

「情報I」に向けたプログラミング研修会

~情報システムとデータの管理(JavaScript版)~

アシアル株式会社 アシアル情報教育研究所 岡本 雄樹



https://edu.monaca.io/ Copyright © Asial Corporation. All Right Reserved.



自己紹介

2 名前

└ 岡本雄樹(アシアル情報教育研究所 所長)

著書

- └ イラストでよくわかるPHP
- └ WordPressプロフェッショナル養成読本
- └ Monacaで学ぶはじめてのプログラミング
- メッセージ
 - 「コンピューター」「インターネット」「プログ ラミング」私は高校生の時にそれらと出会うこと で人生が拓けました。先生方とMonacaによるアプ リ開発を通じて、情報技術の活用方法や作品作り の楽しさを広めてまいります。



- 教材サイトのご案内
 - └ あんこエデュケーション
 - https://anko.education/
 - └ 教科情報研修資料
 - https://anko.education/joho



■ プロジェクトのインポート

- 教材サイトに各種モデルのサンプル
 プロジェクトがあります
- └ Monacaでインポートして進めます
- 予めMonacaのログインと空きプロ ジェクト数の確保をお願いします

_{サンブルアブリ集} 情報 I ・第4章	ッール集 リファレンス 用語集 情報 I 研 を 情報通信ネットワークと	^{勝資料} ^{学習指導案} データの活用
❥ HOME / 共通数科植银科「储银 I 」「信银 I 」に向け	た研修資料 / 倍幅Ⅰ・第4章 情報通信ネットワークとデー3	の 添用 共通数科情報科「情報 I」 「情報 I」 に向けた研修資料
<text><text><text><text></text></text></text></text>	<text><text><text><text></text></text></text></text>	構成は2 (13)かに5100次に5100次に5100次(11) 情報 I ・第3章 コンピュータとプロ グラミング サイコロによる確単モデルのシミ ュレーション モンテカルロ法による円周車の計算 ランダムウォークのシミュレーション (ロジスティクス曲線) 進約法による預金の複利計算(確 変モデルのシミュレーション)
リレーショナルデータベース 文部科学名発行「高等 学校情報科『俳報』』 教員研修用数4.7 「さまざまな形式のデ ータとその表現形式」	WebAPIによるデータの取得 ************************************	 情報I・第4章 情報通信ネットワークとデータの活用 GISを用いたデータの可視化と問題発見〜統計GISでAED設置地域の人口密度を分析 WebAPIによるデータの取得

「Web API によるデー

JavaScriptとe-Statで

学習する方法を紹介。

タの取得」を

A \$ 1 .

データのさまざまな表現形式~離 散グラフと隣接行列

リレーショナルデータベース

```
Copyright © Asial Corporation. All Rights Reserved.
```

「リレーショナルデー

JavaScript & AlaSQLT

学習する方法を紹介。

タベース」を

■ 学習20 情報システムが提供するサービス

- └ 情報システムが提供するサービス
- └ オープンデータの重要性とその活用
- └ データの蓄積・管理・提供する方法
- └ 国や地方公共団体が提供するオープンデータ
- └ GISを用いたデータの可視化と問題発見

■ 学習21さまざまな形式のデータとその表現形式

- └ リレーショナルデータベース
- └ データのさまざまな表現形式~離散グラフと隣接行列
- └ WebAPIによるデータの取得
- └ キー・バリュー形式のデータの処理・蓄積

情報システムが提供するサービス

情報システムが提供するサービス

■ 重要なキーワード

└ ビッグデータ

└ オープンデータ

分野	情報システム	活用の例
商業	POS システム(Point Of Sales System)	店舗経営の効率化・売上分析・在庫管理
金融	電子決済システム	鉄道運賃支払,店頭レジでの支払い
通信	SNS(ソーシャル・ネットワーキング・サービス)	利用者同士の交流, 情報発信
交通	高度道路交通システム(ITS)	渋滞緩和, バスの走行位置提供
気象	緊急地震速報システム	早期避難,工場の機械制御
教育	e ラーニングシステム	教材提供, 学習履歴に基づく指導
行政	住民基本台帳システム	行政手続きの簡略化,住民票写し広域交付

図表1 情報システムが提供するサービスの例

■ オープンデータの重要性とその活用

└ オープンデータの意義や定義などについては総務省が公開している 「オープンデータ基本指針(PDF)」を参照して下さい

https://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20170530/kihonsisin.pdf

推奨データセット

- └ 内閣官房情報通信技術(IT)総合戦略室が定めた、公開を推奨する オープンデータの一覧
 - <u>https://www.kantei.go.jp/jp/singi/it2/senmon_bunka/data_ryutsuseibi/jititai_swg_dai4/siryou4-</u> <u>3.pdf</u>
- └ 本章で利用するAED設置箇所一覧は標準データセットに含まれます

商用・非商用を問わず、誰でも無償で二次利用が可能なライセンス体系が現代 には幾つか存在します

学習20

- └ クリエイティブコモンズ
 - └ CC-BY など
- └ オープンソース
 - └ MITライセンスなど
- └ パブリックドメイン
 - └ 著作権を放棄したもの

補足事項

└ クレジットの表示などが必要な場合があります

■ データの蓄積・管理・提供する方法

システム間でデータをやり取りする場合やオープンデータのような形で地方公共団体がデータを公開 する場合には、なんらかの形でデータを蓄積したり管理したり提供する必要があります。

└ ファイルによるデータの交換

わかりやすい方法の一つがファイルによるデータの交換です。ファイルならWeb上に公開するのは簡単ですしメール などでも送受信が可能です。

└ WebAPIによるデータの交換

- └ WebAPIの場合には指定されたフォーマットに基づいて特定のWebサーバーにリクエストを発行すると、それに応じた レスポンスが返ってきます。
- └ レスポンスは文字列の場合やファイルが返ってくる場合もあります。
- └ 「リクエストでファイルを送ったらレスポンスでファイルが返ってくるWebAPI」というのも実現可能です。

└ データベースサーバーによるデータの蓄積・管理・提供

- 「情報システムの裏側にはたいていデータベースサーバが存在しますが、データベースサーバーはデータを単一ファ イルで管理しているとは限りません。
- └ 複数のファイルに分割して管理している場合や、そもそもファイルという仕組みを使っていない場合もあります。

