

SPARQLでオープンデータ活用！

上田 洋

特定非営利活動法人リンクト・オープン・データ・イニシアティブ

LODチャレンジJapan実行委員会

<http://uedayou.net/>

第4回 Code for Wakayama オープンデータをクラウドで活用しよう！

2015.5.30

オープンデータとの関わり

- 特定非営利活動法人リンクト・オープン・データ・イニシアティブ

- 2014年より理事



- Linked Open Data チャレンジ Japan

- 2014年度より実行委員
 - 関西支部 所属



- Linked Open Dataハッカソン関西

- 運営メンバー

- オープンデータを利用したWebアプリを公開中

- <http://uedayou.net/> とGitHubで公開しています

告知1

Linked Open Data チャレンジ Japan

- 今年度も開催が決定しました！

2015年10月募集開始、2016年1月上旬締め切り
授賞式は2016年3月19日(土)

- キックオフシンポジウムを開催します！

日時: 2015年9月26日(土) 13:00-18:00

場所: 慶應義塾大学 三田キャンパス 南校舎ホール

- 公式サイト:<http://lod.sfc.keio.ac.jp/challenge2014/>
- Twitter:@LodJapan
- Facebook:LOD.challenge.Japan

受賞者、スポンサー、会場の皆様で盛り上がった授賞式！

告知2

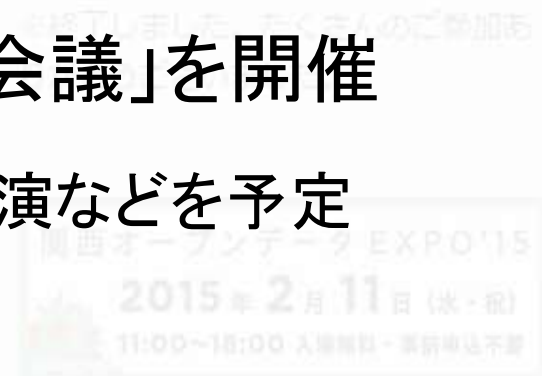
Linked Open Data ハッカソン関西

- 関西を中心にLinked Open Data(LOD)の技術普及を目的に開催しているイベントを企画・運営
- 「自治体オープンデータ推進協議会関西会議」を開催
 - IPA共通語彙基盤とGitHubの担当者による講演などを予定
 - オブザーバ参加者募集中！

<http://peatix.com/event/93254/>

- 公式サイト

<http://wp.lodosaka.jp/>



告知3

Linked Open Dataとオントロジー勉強会

- 大阪大学 古崎晃司准教授が主催する
Linked Open Data、オントロジー、セマンティックWeb
に関する勉強会
 - 大阪を中心に過去5回開催
 - 今年は、
 - LODの作り方
 - LODの公開の仕方
 - LODの使い方(アプリの開発方法)
 - 公式FBページで近日中に告知予定
- をテーマにハンズオン形式の勉強会を企画中

<https://www.facebook.com/LodOnt>

Lined Open Data(LOD)とは？

Linked Open Dataとは？

- **Linked Data**形式で公開される**オープンデータ**
- 提唱者：WWW発明者 **ティム・バーナーズ・リー氏**
- Linked Data4原則
 1. あらゆる「もの」の名前にURI(Uniform Resource Identifier)をつけましょう
 2. HTTPプロトコルのURI(<http://hoge.net/hoge>)で、「もの」についてのデータにアクセスできるようにしましょう
 3. URIでアクセスしたデータは標準的なフォーマット(RDFを推奨)で提供しましょう
 4. 他のデータを見つけやすくするために他のデータとURIでリンクしましょう

オープンデータの定義

Open Definition

利用できる、そしてアクセスできる

データ全体を丸ごと使えないといけなし、再作成に必要以上のコストがかかってはいけない。望ましいのは、インターネット経由でダウンロードできるようにすることだ。また、データは使いやすく変更可能な形式で存在しなければならない。

再利用と再配布ができる

データを提供するにあたって、再利用や再配布を許可しなければならない。また、他のデータセットと組み合わせて使うことも許可しなければならない。

誰でも使える

誰もが利用、再利用、再配布をできなければならない。データの使い道、人種、所属団体などによる差別をしてはいけない。たとえば「非営利目的での利用に限る」などという制限をすると商用での利用を制限してしまうし「教育目的での利用に限る」などの制限も許されない。

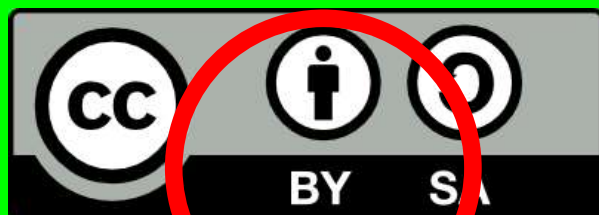
<http://opendatahandbook.org/guide/ja/what-is-open-data/>

