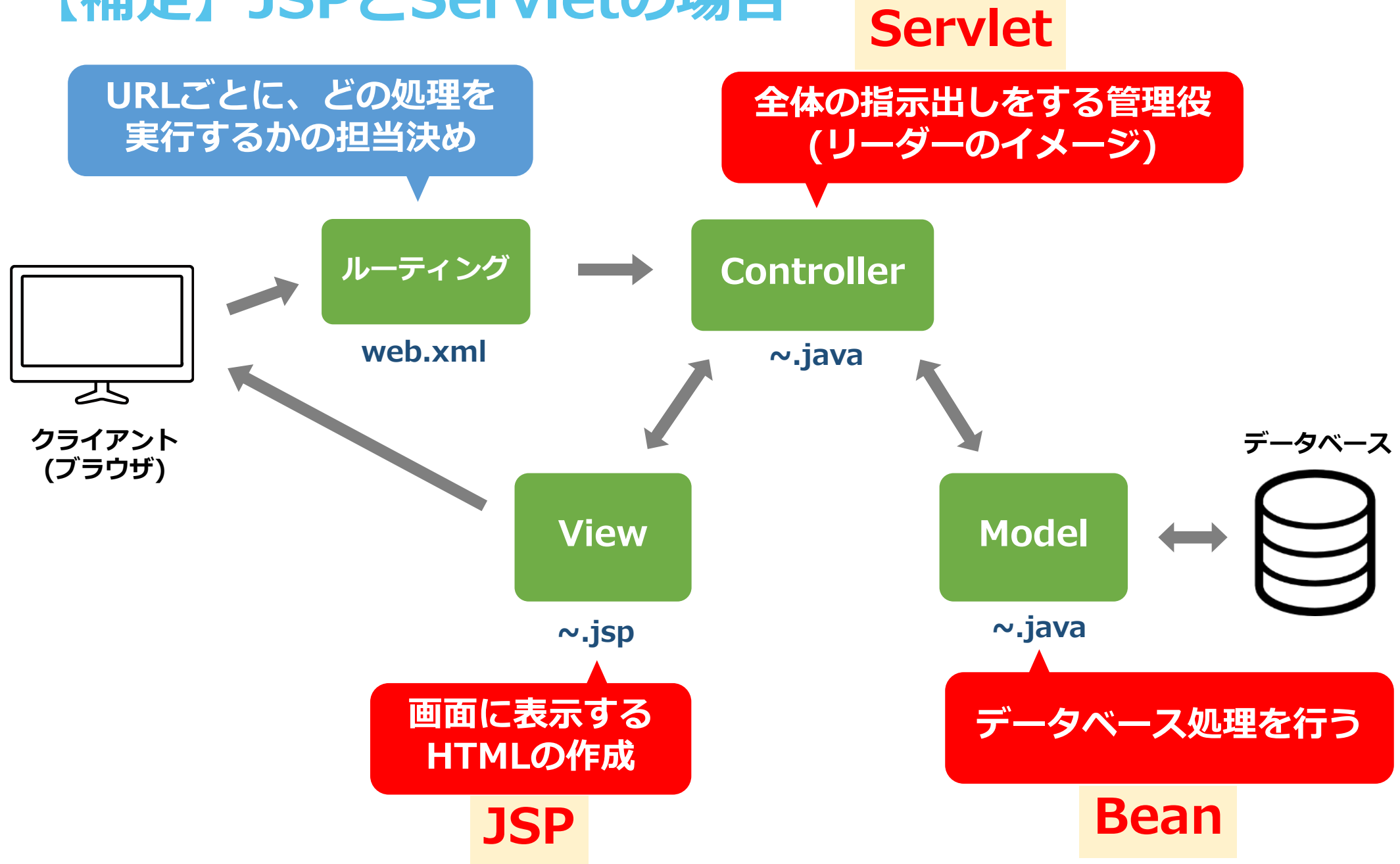
Spring Boot

Spring MVCで必要だった、様々なSpring系のライブラリを導入して環境構築をする手間を自動化してゼロにしたもの。いわばSpring MVCの完成形。

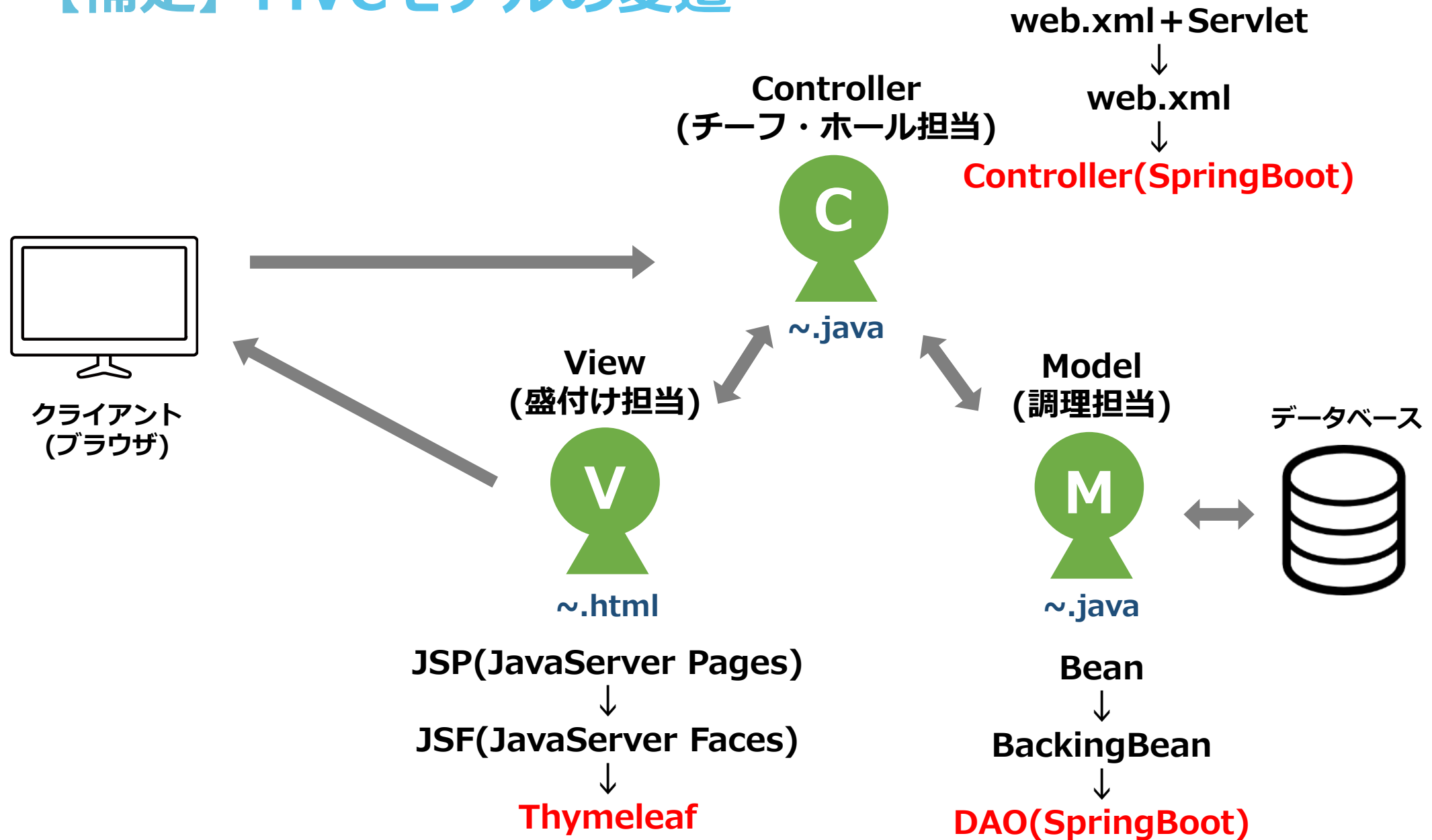
Spring BootにおけるMVCモデル



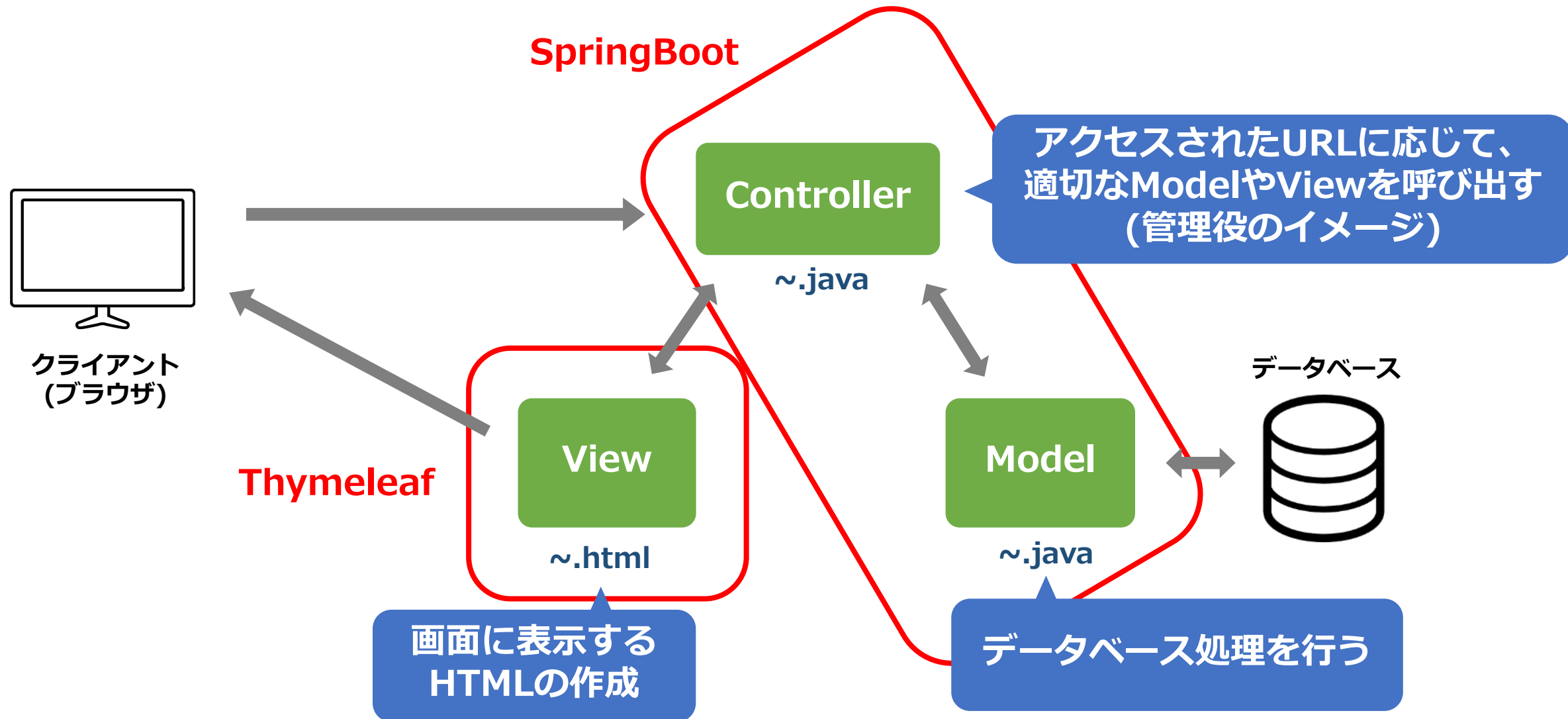
【補足】JSPとServletの場合



【補足】 MVCモデルの変遷



Spring BootにおけるMVCモデル



概要・環境構築

- Javaフレームワーク「Spring Boot」とは
- **テンプレートエンジン「Thymeleaf」とは**
- インストール (Eclipse)
- Spring Tool Suite
- ビルドツール「Gradle」とは
- Eclipseの初期設定

Thymeleaf(タイムリーフ)とは

SpringBootで使われているテンプレートエンジン。

JSFのFaceletsでは「～.xhtml」で作るためコーディングがしづらい点があった。Thymeleafでは「～.html」として作るため、Spring Bootを介さずhtml単体での編集が可能であり、コーディングがしやすい。

ブラウザで直接HTMLを開いてプレビューできるので、デザイナーが開発に入りやすい



Thymeleaf

https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf_ja.html

概要・環境構築

- Javaフレームワーク「Spring Boot」 とは
- テンプレートエンジン「Thymeleaf」 とは
- **インストール (Eclipse)**
- Spring Tool Suite
- ビルドツール「Gradle」 とは
- Eclipseの初期設定