テキストファイルとバイナリファイル

ファイルには大別すると「テキストファイル」と「バイナリファイル」が存在します。 テキストファイルは仕様がOSやアプリケーションを越えて広く共通化されており、様々 なソフトウェアやプログラミング言語上で利用できます。プログラムから扱いやすいの は圧倒的にテキストファイルになります。以下に主要なテキスト形式を紹介します。

L TXT形式

└ テキストファイル全般を指すファイル形式です。

└ CSV形式

└ データをカンマ区切り(と改行)で表現している形式です。2次元の表データなどで使われます。

└ XML形式

- システム間データのやり取りやSVG画像の表現、記事更新情報のRSSやブラウザのお気に入り情報管理の OPMLなどで使われる汎用性の高いマークアップ言語です。
- └ HTMLの兄弟みたいな存在でもあります。WebAPIのレスポンス文字列にも良く使われていました。

■ 主要なバイナリファイル

テキスト形式以外のファイルは全てバイナリファイルです。SVG形式以外の画像ファイル やzip圧縮されたファイル、また、ワープロソフトや表計算ソフトなどの標準的な保存形 式もバイナリ形式です。

└ PDF形式

- PDF(Portable Document Format)はアドビシステムズが開発した文章を扱うためのフォーマットの一つ で現在は国際標準規格にもなっています。1つのファイル上に画像や図形などが埋め込め、また、文字列 を画像ではなく文字列として保持することができ、フォントの埋め込みも可能です。
- Portableとある通り、コンピューター側にPDFリーダーソフトが入っていればファイルを開くことができます。PDFリーダーソフトはアドビシステムズが無料で配布しているほか、最近ではブラウザに標準で組み込まれています。

└ Excel形式

└ Microsoft社のMicrosoft Excelのファイル形式です、拡張子はxlsx。CSV形式では保持できないシートの 情報やグラフなどの情報も単一ファイルで管理できます

└ Word形式

└ Microsoft社のMicrosoft Wordのファイル形式です、拡張子はdocx。

国や地方公共団体が提供するオープンデータ

前述の通り「標準データセット」として地方公共団体はオープンデータを公開していま す。以下に一例を列挙しますが、自身でもお住まいの地域や出身地のオープンデータを 探してみて下さい。

- └ <u>川越市 Open Data Saitama</u>
- └ <u>和歌山市オープンデータ</u>
- └ <u>和歌山県 オープンデータ提供</u>



Wakayama Prefecture

Google検索

GitHubアカウント「Wakayama Prefecture」について

オープンデータとは、国や地方公共団体が公開する行政情報等について、コンピュータで扱いやすいデータ形式で、二次利用が 可能なルールで提供さ れるデータのことです。近年、各国政府や地方公共団体等において、このオープンデータの提供拡大や、オープンデータを経済活性化や官民協働等に 活用する取組が進められているところです。和歌山県では、平成27年2月にオープンデータのニーズ把握や、技術者の方等と協働した地域の再発見や 課題解決等の推進のため、ソフトウェア開発プロジェクトのための共有ウェブサービス「GitHub」にアカウント「Wakayama Prefecture」を開設 し、平成28年10月からオープンデータサイトとしての本格運用を開始しました。

新型コロナウイルス感染症に関連する情報

県内の最新感染動向(陽性患者数の推移、PCR検査の状況)

統計情報

統計情報(調査統計課)

統計年鑑【平成27年度刊行】

観光統計(観光振興課)

広報・観光情報

県民の友・おしらせ(広報課)

- 県民の友・催し(広報課)
- わかやま観光情報(観光連盟)
- WAKAYAMA FREE Wi-Fi

調達情報

- <u>公共工事等入札情報システム(技術調査課)</u>
- <u>公共工事等に関する入札結果(技術調査課)</u>
- 物品電子調達システム(総務事務集中課)
- 和歌山県役務等公開システム(総務事務集中課)

地理空間情報

地理空間情報(情報政策課)

Wakayama Prefecture. All Rights Reserved.

https://wakayama-pref-org.github.io/

GISを用いたデータの可視化と問題発見 ~統計GISでAED設置地域の人口密度を分析

GISとは

GIS (Geographic Information System) は地理情報システムのことです。地図上に データを展開して分析などを行える統合的なシステムです。

地図アプリとはなにが違うのか?

地図アプリは主に、路線検索や移動経路の確認・ナビ機能など「移動に関する問題」 を解決するための手段として使います。「効率的に移動したい!」という問題はすでに 発見されています。また、地図アプリ側に駅間のネットワーク情報などが登録されてい る前提で利用します。

GISは地理的な問題の発見と解決を行える汎用的な情報システムとなります。地図デー タの表示機能や緯度経度情報を視覚的に表示する機能は備わっていますが、汎用ツール なので問題の仮説は自分で立てる必要がありますし、分析用のデータも自分で探してく る必要があります。

GISの入手方法

GISのソフトウェアは無料で使えるものからブラウザで動作するもの、商用製品まで多 数存在します。今回は「e-Stat 政府統計の総合窓口」にある、「統計GIS(jSTAT MAP)」 を使います。

統計GISを使った実習の流れ

文科省の教員研修資料で示されている実習では、「統計GIS」を使って川越市の人口密 度とAED設置場所を地図上に表示します。

「e-Stat 政府統計の総合窓口」の基本機能は登録不要で使えますが、「統計GIS」は 別途アカウントの登録が必要となります。また、人口密度は国勢調査のデータなので統 計GISから参照できますが、AEDの設置データは埼玉県のオープンデータポータルサイト から別途取得する必要があります。

- 実習の流れ
 - └ データの準備
 - └ アカウント作成
 - └ 地図の描画
 - └ 地図の分析

■ データの準備

地図上に展開させたいデータを用意します。文部科学省の研修教材では「川越市のAED 設置場所データ」を題材にしています。川越市のデータで動作が確認できた暁には、ご 自身に関係のある地域のデータでも挑戦してみて下さい。

川越市のAED設置場所データを用意する

<u>"【川越市】AED設置場所(CSVデータ)令和2年3月30日現在(UTF-8)" by Saitama</u> Prefectrure, used under CC BY / 一部の列を削除しつつ並び替えたもの