これらを許可するライセンス(オープンライセンス)を持つデータ

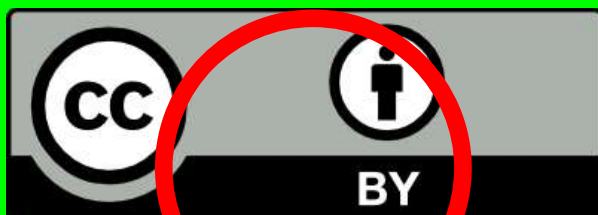


オープンデータ

クリエイティブコモンズライセンス でのオープンライセンス



CC BY-SA
表示-継承



CC BY
表示



CC0
権利放棄



CC BY-NC-ND
表示-非営利-改変禁止



CC BY-NC
表示-非営利



CC BY-ND
表示-改変禁止



全ての権利を主張



CC BY-NC-SA
表示-非営利-継承

和歌山県オープンデータは？

オープンデータに関する取り組みについて

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。

現在、日本政府においても「電子行政オープンデータ戦略」（平成24年7月 IT戦略本部決定）等に基づき、オープンデータの取組を推進しています。

今般、和歌山県では、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※1」にアカウント「wakayama pref org」を開設しました。

GitHubアカウント「wakayama pref org」(<https://github.com/wakayama-pref-org>)

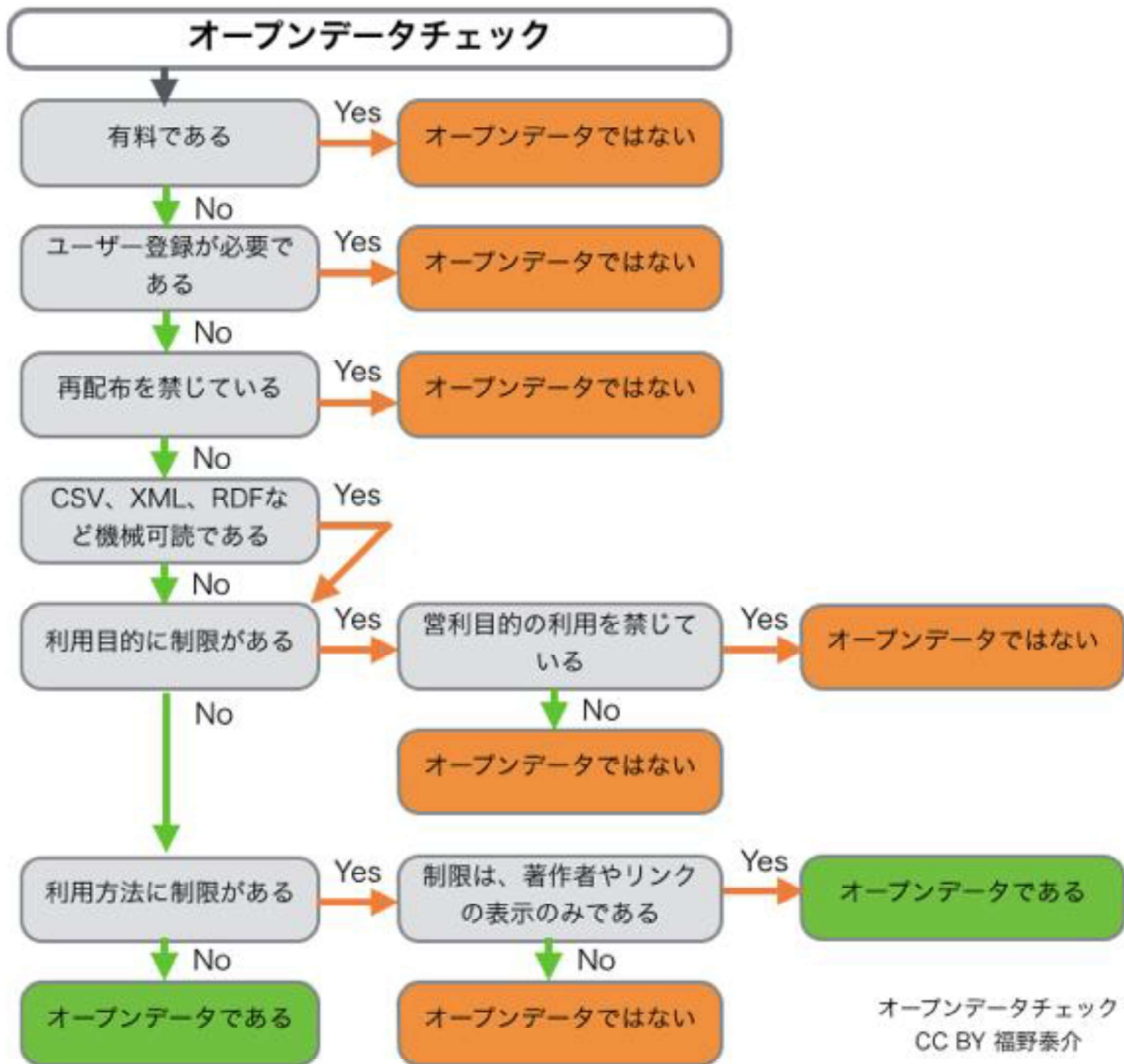
オープンデータの利用にあたって



(ライセンス)

掲載している各種データについては、**クリエイティブ・コモンズ・ライセンス表示2.1のもとでライセンスされています。**

<http://www.pref.wakayama.lg.jp/prefg/020400/opendata/index.html>



オープンデータチェック
CC BY 福野泰介

5つ星オープンデータ

星が多いほど良いオープンデータ

外部のデータ
とリンクをつける



ソフトウェアに
依存しない
ファイル形式
で公開



オープン
ライセンス
でデータ公開



構造化された
データを公開



他のデータと
リンクするため
にURI使い、
アクセス可能で
あること



Linked Open Data (LOD)