インストール (Eclipse)

「Pleiades All in One」のサイトより、最新版のEclipseをダウンロードする
<https://mergedoc.osdn.jp/>

MergeDoc Project

Qiita GitHub Follow @willbrains B! ブックマーク 1470

7,968,012 downloads

Pleiades 日本語化プラグイン

問題・不具合などの報告

Pleiades All in One ダウンロード

Pleiades All in One は Windows、Mac 向けに Eclipse 本体と日本語化を行うための Pleiades プラグインおよびプログラミング言語別に便利なプラグインをまとめたパッケージです。また、Full Edition の場合は JDK のインストールや環境変数の設定が不要で Eclipse の各種設定も自動で行われるため、ダウンロードして起動すれば、すぐに日本語化された Eclipse を利用できます。既にインストール済みの Eclipse に適用したい場合や、Linux で使用する場合は、All in One ではなく [Pleiades プラグイン本体](#)のみをダウンロードしてください。

Eclipse 2021 最新版	Eclipse 2020 2020/12/22	Eclipse 2019 2020/02/13	Eclipse 2018 2018/10/04 (32bit 最終版)
Eclipse 4.8 Photon 2018/06/27	Eclipse 4.7 Oxygen 2017/06/28	Eclipse 4.6 Neon 2016/06/22	Eclipse 4.5 Mars 2015/06/24
Eclipse 4.4 Luna 2014/06/25	Eclipse 4.3 Kepler 2013/06/26	Eclipse 4.2 Juno 2012/06/27	Eclipse 3.7 Indigo 2011/06/22
Eclipse 3.6 Helios 2010/06/23	Eclipse 3.5 Galileo 2009/06/24	Eclipse 3.4 Ganymede 2008/06/25	Eclipse 3.3 Europa 2007/06/29

ひと手間で写真を魅力的な映像に
グラフィックデザイナーならでは
の、動画表現を。Photoshop+After
Effects+Premiere Pro

Eclipse STS 開発環境構築

Spring Boot 入門

Jakarta EE Javadoc 日本語訳

JetBrains マニュアル日本語版

※ダウンロードしたzipファイルを解凍する際は、7-zipで解凍することが推奨されています

インストール (Eclipse)

Pleiades All in One Eclipse ダウンロード

リリース 2021-03

- 開発対象となる言語に合わせてパッケージをダウンロードしてください。
- Windows では [7-Zip](#) で解凍してください。それ以外で解凍した場合、パス長 260 文字制限により正常動作しません。
- Full Edition には Eclipse 実行用の Java が付属しているため、Java のインストールや環境変数の設定は不要です。また、Java め、すぐに [Spring Boot](#) での開発ができます。([Spring Boot 入門](#))
- 全プラットフォームの Java Full Edition および Windows 向けの各 Full Edition には各言語の処理系も含まれており、[自動でコンパイラなどのパスが自動でセット](#)されます。特に理由が無ければ、すでにコンパイラなどがインストールされている環境

plugins、features ディレクトリーに格納されたプラグイン

dropins ディレクトリーに格納されたプラグイン

JDK などの各言語のコンパイラ、ランタイムなどの処理系

		Platform	Ultimate	Java	C/C++	PHP	Python
Windows 64bit <small>32bit は 2018-09 で終了</small>	Full Edition	Download	Download	Download	Download	Download	Download
	Standard Edition	Download	Download	Download	Download	Download	Download
Mac <small>Mac 版について (Q11a)</small>	Full Edition	Download	Download	Download	Download	Download	Download
	Standard Edition	Download	Download	Download	Download	Download	Download

Eclipse 実行用 JDK 11

JDK [6u48](#)、[7u80](#)、[8u202](#)、[11.0.10](#)、[16.0.0](#)

[Lombok 1.18.18](#)

Tomcat [6.0.53](#)、[7.0.108](#)、[8.5.64](#)、[9.0.44](#)

(Win のみ) MinGW 64bit [8.1.0](#)

(Win のみ) [Python](#) [2.7.18](#)、[3.9.2](#)

(Win のみ) [XAMPP x64](#) (PHP) [7.4.16](#)

[Pleiades](#) 2021.03.21

[Autodetect Encoding](#) 1.8.5

[Platform](#) 4.19.0

[Buildship](#) 3.1.5

※JavaのFull Editionのものを選択

概要・環境構築

- Javaフレームワーク「Spring Boot」 とは
- テンプレートエンジン「Thymeleaf」 とは
- インストール (Eclipse)
- **Spring Tool Suite**
- ビルドツール「Gradle」 とは
- Eclipseの初期設定

Spring Tool Suite

Spring Framework利用のための各種プラグインを追加して作られたSpring専用の開発ツール。
最新版のEclipseには同梱されているため、特に環境構築をする必要はない。

インストールされているかは、**Eclipseの[ヘルプ] → [Eclipse マーケットプレイス]**で確認できる



springと検索した結果に表示され、
「インストール済み」と出ている

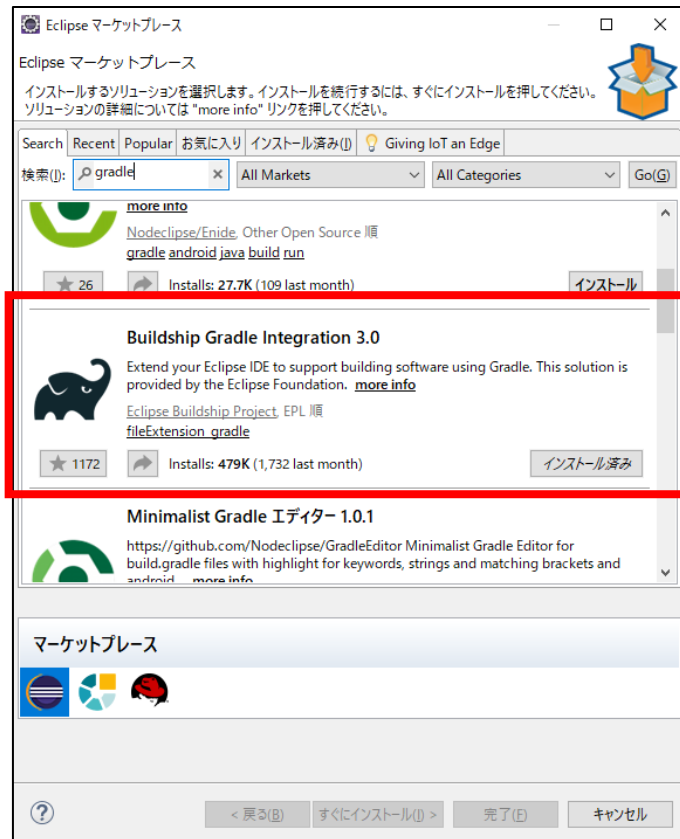
概要・環境構築

- Javaフレームワーク「Spring Boot」 とは
- テンプレートエンジン「Thymeleaf」 とは
- インストール (Eclipse)
- Spring Tool Suite
- **ビルドツール「Gradle」 とは**
- Eclipseの初期設定