このデータにはAEDの設置場所名称と経度と緯度が記録されています。列の末尾に「経 度」「緯度」の順で列挙するのがポイントです。

設置場所_名称	経度	緯度
川越市役所本庁舎	139.4856666	35.9251276
川越市役所東庁舎	139.4861841	35.92513661
川越市役所庁舎分室	139.4865436	35.9252954
川越市北部地域ふれあいセンター	139.4803489	35.94376325
川越市東部地域ふれあいセンター	139.5226993	35.9021551

■ アカウントの作成

こちらの解説は省略します。

「e-Stat 政府統計の総合窓口」にある、「統計GIS(jSTAT MAP)」を選択します。



地図で見る統計 (jSTAT MAP)を選択します。



データの取り込みや地域分析レポート作成にはログインが必要という案内が表示されます。

地図で見る統計(jSTAT MAP) X		
地図で見る統計 (jSTAT MAP) は、誰でも使えるWebサイトの地理情報システムです。 都道府県、市区町村、小地域、地域メッシュ統計の結果を地図に表示することができます。 また、地域の分析が可能となるように、以下の機能を提供します。 ・利用者の保有するデータを取り込んで分析する機能 ・任意に指定したエリアにおける統計算出機能 ・地域分析レポート作成機能		
【ログインすると利用できる機能】 ・ジオコーディング、リッチレポートなどが利用できます。 ・利用者が取り込んだデータが保存できます。		
ログインしないでGISを始める		
【ログイン方法】 以下のアカウントをお持ちの方は、「ログイン」ボタンからログインしてください。 お持ちでない方はアカウントを作成してください。 ・jSTAT MAPユーザID(平成29年12月末までに取得された方) ・政府統計の総合窓口のアカウント		
ログイン アカウントを作成する		

統計GISを起動して「川越」で検索した結果。なお、地図の種類は「地理院地図」を選択 していますが、初期状態では「Google Maps」が表示されます。

次に「統計地図作成」を選択して、川越市の人口密度とAEDの場所を地図上に視覚化していきます。



先ほど用意したAEDの場所データを統計GISに取り込みます。「統計地図作成」を選択して から「プロット作成」を選択して下さい。



「緯度経度付きファイル」を選択します。※ なお、緯度経度さえあればAEDに限らず観光 スポット情報でも和菓子屋の情報でもなんでも取り込めます。

X
す。
与して、地図上にプロットを作成する方法です。
こプロットを作成する方法です。

ファイルアップローダーが起動します。注意書きの一つとして緯度経度の並びを「経度・ 緯度」の順で末尾に付けるように記載されています。また、「文字コード」にも言及があ り、「UTF-8」形式でアップロードする場合にはチェックボックスへのチェックが必要と なります。

怒度付きつ;			ち注です						
込む緯度経	度付きCSVJアイル(.csv)を指定し(トさい。						
SVファイル:	kawagoe_cus	tom.csv						参照	
	χ ,		SC9. UIF	-00% 高		990CF970-			
)込むCSVフ	ファイルは以下のような	ネフォーマットとしま	す。						
								L	
					7	1日本ので経度、	緯度		
					¥	床尾は必ず経度、	、緯度		
店舗名	住所	電話番号	面積 駐車	國 売上	URL	k尾は必ず経度、 経度	、緯度 緯度		
店舗名 S	住所 S	電話番号 S	面積 駐車 N N	i数 売上 N	URL U	R尾は必ず経度、 経度 N	、緯度 ^{緯度}		
店舗名 S 新宿店	住所 S 東京都新宿区〇〇	電話番号 S 03-1234-5678 02-1224-0976	面積 駐車 N N 100	i数 売上 N 5 10	URL U 0 http://www	R尾は必ず経度、 経度 N 139.691713 129.642224	、緯度 N 35.6894431 25.4475001		
店舗名 S 新宿店 高田馬場店	住所 S 東京都新宿区〇〇 東京都新宿区〇〇	電話番号 S 03-1234-5678 03-1234-9876	面積 駐車 N N 100 120	·数 売上 N 5 10 10 9	URL U 0 http://www 6 http://www	経度 N 139.691713 139.642324	、緯度 N 35.6894431 35.4475001		
店舗名 S 新宿店 高田馬場店	住所 S 東京都新宿区〇〇 東京都新宿区〇〇	電話番号 S 03-1234-5678 03-1234-9876	面積< 駐車 N N 100 120	i数 売上 N 5 10 10 9 性項目の3	ス URL U U http://www・・・ 6 http://www・・・ 型を指定する場合は	程度 N 139.691713 139.642324 2行目に記載する	、緯度		
店舗名 S 新宿店 高田馬場店	住所 S 東京都新宿区〇〇 東京都新宿区〇〇	電話番号 S 03-1234-5678 03-1234-9876	面積<	i数 売上 N 5 10 10 9 性項目の	ス URL U 0 http://www・・・・ 16 http://www・・・・ 型を指定する場合は	R は 必ず 経度 経度 N 139.691713 139.642324 2行目に記載する。	韓度 N 35.6894431 35.4475001 。(省略可)		
店舗名 S 新宿店 高田馬場店 (世界測地3	住所 S 東京都新宿区〇〇 東京都新宿区〇〇	電話番号 S 03-1234-5678 03-1234-9876	面積 駐車 N N 100 120	i数 売上 N 5 10 10 9 性項目の	ス URL U 0 http://www・・・・ 16 http://www・・・・ 型を指定する場合は	R 定 総度 N 139.691713 139.642324 2行目に記載する。	韓度 N 35.6894431 35.4475001 。(省略可)		
店舗名 S 新宿店 高田馬場店 (世界測地3 (最初の属性	住所 S 東京都新宿区〇〇 東京都新宿区〇〇	電話番号 S 03-1234-5678 03-1234-9876 ジ記載下さい。	面積< 駐車 N N 100 120	i数 売上 N 5 10 10 9 性項目の す。	ス URL U 0 http://www・・・・ 6 http://www・・・・ 型を指定する場合は	R は 必ず 経度 経度 N 139.691713 139.642324 2行目に記載する。	、緯度 N 35.6894431 35.4475001 。(省略可)		
店舗名 S 新宿店 高田馬場店 (世界測地3 (最初の属性 (取り込み可	住所 s 東京都新宿区〇〇 東京都新宿区〇〇 (10進の経度緯度を まが名称、末尾の29 「能な件数は最大20	電話番号 S 03-1234-5678 03-1234-9876	面積< 駐車 N N 100 120	■数 売上 N 5 10 10 9 性項目の す。 のテ [×] -9(よ	7 URL U 0 http://www・・・・ 16 http://www・・・・ 型を指定する場合は 取り込まれません。	R は 必ず 経度 経度 N 139.691713 139.642324 2行目に記載する。	、緯度 ^N 35.6894431 35.4475001 。(省略可)		
店舗名 s 新宿店 高田馬場店 (世界測地3 (最初の属性 (取り込み可 (属性項目の	住所 s 東京都新宿区〇〇 東京都新宿区〇〇 (10進の経度緯度を 5が名称、末尾の29 1能な件数は最大20 の型を指定する場合	電話番号 03-1234-5678 03-1234-9876 記載下さい。 が必ず、経度、新 00件であり、それ は、2 行めを型指	面積< 駐車 N 100 120 [] 緯度となりま [] 108 [] 100 []	 数 売上 N 5 10 9 性項目の す。 のデ[*] - 9(よ) ます。 	ス URL U D http://www・・・・ M たちょうる場合は 取り込まれません。	R 定 総度 N 139.691713 139.642324 2行目に記載する。	、緯度 N 35.6894431 35.4475001 。(省略可)		
店舗名 s 新宿店 高田馬場店 (世界測地3 (最初の原性 (取り込み可 (型を指定し	住所 s 東京都新宿区〇〇 東京都新宿区〇〇 東京都新宿区〇〇 (10進の経度緯度を 5が名称、末尾の29 「能な件数は最大20 の型を指定する場合 ない場合は、経度、	電話番号 S 03-1234-5678 03-1234-9876 注記載下さい。 別が必ず、経度、約 00件であり、それ は、2 行めを型指 緯度以外は文字	 画積 駐車 N N 100 120 属 緯度となりま 1を超えた分 皆定の行とし ■属性として) 	i数 売上 N 5 10 10 9 性項目の す。 のデ [、] -タは ます。 取り込まれ	オ URL U 00 http://www・・・ 66 http://www・・・ 型を指定する場合は 取り込まれません。 れます)	ELL公ず経度、 経度 N 139.691713 139.642324 2行目に記載する。	、緯度 N 35.6894431 35.4475001 。(省略可)		