とても良い
オープンデータ



日本のオープンデータ都市数:151

全体図/ALL

都市名	ライセンス	開始日	星	内容
奈良県奈良市	CC BY	2014-08-11	5	避難所、人口、国勢調査、農林、水産、事業所、工業、商業、物価、消費生活、労働
長野県須坂市	CC BY	2014-05-01	2	統計、財政
神奈川県茅ヶ崎市	CC BY	2014-11-26	3	人口、防災、選挙
愛知県長久手市	CC BY	2014-10-21	5	避難所
大阪府	CC BY	2015-02-02	2	くらし、福祉、健康、観光、エネルギー、文化、都市計画、防災、給与、財産
三重県	CC BY	2015-01-30	3	防災・災害、健康・保健・医療・福祉、産業・商工業、文化・教育・観光、環境、法令・土地、統計、公共施設情報
静岡県浜松市	CC BY	2014-12-10	3	ごみ
大阪府大阪市生野区	CC BY	2015-02-01	2	広報誌
宮城県大崎市	CC BY	2013-12-19	3	避難所
宮城県仙台市	CC BY	2015-02-05	3	人口、避難所、観光、病院、AED
千葉県浦安市	CC BY	2015-02-18	3	地図、公共施設、防災、自治会、バス、都市計画、ギョーリング柱状図
和歌山県	CC BY	2015-02-20	3	公共トイレ、統計、避難所、道路規制、公共施設
新潟県見附市	CC BY	2015-02-23	4	人口、公共施設、教育・子育て、統計
滋賀県草津市	CC BY	2015-03-05	3	バス
北海道八雲町	CC BY	2015-03-06	4	避難所、医療機関、学校

<http://fukuno.jig.jp/2013/opendatamap>

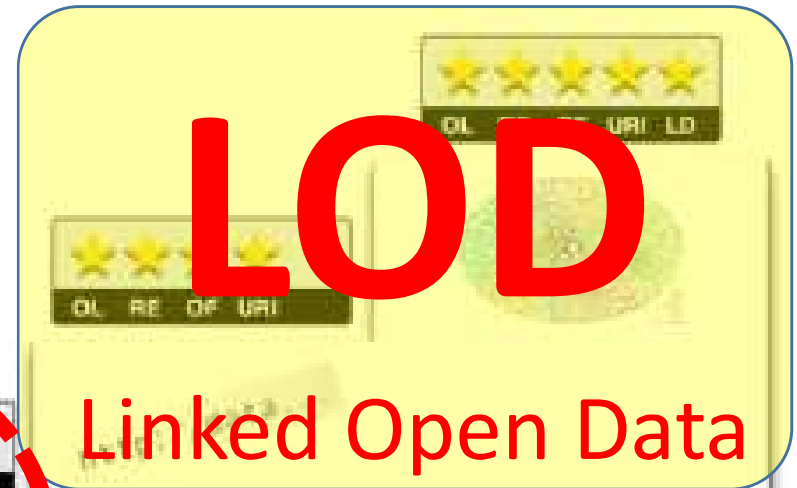
5つ星オープンデータ

星が多いほど良いオープンデータ

外部のデータ
とリンクをつける



ソフトウェアに
依存しない
ファイル形式
で公開



他のデータと
リンクするため
にURI使い、
アクセス可能で
あること



オープン
ライセンス
でデータ公開



構造化された
データを公開



リンクして何が嬉しい？



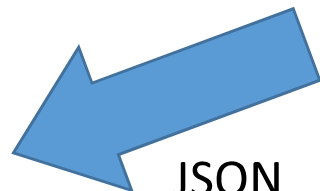
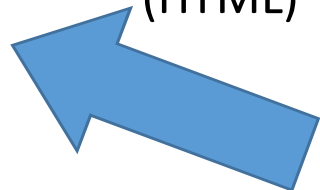
個々のデータにURI(URL)で直接アクセスできれば...

- 例: レストランのメニュー



アプリ

Webページ
(HTML)



JSON
とかXML

<http://cafe-a.jp/>

<http://cafe-a.jp/menu>

メニュー	価格
Aランチ	600円
Bランチ	700円
Cランチ	800円
...	

<http://cafe-a.jp/lunch-a>

料理名	Aランチ
説明	Aランチは...
材料	〇〇産牛肉 〇〇産米

<http://cafe-a.jp/lunch-b>

料理名	Bランチ
説明	Bランチは...
材料	〇〇産豚肉 〇〇産キャベツ

常に最新のデータを提供・取得できる

メリット: 今までのWebサイトの延長線上で提供できる

URLを指定すればそのデータにアクセスできる(リンクが出来る)
データの改ざんがすぐにばれる(データの検証性・信頼性)



URI(URL)で他のデータに リンクすると...

<http://cafe-a.jp/>

<http://cafe-a.jp/lunch-a>

料理名	Aランチ
説明	Aランチは...
材料	〇〇産牛肉 〇〇産米

<http://cafe-a.jp/lunch-b>

料理名	Bランチ
説明	Bランチは...
材料	〇〇産豚肉 〇〇産キャベツ

<http://cafe-a.jp/menu>

メニュー	価格
Aランチ	600円
Bランチ	700円
Cランチ	800円
...	