Gradle(グレイドル)

Gradleは、オープンソースのビルドシステム（ビルドとは、コンパイルをして、必要なライブラリの関連づけも行って、実行可能な状態に変換すること）。

他のビルドツールにはMavenがあり、ビルド手順を定義する言語にXMLを用いる。一方Gradleは、GroovyというJavaに似た言語で定義するため、Javaプログラマーがわかりやすいメリットがある。



gradleと検索した結果で、少しスクロールすると「インストール済み」と出ている

概要・環境構築

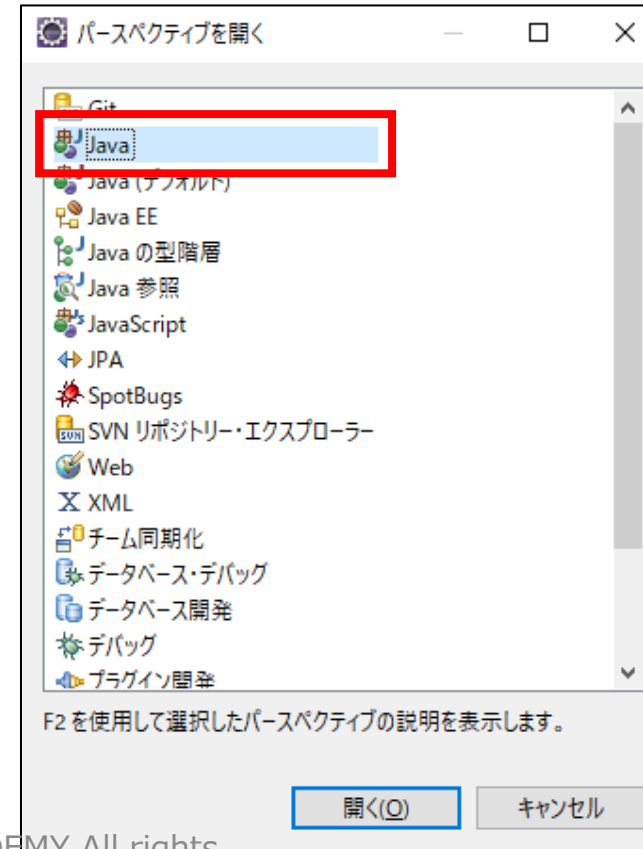
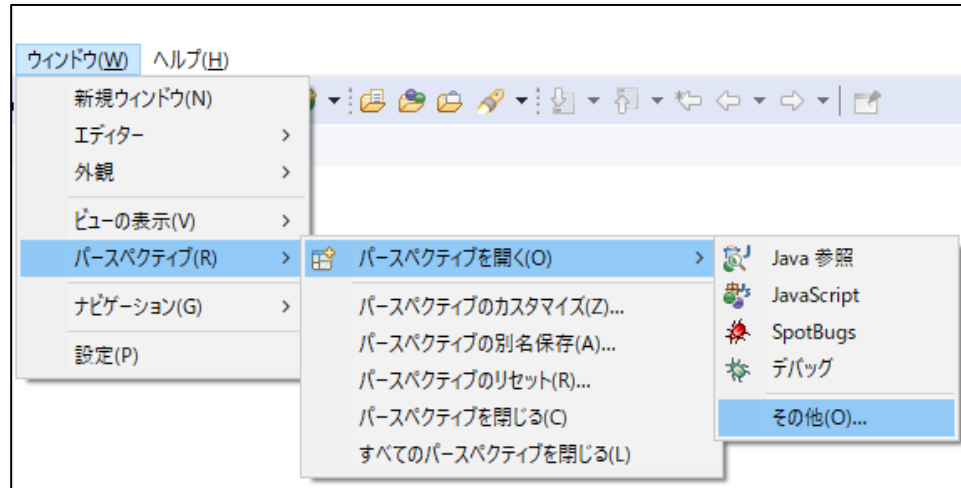
- Javaフレームワーク「Spring Boot」 とは
- テンプレートエンジン「Thymeleaf」 とは
- インストール (Eclipse)
- Spring Tool Suite
- ビルドツール「Gradle」 とは
- **Eclipseの初期設定**