アップロードする

アップロードに成功してもデータの取り込みに成功するとは限りません。取り込み完了画 面のメッセージを確認して下さい。失敗した場合は結果ファイルをダウンロードすること で失敗の理由を確認できます。

また、「登録する」を押下するまえに「アイコン選択」を変えることをお勧めします。 AEDは「人型」だと違和感がありますので、適当に別のアイコンを選択しましょう。



登録すると地図上にAEDがマッピングされます。もし何も出なければ表示している場所が「川越」になっているかを改めて確認して下さい。



■ 人口密度を描画する

次に川越市の町丁・字等別人口密度を描画します。この情報は川越市のオープンデータで はなく、政府の国勢調査データを参照します。国勢調査データはe-statから参照できるた め、CSV取り込みなどは不要です。「統計地図作成」を選択してから「統計グラフ作成」 を選択して下さい。



■ 人口密度を描画する

統計グラフ作成の画面が表示されます。以下の項目を選択して下さい。

- 調查名:国勢調查
- 年:2015(最新のものを選択、2020などがあればそれでもOK)
- 集計単位:[小地域(町名・字等別)]
- 統計表:[人口性比、密度]
- 指標:[密度(人口総数)]

統計 グラフ作成 作成するグラフの指標を選択して下さい。	×
語十データ ユーザデータ お気に入り	
 統計調査(集計) 国勢調査 統計表 住宅の種類・所有の関係別一般世帯数 住宅の建て方別世帯数 人口性比、密度 	2015年 ✓ 小地域(町丁・字等別) ✓ 指標 ○ 人口性比 ✓ 密度(人口総数) ○ 密度(男)
年齡別割合(5歲階級、4区分)	 密度(女) ▼ 指標選択 選択解除
選択指標/データ 統計データ/グル あ気に入りに追加	ループ 次へ

Ⅰ 人口密度を描画する

次は集計単位の選択に画面が切り替わります、集計範囲を「行政界選択」にしたうえで埼玉県川越市を選択します。そして「集計開始」を選択します。

統計グラフ作り	成 集計単位を選択して下さい。		×
集計単位	עד 🔿	集計範囲 〇 画面表示範囲	● 行政界選択 ○ 全国
行政界単位	市区町村 ~	選択した行政界	集計イメージ
山形県	さいたま市緑区	川越市	
福島県	さいたま市岩槻区		
茨城県	熊谷市		
栃木県	川口市		
群馬県	行田市		
埼玉県	秩父市		
 按分しない 	() 按分する		
(小地域)工!	リア範囲で集計する 選		~
			戻る 集計開始

統計地図の作成結果

以下のような統計地図が生成されます。人口密度が高い割にAEDが一つも無いエリアなど が確認できると思います。



さまざまな形式のデータとその表現形式

Copyright © Asial Corporation. All Rights Reserved.

情報システムが提供するサービス

学習21

■ リレーショナルデータベース

リレーショナルデータベース(関係データベース)はIBMのエドガー・F・コッド氏が発明した関係モデルを元にしたデータベースで、商用のリレーショナルデータベースが最初に登場したのは1979年と比較的古いですが、現在でも主要なデータベースの一つです。

関連技術

リレーショナルデータベースにはデータベースの表を設計するための正規化理論やER 図、データを取得するためのSQL(Structured Query Language)という強力な問い合わせ 言語が備わっています。

研修内容

蔵書目録データベースを作成してSQL文でデータを操作します。

蔵書目録データベースのER図

ER図は表と表の関係を表せる図です。以下の図はブラウザ上で動作する www.sqldesignerにて作成した図です。汎用的な作図ツールでも作成できますが、メンテ ナンスが手間になります。また、本格的なER図では1:多の関係を表す矢印も自動的に表 示できます。



Copyright © Asial Corporation. All Rights Reserved.