<http://▲▲□□.jp/beef>

料理名	〇〇産牛肉
説明	この牛肉は...
生産者	▲▲□□
連絡先	

<http://◆◆△△.jp/rice>

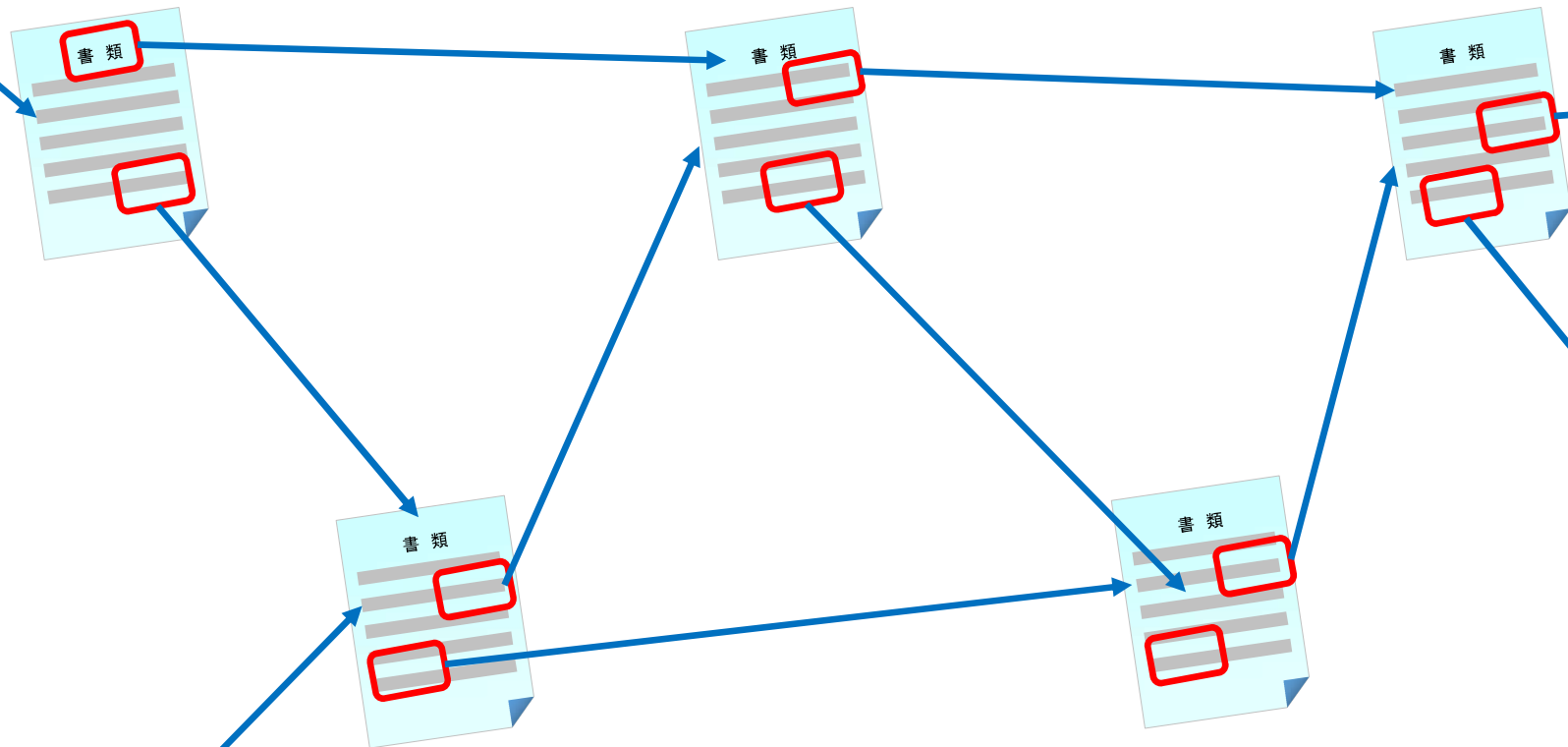
料理名	〇〇産米
説明	〇〇は...
生産者	◆◆△△
連絡先	

<http://●▲×◆.jp/pork>

料理名	〇〇産豚肉
説明	〇〇の豚は...
生産者	●▲×◆
連絡先	

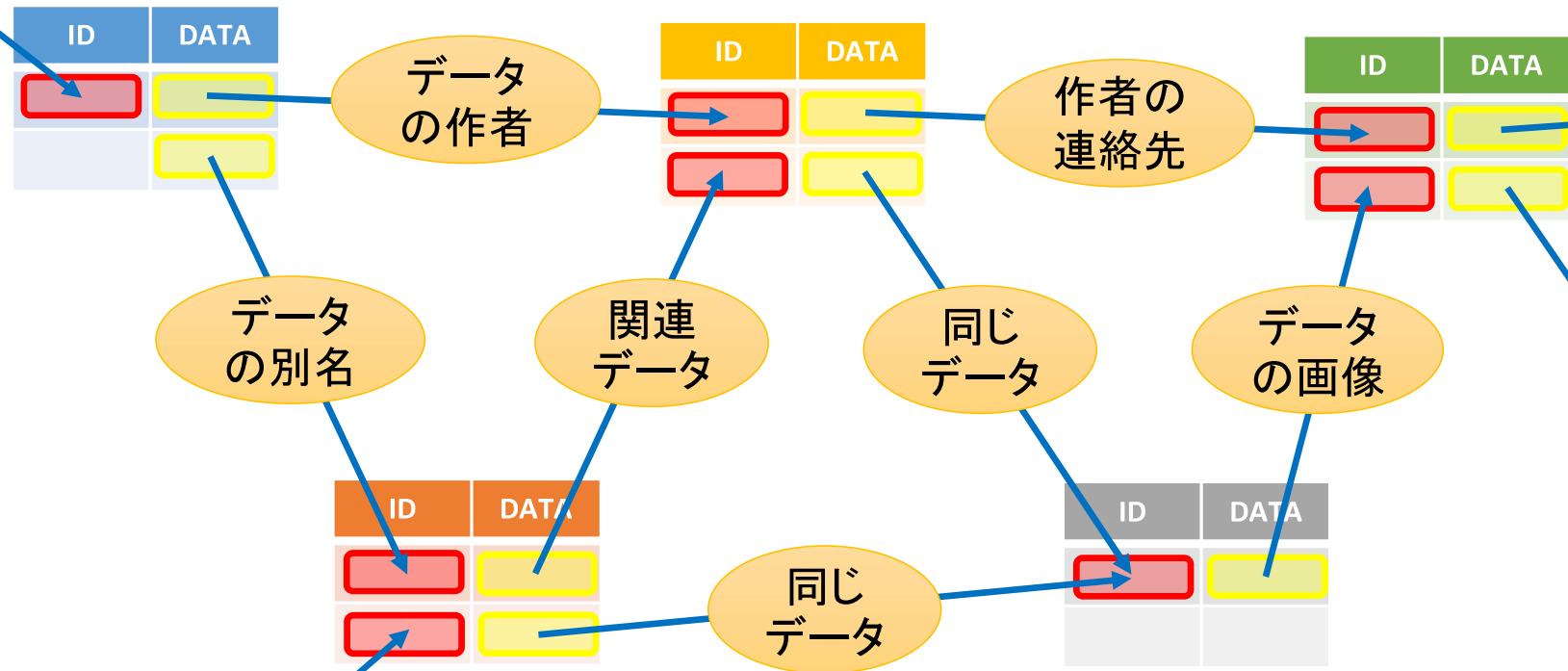
メリット: 分散してデータを管理できる

World Wide Web(文書のWeb)



メリット:リンクによりたくさんのページ(情報)を簡単に発見できる

データのWeb=Linked Open Data(LOD)



メリット: リンクによりたくさんのデータを簡単に発見できる
データを分散して管理

共通の標準フォーマットでデータが簡単に取得できる

Linked Data

- ティム・バーナーズ＝リーのLinked Data4原則

1. あらゆる「もの」の名前にURI(Uniform Resource Identifier)をつけましょう
2. HTTPプロトコルのURI(<http://hoge.net/hoge>)で、「もの」についてのデータにアクセスできるようにしましょう
3. URIでアクセスしたデータは標準的なフォーマット(RDFを推奨)で提供しましょう
4. 他のデータを見つけやすくするために他のデータとURIでリンクしましょう

<http://www.w3.org/DesignIssues/LinkedData.html>

RDFとは

(Resource Description Framework)

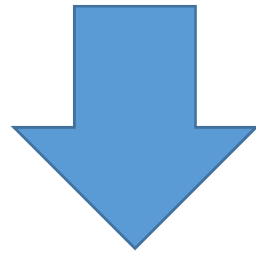


- データを「主語」「述語」「目的語」を1つのセット (トリプル、三つ組み) として記述

例えば...



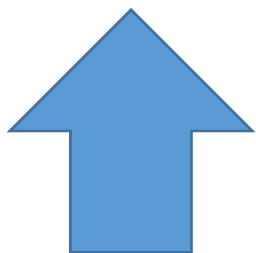
RDF用データベース 検索クエリ



SPARQL

(SPARQL Protocol and RDF Query Language)