Eclipseの初期設定

■ パースペクティブの設定

パースペクティブとは、Eclipseの1つ1つのパネルの配置のことです。

[ウィンドウ]>[パースペクティブ]>[パースペクティブを開く]>[その他] → **Java**

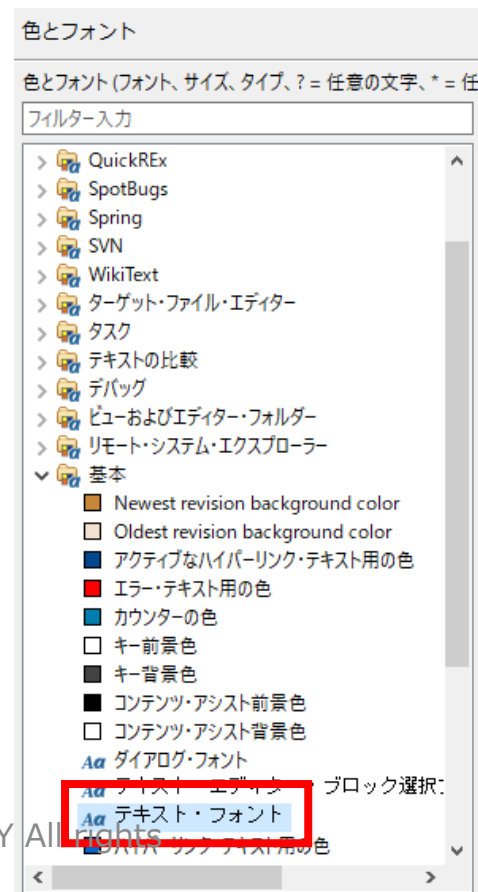
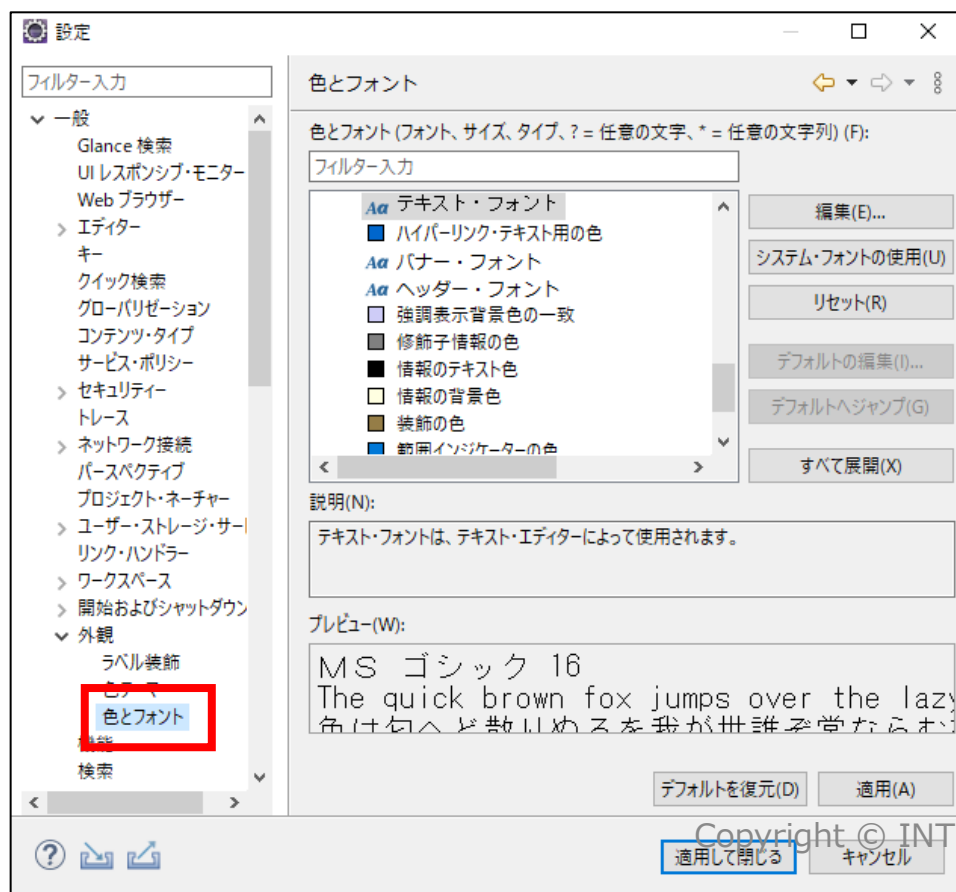


Eclipseの初期設定

■コードの文字サイズの設定

ソースコードの文字サイズの変更は環境設定で行います。

[ウィンドウ]>[設定]をクリックし、一般>外観>色とフォント → 基本>テキスト・フォント





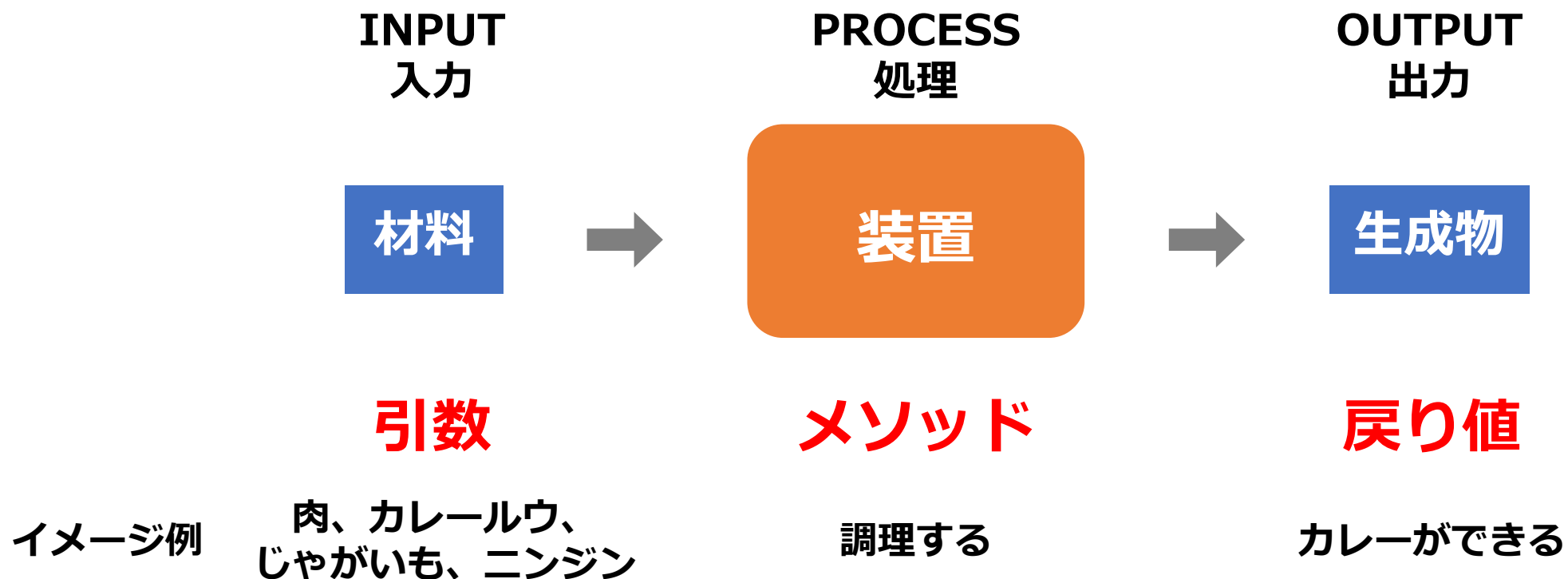
知識確認(Java)