■ 蔵書データベースのデーター覧(結合後)

理解はしやすいが、作者名の変更があったときに更新漏れが発生す る可能性があります。また、作者の生年や没年など、列の情報を増や していくと維持管理が難しくなってきます。

書籍ID	書籍名	書籍作者ID	作者名
1	羅生門	1001	芥川龍之介
2	こころ	1002	夏目漱石
3	坊っちゃん	1002	夏目漱石
4	鼻	1001	芥川龍之介
5	舞姫	1003	森鴎外
6	吾輩は猫である	1002	夏目漱石

■ 書籍テーブルと作者テーブル(結合前)

作者を分離すれば作者名を後から変えることが容易となります。また、作者の情報を扱うための列を増やしても安心です。

書籍ID	書籍名	書籍作者ID
1	羅生門	1001
2	こころ	1002
3	坊っちゃん	1002
4	鼻	1001
5	舞姫	1003
6	吾輩は猫である	1002

書籍作者ID	作者名
1001	芥川龍之介
1002	夏目漱石
1003	森鴎外

■ SQLの基本

作者を分離すれば作者名を後から変えることが容易となります。また、作者の情報を扱うための列を増やしても安心です。

命令	意味
CREATE	テーブルを作成する
DROP	テーブルを削除する
INSERT	テーブルにレコードを登録する
SELECT	テーブルからレコードを取得する
DELETE	テーブルのレコードを削除する
UPDATE	テーブルの値を更新する

学習21

■ 選択・射影・結合について

リレーショナルデータベースからデータを選択して取得することを 「選択」と呼びます、具体的にはWHERE句でレコード(行)の絞り込み を行うことを指します。

また、射影はフィールド(列)の絞り込みを行うことを指します。 「SELECT * …」と記述した場合には全てのフィールドを取得しますが 例えば「SELECT id …」と記述した場合にはIDのみを取得します。

「結合」は複数のテーブルを「JOIN句」で結合することを指します。

サンプルプロジェクトをインポートし、データベースを構築して下さい。

```
let result;
// テーブル定義とデータのINSERT
alasql("CREATE DATABASE zosho");
alasql("USE zosho");
alasql("CREATE TABLE author (id INT, name TEXT)");
alasql("INSERT INTO author VALUES (1001, '芥川龍之介')");
alasql("INSERT INTO author VALUES (1002, '夏目漱石')");
alasql("INSERT INTO author VALUES (1003, '森鴎外')");
alasql("CREATE TABLE book (id INT, name TEXT, author_id INT)");
alasql("INSERT INTO book VALUES (1, '羅生門', 1001)");
alasql("INSERT INTO book VALUES (2, 'こころ', 1002)");
alasql("INSERT INTO book VALUES (3, '坊っちゃん', 1002)");
alasql("INSERT INTO book VALUES (4, '鼻', 1001)");
alasql("INSERT INTO book VALUES (5, '舞姫', 1003)");
alasql("INSERT INTO book VALUES (6, '吾輩は猫である', 1002)");
// データのSELECT
result = alasql("SELECT * FROM author");
console.log(result);
result = alasql("SELECT * FROM book");
console.log(result);
result = alasql("SELECT * FROM book INNER JOIN author ON book.author id = author.id");
console.log(result);
```

出力結果



■ INSERT文によるレコードの追加

alasql("INSERT INTO book VALUES (7, '高瀬舟', 1003)")

■ DELETE文によるレコードの削除

alasql("DELETE FROM book WHERE ID = 7");

※ DELETEを行う際にはWHERE句を使って削除対象を絞り込む必要があります、WHEREを付け忘れた場合には無慈悲にテーブルのデータが全て削除されます。

WebAPIによるデータの取得

WebAPIによるデータの取得

第三章では事前登録が不要の郵便番号WebAPIを使用していましたが、一般的にWebAPI は開発者登録などを必須としているものが多いため、第四章では登録が必要なWebAPIを 題材にしています。

本研修資料で扱うWebAPIについて

学習20で扱った「e-Stat 政府統計の総合窓口」は会員登録をすると「統計GIS」だけ でなく「WebAPI」機能も使えるようになるため、本研修資料では登録が必要なWebAPIの 例として「e-Stat 政府統計の総合窓口」を使用します。

研修内容

サンプルプロジェクトにアプリケーションキーを登録してWebAPIを呼び出します。

e-StatのWebAPI機能・管理画面

e-Statに会員登録をしてログインすると「API機能」が使えるようになります。以下は マイページの中にあるAPI機能の画面です。

マイページ表示

統計データ 新着情報	公表予定 レイアウトー覧 統計較込み条件 検索屈歴 ダウンロード屈歴 API機能
アプリケーションIDの	取得
明 *名称	sample 987
*URL	https://anko.education/ ※http://localhostやプライベートIPアドレス(127.0.0.1等)は登録できません
概要	廃止
appId	34eeabefad1b4
*名称	発行
*URL	※http://localhostやプライベートIPアドレス(127.0.0.1等)は登録できません
概要	· · · · · · · · · · · · · · · · · · ·
appId	
*名称	発行
*URL	変更 ※http://localhostやブライベートIPアドレス(127.0.0.1等)は登録できません
概要	· · · · · · · · · · · · · · · · · · ·
appId	

e-StatのAPIでデータを取得する

e-Statには様々な統計データが保存されています。今回は2020年7月27日現在の最新の 国勢調査(つまり2015年のデータ)から全国の人口をAPI経由で取得してみたいと思います。

参照する統計データ 政府統計名:国勢調査 提供統計名:平成27年国勢調査 提供分類 |:最終報告書「日本の人 ロ・世帯」統計表

上記統計データを参照するためのリンク

選択条件: デ	ータベース 🗙	/ 人口・世帯 🗙 / 国勢調査 🗙 / 平成27年国勢調査 🗙 / -	× / 最終報告書	「日本の人口・世帯	訂 統計表 🗙		
					政府統計一覧に見	<u> える(すべて解除)</u>	
データセット	▼ キーワード	を入力			x Q 検	索条件の保存	
· 検索オプション ✓ 提供分類、表	 2 題を検索 テ	データベース、ファイル内を検索				<u>検索のしかた</u>	
保存した検索条件							
データセット一覧							
<u>(戻る</u>						一覧形式で表示	
政府統計名		国勢調査				0	
提供統計名		平成27年国勢調査					
提供分類1		最終報告書「日本の人口・世帯」統計表					
表番号		統計表	調査年月	公開(更新)日	表示・ダウン	D-F	
1	人口及び人口	の割合-全国,全国市部・郡部(大正9年~平成27年)	2015年	2020-05-22	🗕 DB 😂 🗢 API		
2	面積, 面積の 27年)	割合及び人口密度 – 全国, 全国市部・郡部(大正 9 年~平成	2015年	2020-05-22	🔿 DB 🛛 🗢 API		
3	人口,面積及び人口密度-全国※,全国市部※・郡部※(昭和35年~平成 2015年 2020-05-22 → DE				🔿 DB 🛛 🌣 API		