Linked Open Dataを
実現するための基盤技術



SPARQL

SPARQL

(SPARQL Protocol and RDF Query Language)

SPARQL

いろんなデータベース
で使えます

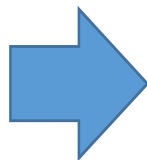
- RDF(LOD)用の**汎用**クエリ言語
- RDF(LOD)用のデータベースからトリプル(3つ組)
データを検索して、**表形式**でデータが取得できます

扱いやすい！

```
PREFIX rdfs:
<http://www.w3.org/2000/01/rdf-schema#>

SELECT * WHERE {
    ?uri rdfs:label ?label .
}

LIMIT 10
```



?uri	?label
http://ja.dbpedia.org/resource/岩手県	"岩手県"
http://ja.dbpedia.org/resource/石川県	"石川県"
http://ja.dbpedia.org/resource/愛媛県	"愛媛県"
http://ja.dbpedia.org/resource/岡山県	"岡山県"

SPARQLで検索できるデータが増えています

データ種別	Web API
Wikipedia	DBPedia Wikipediaオントロジー
行政データ	データシティ鯖江 都道府県・市区町村コード情報 Open Data METI
イベントデータ	ヨコハマ・アート・LOD
博物館データ	Europeana LODAC Museum The British Museum
図書館データ	The British National Bibliography Web NDL Authorities
地理データ	Linked Geo Data LODAC Location
気象データ	気象庁XML用API
生物種データ	LODAC Species

この他にも多くのオープンデータがSPARQLで検索できます

LOD推奨形式RDFとは (Resource Description Framework)



- 「主語」「述語」「目的語」を1つのセット(トリプル、三つ組み)として記述

例えば...