知識確認(Java)

- メソッド
- ループ処理
- クラスとオブジェクト
- リスト
- マップ

メソッド

入力されたデータを受け取って、きめられた処理を行い、その結果が出力されるもの。ただし、入力値(引数)や出力(戻り値)はなくても構わない。



メソッド

メソッドは関数と同じ構造をしている。インスタンスの中に書かれた関数をメソッドと呼んでいるにすぎない。

メソッドの定義

引数(材料)

```
public int add(int a, int b){  
    return a + b;  
} 戻り値(生成物)
```



呼び出し

```
オブジェクト名.add();
```

メソッドの例

Keisan.Java(メソッドの定義)

```
public class Keisan {  
    public int add(int a,int b) {  
        return a+b;  
    }  
}
```

Main.java(メソッドの呼び出し)

```
public class Main {  
    public static void main(String[] args){  
        Keisan k = new Keisan();  
        int num = k.add(1,2);  
        System.out.print(num);  
    }  
}
```

ループ処理

Main.java

```
public class Main {  
    public static void main(String[] args){  
        for ( int i = 0; i < 5; i++ ){  
            System.out.print(i);  
        }  
    }  
}
```

① : 変数iに0を代入する

② : iの値が5より小さいかをチェック(0なのでTrue)

③ : 処理が実行される

④ : iの値が+ 1 されて、②に戻る

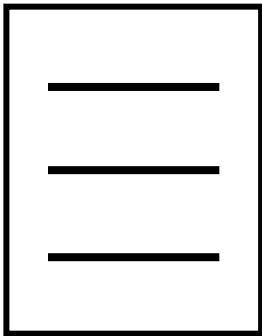
※iの値が5の時に②はFalseになり、ループを抜ける

実行結果(コンソール)

0 1 2 3 4

クラスとオブジェクト

クラス(設計図)



①クラス(設計図)
を作成する

②オブジェクト
を生成する

オブジェクト
(設計図を元に
作ったキャラ)



③オブジェクト
を操作する

足し算する

メソッド
(オブジェクトが
できる処理)

name

フィールド変数
(オブジェクトが
持つ入れ物)

クラスとオブジェクトの例

Chara.java(クラスの定義)

```
public class Chara {  
    String name;           ...フィールド変数  
    public int add(int a, int b) {  
        return a+b;       ...メソッド  
    }  
}
```

Main.java(オブジェクトの生成、操作)

```
public class Main {  
    public static void main(String[] args){  
        Chara c = new Chara(); ...生成  
        c.name = "キャラ名";    ...操作 1  
        int num = c.add(1,2);   ...操作 2  
        System.out.print(num);  
    }  
}
```

リスト(ArrayList)

仕切りのついた入れ物のようなデータ構造

中に入るデータのデータ型

```
ArrayList<String> list = new ArrayList<String>();
```

```
list.add("たなか");
```

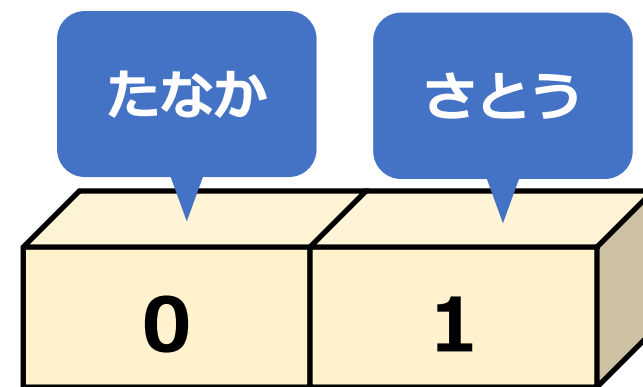
```
list.add("さとう");
```

listへデータ格納

```
list.get(0);
```

```
list.get(1);
```

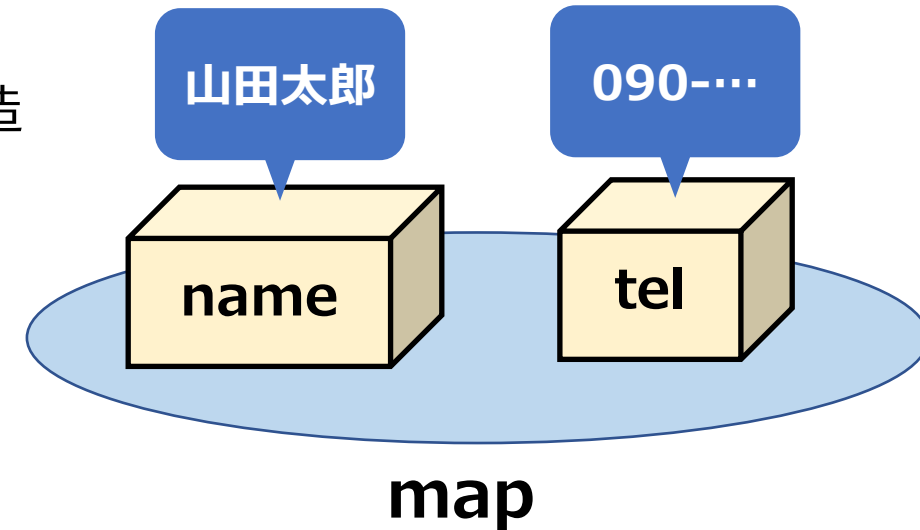
listのデータ参照



list

マップ(HashMap)