	27年)			
3	人口,面積及び人口密度-全国※,全国市部※・郡部※(昭和35年~平成 27年)	2015年	2020-05-22	→ DB 🗘 🗘 API
4	人口及び人口の指数-全国,都道府県(大正9年~平成27年)	2015年	2020-05-22	DB 🗱 API
5	5年ごとの人口増減-全国,都道府県(大正9年〜平成27年)	2015年	2020-05-22	🔿 DB 🛛 🗢 API
6	面積及び人口密度-全国,都道府県(大正9年~平成27年)	2015年	2020-05-22	→ DB 🗖 🌣 API
7	人口及び人口の割合-全国,全国市部・郡部,都道府県,都道府県市部・郡 部(大正9年〜平成27年)	2015年	2020-05-22	DB 🗘 API
8	人口,面積及び人口密度-全国※,都道府県※(昭和40年~平成27年)	2015年	2020-05-22	→ DB 🌣 API

■ e-StatのWebAPI機能・管理画面

「DB」をクリックするとブラウザ上で都道府県別の人口が5年毎に表示されます。

統計表・グラフ表示								
統計名				国勢調査 平成27年国勢調	1査 最終報告書「日本の人	ロ・世帯」統計表		
表番号				1				
表題				人口及び人口の指数-全国	国,都道府県(大正9年~	平成27年)		
							•	
		≡ #	統計表表示			□ グラフ表示	7	0
表	表章項目		~	再表示	1.例表示			
項			1920年	1925年	1930年	1935年	1940年	1945年
選	全国		55,963,053	59,736,822	64,450,005	69,254,148	73,114,308	71,998,104
択	北海道		2,359,183	2,498,679	2,812,335	3,068,282	3,272,718	3,518,389
>	青森県		756,454	812,977	879,914	967,129	1,000,509	1,083,250
	岩手県		845,540	900,984	975,771	1,046,111	1,095,793	1,227,789
4	宮城県		961,768	1,044,036	1,142,784	1,234,801	1,271,238	1,462,254
7	秋田県		898,537	936,408	987,706	1,037,744	1,052,275	1,211,871
۲ ۲	山形県		968,925	1,027,297	1,080,034	1,116,822	1,119,338	1,326,350
設定	福島県		1,362,750	1,437,596	1,508,150	1,581,563	1,625,521	1,957,356
~	茨城県		1,350,400	1,409,092	1,487,097	1,548,991	1,620,000	1,944,344
1	栃木県		1,046,479	1,090,428	1,141,737	1,195,057	1,206,657	1,546,355
	群馬県		1,052,610	1,118,858	1,186,080	1,242,453	1,299,027	1,546,081
	埼玉県		1,319,533	1,394,461	1,459,172	1,528,854	1,608,039	2,047,261
	千葉県		1,336,155	1,399,257	1,470,121	1,546,394	1,588,425	1,966,862
	東京都		3,699,428	4,485,144	5,408,678	6,369,919	7,354,971	3,488,284

Copyright © Asial Corporation. All Rights Reserved.

e-StatのWebAPI機能・管理画面

「API」をクリックすると以下のようなポップアップが表示されます。

API リクエスト URL

クエリー

http://api.e-stat.go.jp/rest/2.1/app/getStatsData? appId=&lang=J&statsDataId=0003411191&metaGetFlg=Y&cntGetFlg=N&sec tionHeaderFlg=1

閉じる

http://api.e-stat.go.jp/rest/2.1/app/getStatsData
?appId=&lang=J&statsDataId=0003411191
&metaGetFlg=Y&cntGetFlg=N&ionHeaderFlg=1

API解説

- 「http://」は暗号化無しの通信です。「https://」に書き換えれば暗号化されます。
- restはAPI設計の一つです
- 2.1はAPIのバージョンを表しています、e-StatのAPIは現在「3.0」が最新です。
- 「http://api.e-stat.go.jp/rest/2.l/app/getStatsData」までがAPIのエンドポイントです。 getStatsDataというのはe-Statのデータを取得しますという意味でしょう。このエンドポイントに対して、更にappIdなどのパラメーターを付与して参照します。
- 「appId=」の=に続けて、マイページで取得したappIDを付与します。これによってe-Stat側は「誰が WebAPIを呼び出したのか」を認識できます。
- 「statsDataId=0003411191」は統計データを参照しています。「0003411191」はe-Statの内部では、国 勢調査 平成27年国勢調査 最終報告書「日本の人口・世帯」統計表 を指しています。

例:埼玉県の人口を取得する(XML版)

パラメーターに「cdArea=11000」を追加することで埼玉県のデータだけを取得できま す。以下のパラメーターに各自appIdの値を追加して実行してみて下さい。





学習21

例:埼玉県の人口を取得する(XML版)

XML形式は汎用的なマークアップ言語として様々なところで使われているのですが、タ グの分だけデータ量が増えてしまったり、プログラムで利用する際にタグを解析(パー ス)するコストが重いため、JSONという形式がWebAPIの世界では広まっています。

JSONなら簡単にプログラミング言語の連想配列に変換してプログラム中で利用ができま す。なお、JSONはJavaScript Object Notationの意味で、つまりJavaScriptのオブジェ クト表記なのですが他の言語でもサポートが進んでいます。

https://api.estat.go.jp/rest/3.0/app /json/getStatsData?appI d=&lang=J&statsDataId=0 003411191&cdArea=11000& metaGetFlg=Y&cntGetFlg= N§ionHeaderFlg=1