トリプルデータを表形式に変換

主語	述語	目的語
dbpedia: 東京都	rdfs:label	"東京都"
dbpedia: 東京都	rdfs:comment	"東京都(とうきょうと)は、日本の都道府県の一つである。"
dbpedia: 大阪府	rdfs:label	"大阪府"
dbpedia: 大阪府	rdfs:comment	"大阪府(おおさかふ)は、近畿地方(関西地方)に属する日本の都道府県の一つ。"
dbpedia: 京都府	rdfs:label	"京都府"
dbpedia: 京都府	rdfs:comment	"京都府(きょうとふ)は、日本国・近畿地方の都道府県。"



SPARQLなら簡単にできます！

HTML・CSV・XML・JSONなど さまざまな形式で出力可能！

- formatパラメータを変更するだけ

JSON

<http://db.lodc.jp/sparql?query=...&format=json>

XML

<http://db.lodc.jp/sparql?query=...&format=xml>

CSV

<http://db.lodc.jp/sparql?query=...&format=csv>

“format”はSPARQLエンドポイントによって異なる場合があります



ORDER+BY+%3Fs%0D%0ALIMIT+100%0D%0A&format=json



```
{ "head": { "link": [], "vars": [ "s", "p", "o" ] },
  "results": { "distinct": false, "ordered": true, "bindings": [
    { "s": { "type": "uri", "value": "http://creativecommons.org/ns#license" },
      "p": { "type": "uri", "value": "http://creativecommons.org/licenses/by/3.0/deed.ja" },
      "o": { "type": "uri", "value": "http://creativecommons.org/ns#attributionName" },
      "literal": { "xml:lang": "ja", "value": "\u53E4\u5D0E\u6643\u53F8" } },
    { "s": { "type": "uri", "value": "http://linkdata.org/resource/rdf1s933i#1" },
      "p": { "type": "uri", "value": "http://www.w3.org/1999/02/22-rdf-syntax-ns#type" },
      "o": { "type": "uri", "value": "http://lodosaka.hozo.jp/class/\u65BD\u8A2D\u60C5\u5831" } },
    { "s": { "type": "uri", "value": "http://linkdata.org/resource/rdf1s933i#1" },
      "p": { "type": "uri", "value": "http://www.w3.org/2000/01/rdf-schema#label" },
      "o": { "type": "literal", "xml:lang": "ja", "value": "1" } },
    { "s": { "type": "uri", "value": "http://linkdata.org/resource/rdf1s933i#1" },
      "p": { "type": "uri", "value": "http://www.w3.org/2000/01/rdf-schema#label" },
      "o": { "type": "literal", "xml:lang": "ja", "value": "\u5927\u962A\u5E02\u7ACB\u962A\u5357\u4DD\u80B2\u6240" } },
    { "s": { "type": "uri", "value": "http://linkdata.org/resource/rdf1s933i#1" },
      "p": { "type": "uri", "value": "http://xmlns.com/foaf/0.1/homepage" },
      "o": { "type": "uri", "value": "http://www.city.osaka.lg.jp/kodomo/page/0000002800.html" } },
    { "s": { "type": "uri", "value": "http://linkdata.org/resource/rdf1s933i#1" },
      "p": { "type": "uri", "value": "http://www.w3.org/2003/01/geo/wgs84_pos#lat" },
      "o": { "type": "typed-literal", "datatype": "http://www.w3.org/2001/XMLSchema#float", "value": "34.6285" } },
    { "s": { "type": "uri", "value": "http://linkdata.org/resource/rdf1s933i#1" },
      "p": { "type": "uri", "value": "http://www.w3.org/2003/01/geo/wgs84_pos#long" },
      "o": { "type": "typed-literal", "datatype": "http://www.w3.org/2001/XMLSchema#float", "value": "135.515" } },
    { "s": { "type": "uri", "value": "http://linkdata.org/resource/rdf1s933i#1" },
      "p": { "type": "uri", "value": "http://schema.org/address" },
      "o": { "type": "literal", "xml:lang": "ja", "value": "\u002F\u5000\u01CF\u52A\u002A\u5257\u752A\u2012\u2012" } } ]
}
```

なにが作れる？

Civic Tech Osaka ～大阪から考えるCivicTech～



Civic Tech Osaka
大阪から考えるCivicTech

アプリコンテスト開催決定！
応募締切りは平成27年2月22日17時まで！

<公開プレゼン・表彰式イベントのご案内>
平成27年3月21日（土）開催！！
会場：富士通エフサス Osaka knowledge wharf
(大阪府大阪市中心区城見 2-2-22 マルイト OBP ビル 10 階)

詳細はこちら！

<http://civictech-osaka.jp/index.html>

コンテスト結果

(アプリ・Webサービス部門)

- 優勝(グランプリ)

- PUSH大阪

※ハッカソン(Civic Hack OSAKA 2014)内で開発

- 準優勝(優秀賞)

- 大阪もよりナビ

※ハッカソン(インターナショナルオープンデータデー)内で開発

- 特別賞(オープンデータ活用賞)