フィールド内に複数の小さな箱が存在するようなデータ構造



中に入るデータのデータ型

```
HashMap<String, String> map = new HashMap<String, String>();
```

```
map.put("name","山田太郎");
```

```
map.put("tel","0900-1111-2222");
```

```
System.out.println(map.get("name"));
```

```
System.out.println(map.get("tel"));
```

mapへデータ格納

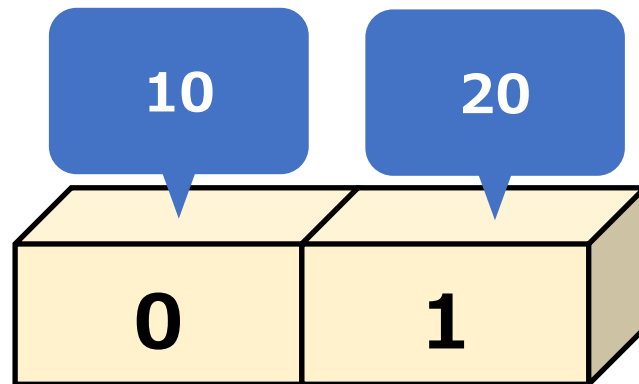
mapのデータ参照

確認テスト

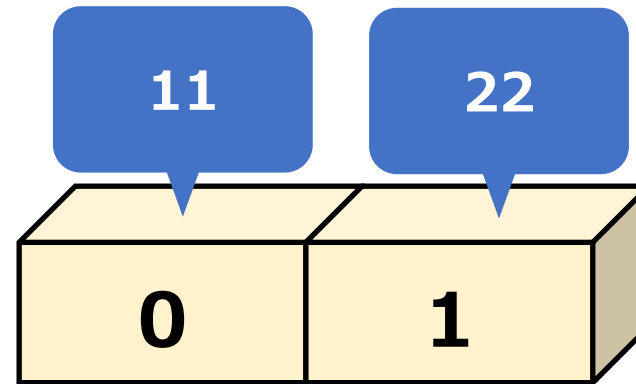
以下のようなlist1、list2を作り、0の合計値と1の合計値をコンソールに出力してください

※ループ処理を用いて計算してください

※合計値を計算するのは、Keisanクラスのaddメソッドを用いてください



list1



list2



知識確認(データベース)

知識確認(データベース)

- ・ リレーショナルデータベース
- ・ データベースとテーブル
- ・ よく使うデータ型
- ・ CRUD操作

データベースとは

コンピュータに大量の情報を保存し、効率よく検索できるよう整理したデータの集まり

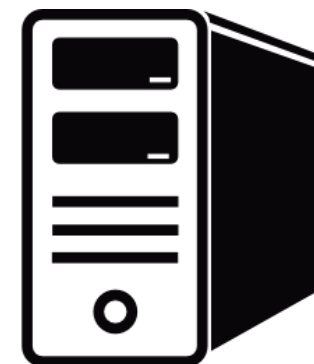
用途

ECサイトや顧客管理システムなど、ほぼシステム全般で使われている

メリット

テキストファイル(～.txt)やExcel(～.xlsx)などのファイルを使ったデータ管理と違って、

- ・ 大人数でのデータ共有がしやすい
- ・ 大量のデータを扱う場合でも、処理速度がそれほど落ちない
- ・ セキュリティ上有利であり、障害対応もしやすい



リレーショナルデータベース

データ間の関係を二次元の表形式で記述するモデル。現在、多くのDBMS(データベースマネジメントシステム)がこのモデルを採用している。

個々のデータを表す行（レコード）と、データの属性を表す列（カラム）からなる表（テーブル）を操作し、データの参照・更新を行う。

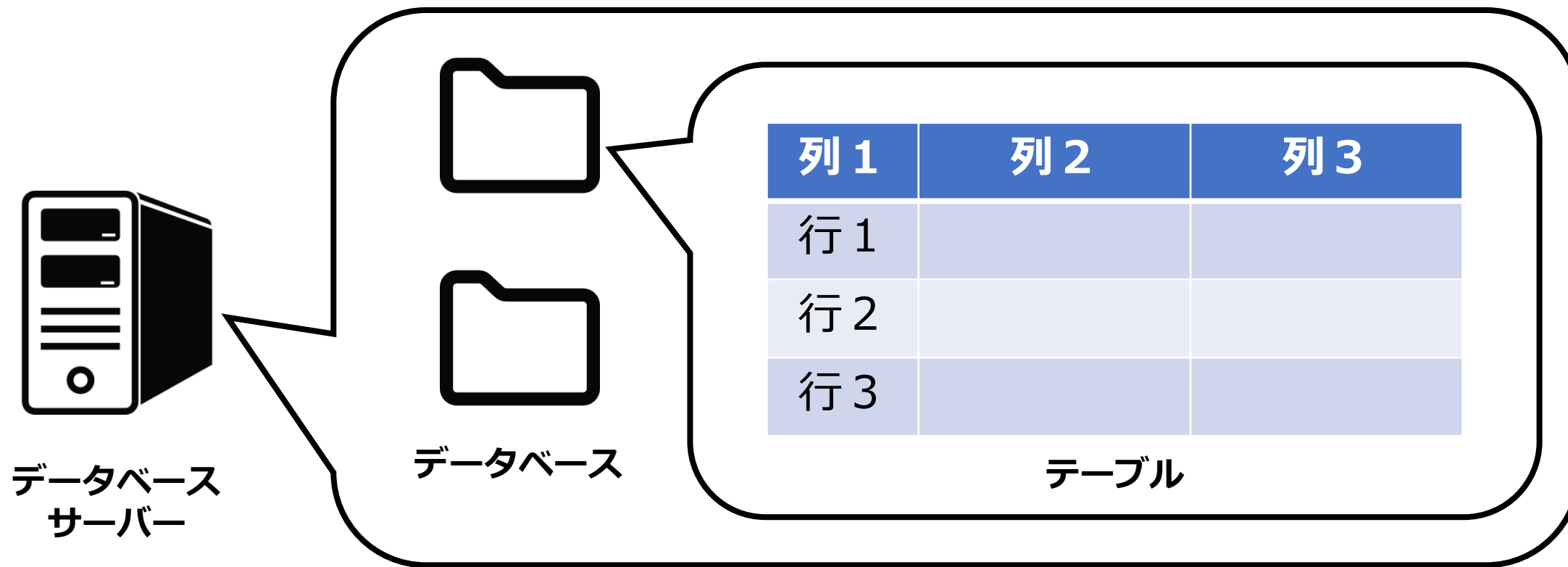
列 1	列 2	列 3
行 1		
行 2		
行 3		

テーブル(表)

レコード(行)

カラム(列)

リレーショナルデータベースの構造



！ポイント

表形式のデータであるテーブルが、データベースという入れ物に入るイメージ。
データベース、テーブルのどちらも複数作ることができる

データ型

データ型とは、データの種類のこと。テーブルのカラム(列)ごとに、データ型を指定する必要があり、決めたデータ型のデータしか保存できない。

よく使われるデータの種類と、対応するデータ型は以下のとおり。

データの種類	使うデータ型
整数	INT
文字列(備考欄など)	TEXT
小数	FLOAT
日時	DATETIME(日付 + 時刻) DATE(日付)