["GET_STATS_DATA"+ ["RESHLT"+ ["STATUS"+0_"FRR0R_MSG"+"¥LARA3XL6E33XL6D3XL7D42XL4E36X
27T09:44:30_618+09:0071, "PARAMETER":[71 ANG": "J", "STATS DATA ID": "0003411191", "NARROW
{"CODE_AREA_SELECT":11000}, "DATA_FORMAT":"J", "START POSITION":1, "METAGET FLG":"Y", "C
{"RESUET INF":{"TOTAL NUMBER":42,"FROM NUMBER":1,"TO NUMBER":42},"TABLE INF":{"@id":
[~@code~:~~00200521~,~~\$~:~¥u56FD¥u52E2¥u8ABF¥u67FB~},~~GOV_ORG~:
{"@code":"00200","\$":"¥u7DCF¥u52D9¥u7701 <u>"</u> },"STATISTICS_NAME":"¥u5E73¥u621027¥u5E74¥u
¥y6700¥y7U42¥y5831¥u544A¥u66F8¥u300C¥u65E5¥u672C¥u306E¥u4EBA¥u53E3¥u30FB¥u4E16¥u52F
PSEXUE/3 FUDE/1/2/1/2012/4 MDE/4 FUTE/3 / VILLE - SURVET DATE / VILLE
ZZ , SMALL AREA :U, UULLEUI AREA : †USAZZHOPOS†USUDA†USUOZ , MAIN UAIEGURI :[@COOO JZMARAJZ: TOIZ "OF"XY.JEDAV.FZZZZI (OVEDALI TOTAL NIMOEDE":2018 "HODATED DATE": "@COOO
1. "CODE . 01 . 0. TECOPY". JE723. 6010274. JE740. SETU: SETU: AREV. (AREV. AREV.
EFX.JERAY.53E3V.30ERY.JERAY.552EX.30NDV.7771V.38N2V.382871 JESCRIPTION.7.99
1"TABLE NAME" 'YU JEBAYI 5353Y 530AYI 3073YI JEBAYI 5353YI 306FY 6307YI 6570YI FODYI 5168Y-5
E5F¥u5E73¥u621027¥u5E74¥uEE09", "TABLE EXPLANATION": "¥u6CE8)¥u0026nbsp;¥u30EB¥u90ED¥u
¥u3092¥u53C2¥u7167¥uFF61¥u0026nbsp;¥u0026nbsp;¥u0026nbsp;¥u0026nbsp;¥u0026nbsp;¥u30F
¥u9E7F¥u5150¥u5CF6¥u770C¥u53CA¥u3073¥u6C96¥u7E04¥u770C¥u306B¥u3064¥u3044¥u3066¥u306F
¥u5404¥u <u>56DE¥u8ABF¥u67FB¥u806E¥u8A</u> BF¥u67FB¥u5730¥u57DF¥u306B¥ <u>u5DEE¥u7570¥u304C¥u304</u>
¥u8003¥uHE16¥u3092¥u53C2¥u/16/¥uHE09¥u30021J¥u0026pbsp;1960¥u5E/4¥u306E¥u4EBA¥u53E3¥
u9UE1¥u5U71¥u53E3¥u5751¥u3U68¥u5U9U¥u961U¥u77UU¥u4EZD¥u6DD5¥u5DDD¥u5EDZ¥u3U6E¥u9593¥
BA) ¥U300F ¥UFDU ¥U30F B¥U30F B¥U30F B¥U30B ¥U307 F¥U308 L¥U306 ¥U304 ¥U308 B¥U304 L¥U30 A F¥U307 / ¥U30 DEDV 20029 × 2002 × 200
0054U30824U34264U307E4U30864U30064U300444U30084U300444UFF01274U0020nDS5518404U3E744U30 x.50721047X,4574X,202EX,202EX,3150X,5726X,3707X,5022X,5725X,6725X,002X,202X,5725X,
4087/57347+02174+03047-037319477-57474-04054-040-0478-55234-040654-64074-55704-20884-03844-038-4
AY, 30 ABY, 30 B3Y, 30 5 EY, 30 5 EY, 30 8 I Y, ECC 1920 Y, 5 E7 4Y, 30 8 EY, 5 1 68Y, 5 ECH, 30 8 EY, 70 5 0 Y 6 79 CY
u7997¥u51FA¥u3057¥u305F¥uFE61]}. "CLASS_INF": { CLASS_OBJ": [{ @id : 'tab @name : '¥u8
[{"@code":"020","@name":"¥u4EBA¥ú53E3","@level":"","@unit":"Yu4EBA"},
["@code":"1050","@name":"¥u4EBA¥u53E3¥u306E¥u6307¥u6570","@level":"","@unit":"Yu5927
["@id";:"area",."@name";:"¥u5730¥u57DF2015","CLASS":["@code":"11000","@name";:"¥u57EC¥u7
{
{ @code :: 1925000000, , @name :: 1925¥u5E74 , , @level :: 1, { , { , @code :: 1930000000, , , @nam
[@code : 19350000000, @name : 1935405274, @level : 1], [@code : 1940000000, @nam
[@code : 1949000000 , @name : 194940574 , @level : 1, [@code : 1947000000 , @nam
["Goode": "1960000000", "ename": "19604.5E74", "elevel": "11, ["Goode": "19650000000", "enam
["Coode": "19700000000", "Chame": "197074/5E74", "Clevel": "1"], "Coode": "19750000000", "Coam
{"@code": "1980000000", "@name": "1980¥u5E74", "@level": "1", {"@code": "1985000000", "@name
{"@code": "19900000000", "@name": "1990¥u5E74", "@level": "1"}, {"@code": "1995000000", "@nam
["@code":"2000000000","@name":"2000¥u5E74","@level":"1"],["@code":"2005000000","@nam
{"@code":"2010000000","@name":"2010¥u5E74","@level":"1"},{"@code":"2015000000","@nam
[] [] [] [] [] [] [] [] [] [] [] [] [] [
; ¥ugh53¥ugh72¥ugb72¥ugb724¥u304U¥u3044¥u3044¥u3082¥u306E_1+]YALUE;: //alacoo//
[i etab ; 020 , earea : 11000 , etime : 192000000 , eunit : ¥u4EBA , \$: 1319533 },
] grap . 020 , garea . 11000 , grime . 1323000000 , gunit : #04Eb8 , 0, 1334401 }, "@thab": 0200" "@exec." 11000 " @time." 19200000000 " @exit*""%utEbA" @time." 1350172"
[_etab020_,_eatea11000_,_etime100000000_,_eunitf44EDA_,_q1400172_1,