- 大阪市 警察署 × 犯罪発生

※ 4～5時間で完成

3アプリ全てSPARQLを活用しています！

PUSH大阪/PUSH広報

- 自治体が発信するRSSを収集し、**SPARQL**を使って動的に検索、アプリではプッシュ機能で受動的に知りたい情報を受け取れる



<http://push.jp.net/>

大阪もよりナビ

- 大阪の最寄駅、駐輪場、トイレなどをSPARQLで検索し、現在地からナビゲーションするアプリ



<http://moyori.lodosaka.jp/>

大阪市 警察署 × 犯罪発生

- 大阪市の警察施設と犯罪発生地点をSPARQLを使い取得し、地図上で重ね合わせることができるアプリ



<http://uedayou.net/osakacrimemap/>

ハンズオン SPARQLの書き方と 簡単アプリ作成

事前にブラウザに

<http://uedayou.net/sparql-mashup/>

を開いておいてください

URI(IRI)とリテラル

- URI(IRI) : <http://ja.dbpedia.org/resource/東京都>
 - 「<」「>」で挟まれた文字列
 - 人・もの・出来事などを指し示すID
 - 指定したURIを同じURIを含むデータが検索される
- リテラル : “東京”, “100”, “2014-12-07” ...
 - 「”」で挟まれた文字列
 - データそのもの(string, integer, float他)
 - 指定した文字列と同じ文字列を含むデータが検索される
 - 目的語以外は指定できない

RDFのデータベース

主語	述語	目的語
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#label>	"東京都"
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#comment>	"東京都(とうきょうと)は、日本の都道府県の一つである。"
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/山梨県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/千葉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/埼玉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/神奈川県>
<http://ja.dbpedia.org/resource/山梨県>	<http://www.w3.org/2000/01/rdf-schema#label>	"山梨県"
<http://ja.dbpedia.org/resource/千葉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"千葉県"
<http://ja.dbpedia.org/resource/埼玉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"埼玉県"
<http://ja.dbpedia.org/resource/神奈川県>	<http://www.w3.org/2000/01/rdf-schema#label>	"神奈川県"

URIとリテラル

主語	述語	目的語
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#label>	"東京都"
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#comment>	"東京都(とうきょうと)は、日本の都道府県の一つである。"
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/山梨県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/千葉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/埼玉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/神奈川県>
<http://ja.dbpedia.org/resource/山梨県>	<http://www.w3.org/2000/01/rdf-schema#label>	"山梨県"
<http://ja.dbpedia.org/resource/千葉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"千葉県"
<http://ja.dbpedia.org/resource/埼玉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"埼玉県"
<http://ja.dbpedia.org/resource/神奈川県>	<http://www.w3.org/2000/01/rdf-schema#label>	"神奈川県"

(1) シンプルなSPARQLクエリ

```
SELECT ?s ?p ?o WHERE  
{  
    ?s ?p ?o.  
}
```

- 全てのトリプルデータを検索

SELECT

```
SELECT ?s ?p ?o WHERE
```

```
{
```

```
  ?s ?p ?o.
```

```
}
```

- SELECT の後ろに続く代数 (?で始まる文字列)に格納されたデータを表形式で取得
- 代数名は必ず WHERE文の中で指定したものを記述すること
 - 例では ?s、?p、?o

SELECT

```
SELECT * WHERE  
{  
  ?s ?p ?o.  
}
```

=

```
SELECT ?s ?p ?o WHERE  
{  
  ?s ?p ?o.  
}
```

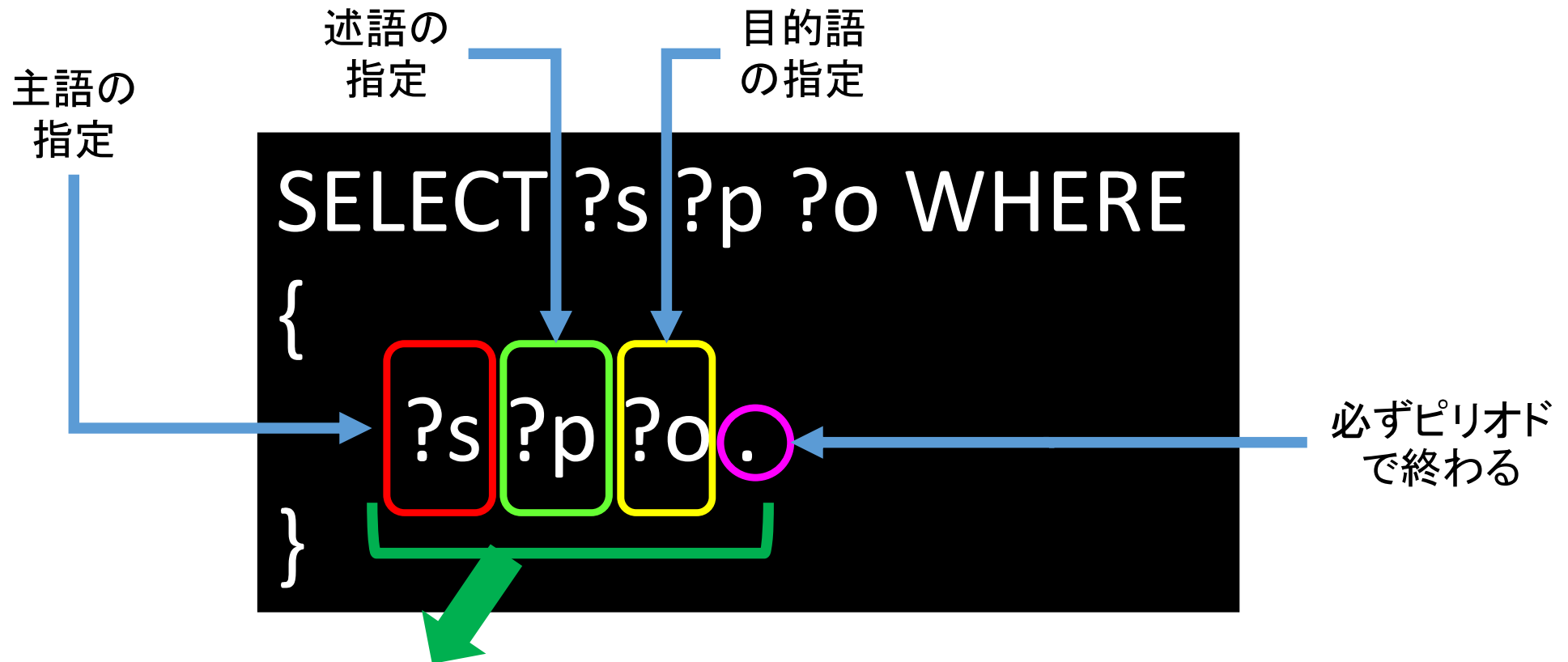
- アスタリスク「*」を指定すると、WHERE文の中の全ての代数を指定したことになる
 - ?s、?p、?o が抽出される