データベースの作成・削除

データベース作成

```
CREATE DATABASE データベース名;
```

データベース一覧を表示して確認

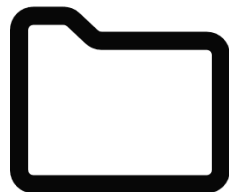
```
SHOW DATABASES;
```

データベース削除

```
DROP DATABASE データベース名;
```

データベースの選択

```
USE データベース名;
```



データベース

user_id	name	age	e-mail

テーブル

テーブルの作成・削除

テーブル作成

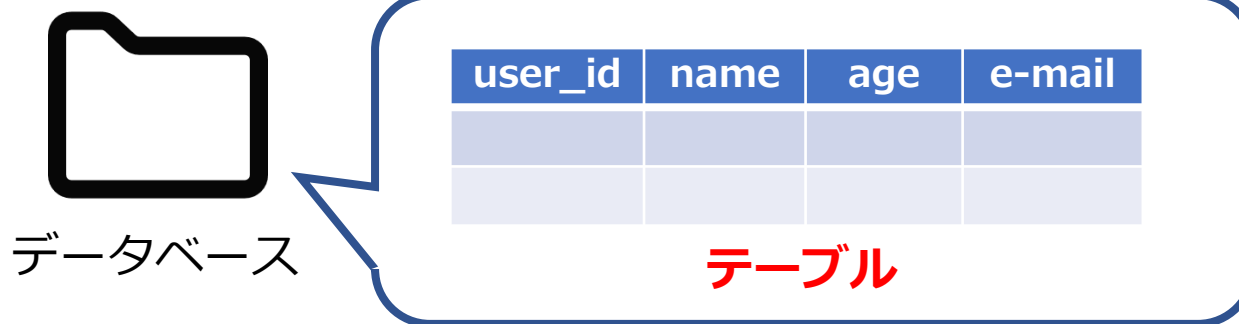
```
CREATE TABLE テーブル名 (テーブルの構成) ;
```

テーブル一覧を表示して確認

```
SHOW TABLES;
```

テーブル削除

```
DROP TABLE テーブル名;
```



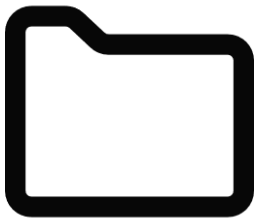
テーブルの作成

```
CREATE TABLE users (
```

```
  user_id    INT,  
  name       TEXT,  
  age        INT,  
  email      TEXT
```

```
);
```

「カラム名 データ型」の順番で指定



infoデータベース

作成されるカラム

user_id

name

age

email

usersテーブル

SQL基礎の復習

CRUD

CREATE, READ, UPDATE, DELETEの頭文字をとった用語。

データベースの基本処理を表す言葉

CRUD	SQL	操作内容
Create（作成する）	INSERT	テーブルにレコードを新規挿入する
Read（読み込む）	SELECT	表からデータを抽出する
Udata（更新する）	UPDATE	表のデータを更新する
Delete（削除する）	DELETE	表からレコードを削除する

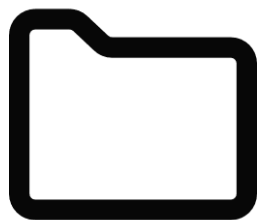
データの挿入（1つだけの場合）

- ・列名を指定する場合

```
INSERT INTO users ( user_id, name, age, email )  
VALUES ( 1, "sato", 30, "sato@test.com" );
```

- ・列名の指定を省略した場合

```
INSERT INTO users  
VALUES ( 1, "sato", 30, "sato@test.com" );
```



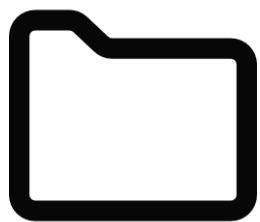
infoデータベース

user_id	name	age	email
1	sato	30	sato@test.com

usersテーブル

データの挿入（複数の場合）

```
INSERT INTO users VALUES  
( 2, "tanaka", 28, "tanaka@test.com" ),  
( 3, "suzuki", 35, "suzuki@test.com" );
```



infoデータベース

user_id	name	age	email
1	sato	30	sato@test.com
2	tanaka	28	tanaka@test.com
3	suzuki	35	suzuki@test.com

usersテーブル

データの検索

検索する列を指定する

```
SELECT name , email FROM users ;
```

検索元のusersテーブル

user_id	name	age	email
1	sato	30	sato@test.com
2	tanaka	28	tanaka@test.com
3	suzuki	35	suzuki@test.com



検索結果

sato	sato@test.com
tanaka	tanaka@test.com
suzuki	suzuki@test.com

すべての列を指定する場合

```
SELECT * FROM users ;
```

条件を指定してデータを検索

WHERE句によって条件を指定し、データを検索する

```
SELECT * FROM users WHERE age >= 30;
```

検索元のusersテーブル

user_id	name	age	email
1	sato	30	sato@test.com
2	tanaka	28	tanaka@test.com
3	suzuki	35	suzuki@test.com



検索結果

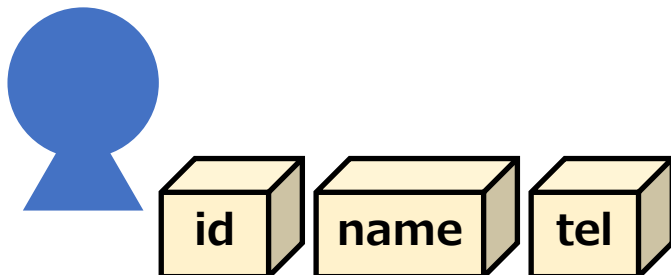
1	sato	30	sato@test.com
3	suzuki	35	suzuki@test.com

O/Rマッピング

オブジェクト(Object)とデータベース(Relational Database)を対応させる仕組み

オブジェクト
(エンティティ)

```
public class User{  
    private int id;  
    private String name;  
    private String tel;  
}
```



リレーショナル
データベース

列名	型
id	INT
name	TEXT
tel	TEXT