XML形式とJSON形式の共通点

どちらもテキスト型のファイル形式であり、キーと値のペアで表現する点と階層型の データを表しやすい点が共通点です。

■ 演習:プログラムからWebAPIを参照する

サンプルプロジェクトをインポートしてWebAPIを参照してみましょう。動かすために はoppIdの指定も必要です。

```
let appId = ""; //ここにappIDを入力して下さい
let url = "https://api.e-stat.go.jp/rest/3.0/app/json/getStatsData?appId=" + appId +
"&lang=J&statsDataId=0003411191&cdArea=11000&metaGetFlg=Y&cntGetFlg=N§ionHeaderFlg=1";
fetch(url)
   .then(function(response) {
       return response.json();
   })
    .then(function(data) {
       console.log(data.GET STATS DATA.STATISTICAL DATA.DATA INF.VALUE);
       let year, populatio;
       let max = data.GET_STATS_DATA.STATISTICAL_DATA.DATA_INF.VALUE.length / 2; //指数情報を取
得しないようにループ回数を半分にする
       for (let i = 0; i < max; i++) {</pre>
           year = data.GET STATS DATA.STATISTICAL DATA.DATA INF.VALUE[i]["@time"] / 1000000; //
年の情報以外を捨てるために100000で除算
           populatio = data.GET STATS DATA.STATISTICAL DATA.DATA INF.VALUE[i].$;
           document.write(year,"年は",populatio,"人です","");
   });
```

キー・バリュー形式のデータの処理・蓄積

WebAPIによるデータの取得

プログラミング言語で複数の値を格納する代表的な仕組みとして配列というものが存 在します。また、配列には大きく分けて2種類あり、「配列(Array)」と「連想配列 (Associative Array)」が存在します。

連想配列は言語によって呼び方が様々です、この項目のタイトルにある「キー・バ リュー形式」というのは、連想配列のことを指します。また、JavaScriptでは連想配列 のことを「オブジェクト」と呼んだりもします。

研修内容

サンプルプロジェクトを取り込んでアドレス帳を作成します。

| 配列から学ぶキー・バリュー形式の学習|

プログラミングで好きな和菓子一覧のような値を用意したい場合には、配列 (Array)を使います。

sweets = ["もなか", "だんご", "すあま", "ようかん"];

単純に要素を列挙しただけの配列を特に|次元配列あるいは線形配列と呼びます。|次元 配列のキー(添え字)は0から順番に自動で数字が振られます。例えばもなかを指定する場 合は以下のようになります。

console.log(sweets[0]);

■ 繰り返し文による配列の参照

複数の要素を全て参照したい場合にはfor文を使って以下のように記述します。

```
for (let i = 0; i < sweets.length; i++) {
    console.log(sweets[i]);
}</pre>
```

■ 2次元配列・3次元配列・多次元配列

1次元配列があると言うことは、2次元配列や3次元配列なども存在します。2次 元以上の配列を多次元配列とも呼びます。

sweets = [["もなか", "だんご", "すあま", "ようかん"], ["ビスケット", "クッキー", "パンケーキ"],];

仮にこの2次元配列からビスケットを参照したい場合はsweets[1][0]を参照します。

console.log(sweets[1][0]);

■ 連想配列

配列のメリットの一つはキー(添え字)が自動的に採番されるので宣言が楽なこ とです。

しかし、上記の例では0が和菓子で1が洋菓子であることをソースコードから確 認することができません。そこでキーに文字列を使える連想配列を活用します。

sweets = { japanese:["もなか", "だんご", "すあま", "ようかん"] western:["ビスケット", "クッキ─", "パンケ─キ"] }

仮にこの配列からビスケットを参照したい場合はsweets.western[0] あるいは sweets["western"][0]と参照します。

■ 連想配列でアドレス帳を作る

まずは一人分のアドレスを連想配列で表現してみます。

```
member = {
    "name" : "アシアル太郎",
    "homephone" : "050-****-****",
    "mobileemail" : "asial@example.com",
};
```

キーが数字の連番では無いためfor文は使えません。連想配列に対応した繰り返 し構文を使う必要があります。

```
for (let key in member) {
   console.log(member[key]);
}
```

for...inという繰り返し構文では、連想配列のキーの数だけ繰り返しを実行します。また、繰り返しの度にキーの文字列を任意の変数に代入させることができます。上記の例ではkeyという変数に対してname, homephone, mobileemailの文字列をそれぞれ代入しています。

■ 複数人のアドレスを格納する

配列と組み合わせてmembers配列を作成します。

■ アドレスの追加

あとからmembers配列に要素を追加したい場合は以下のように記述します。



■ アドレス帳の参照

全員のアドレス帳を単純に確認したいだけの時は以下のように記述します。

console.log(members);

```
データを加工しながらユーザーの画面に出したいときなどには、繰り返し構文
を使って表示する必要があります。
```

```
for (let i = 0; i < members.length; i++) {
  for (let key in members[i]) {
    console.log(members[i][key]);
  }
}</pre>
```

あるいはfor...of構文を併用して以下のように記述します。

```
for (let member of members) {
   for (let key in member) {
      console.log(member[key]);
   }
```

■ キーバリュー型のデータベース

連想配列のような仕組みで値を格納できるデータベースとして「キーバ リュー型データベース」が存在します。リレーショナルデータベースに比べて、 事前にテーブル設計を行わなくても気軽に利用できることや速度上の優位性・ 分散処理上の優位性があります。

身近なキーバリュー型データベースの活用例としてはブラウザに情報を記録 できる「Web Storage」があります。Web StorageはHTML5の標準技術のため、特 別な準備は不用で簡単に利用できます。全員のアドレス帳を単純に確認したい だけの時は以下のように記述します。

■ NoSQL型のデータベース

リレーショナルデータベース以外のデータベースをNoSQL型と呼びます。キー バリュー型のデータベースもNoSQL型の一つです。リレーショナルデータベース は信頼性を保つための仕組みが非常に優れています(ACID特性)。しかし、大量 のセンサーデータを多少欠損しても良いので片っ端から記録するとか、膨大な 会員を抱えるSNSの投稿を記録したりユーザー毎のタイムラインを表示するのは 得意ではありません。

1億人お小遣い帳を記録するならリレーショナルデータベースの機能だけで 比較的簡単に実現できますが、1億人のSNS投稿を支えようと考えた場合は NoSQL型のデータベースを元にデータの扱い方を設計する必要があります。なお、 最近のリレーショナルデータベース製品はNoSQL型のデータをサポートしている 場合があります。