WHERE

```
SELECT ?s ?p ?o WHERE  
{  
  ?s ?p ?o .  
}
```

- WHERE { } 内に検索したいトリプルパターンを書く

基本構造



- トリプルパターンを指定するために、3つの代数またはURI、リテラル(目的語のみ)を1セットで書く
 - 代数: ?で始まる文字列、任意のURIまたはリテラルが入る
- 1セットの終わりには必ずピリオド「.」をつける

RDFのデータベース

主語	述語	目的語
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#label>	"東京都"
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#comment>	"東京都(とうきょうと)は、日本の都道府県の一つである。"
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/山梨県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/千葉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/埼玉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/神奈川県>
<http://ja.dbpedia.org/resource/山梨県>	<http://www.w3.org/2000/01/rdf-schema#label>	"山梨県"
<http://ja.dbpedia.org/resource/千葉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"千葉県"
<http://ja.dbpedia.org/resource/埼玉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"埼玉県"
<http://ja.dbpedia.org/resource/神奈川県>	<http://www.w3.org/2000/01/rdf-schema#label>	"神奈川県"

代数とトリプルパターン

- 代数: ?s , ?p , ?o, ?label, ?comment ...
 - 「?」で始まる文字列
 - パターンに当てはまるデータ全てが格納される
 - 検索式内の同一代数は同じデータが入るという意味にトリプルパターン



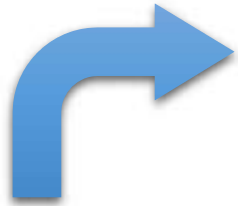
主語	述語	目的語
dbpedia: 東京都	rdfs:label	“東京都”
dbpedia: 東京都	rdfs:comment	“東京都(とうきょうと)は、日本の都道府県の一つである。”
dbpedia: 大阪府	rdfs:label	“大阪府”
dbpedia: 大阪府	rdfs:comment	“大阪府(おおさかふ)は、近畿地方(関西地方)に属する日本の都道府県の一つ。”
dbpedia: 京都府	rdfs:label	“京都府”



id	label
dbpedia: 東京都	“東京都”
dbpedia: 大阪府	“大阪府”
dbpedia: 京都府	“京都府”

rdfs:label は <<http://www.w3.org/2000/01/rdf-schema#label>>
というURIで目的語がラベルであることを意味する

トリプルパターン



?id **rdfs:comment** **?comment.**



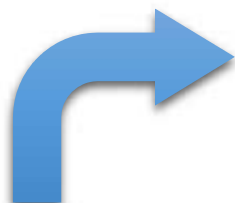
主語	述語	目的語
dbpedia: 東京都	rdfs:label	“東京都”
dbpedia: 東京都	rdfs:comment	“東京都(とうきょうと)は、日本の都道府県の一つである。”
dbpedia: 大阪府	rdfs:label	“大阪府”
dbpedia: 大阪府	rdfs:comment	“大阪府(おおさかふ)は、近畿地方(関西地方)に属する日本の都道府県の一つ。”
dbpedia: 京都府	rdfs:label	“京都府”

id	comment
dbpedia: 東京都	“東京都(とうきょうと)は、日本の都道府県の一つである。”
dbpedia: 大阪府	“大阪府(おおさかふ)は、近畿地方(関西地方)に属する日本の都道府県の一つ。”

rdfs:comment は <<http://www.w3.org/2000/01/rdf-schema#comment>>
というURIで目的語がコメントであることを意味する

同じ名前の代数(?id)を指定すると
論理積に

トリプルパターン



主語	述語	目的語
dbpedia:東京都	rdfs:label	“東京都”
dbpedia:東京都	rdfs:comment	“東京都(とうきょうと)は、日本の都道府県の一つである。”
dbpedia:大阪府	rdfs:label	“大阪府”
dbpedia:大阪府	rdfs:comment	“大阪府(おおさかふ)は、近畿地方(関西地方)に属する日本の都道府県の一つ。”
dbpedia:京都府	rdfs:label	“京都府”



id	label	comment
dbpedia:東京都	“東京都”	
dbpedia:東京都		“東京都(とうきょうと)は、日本の都道府県の一つである。”
dbpedia:大阪府	“大阪府”	
dbpedia:大阪府		“大阪府(おおさかふ)は、近畿地方(関西地方)に属する日本の都道府県の一つ。”



id	label	comment
dbpedia:東京都	“東京都”	“東京都(とうきょうと)は、日本の都道府県の一つである。”
dbpedia:大阪府	“大阪府”	“大阪府(おおさかふ)は、近畿地方(関西地方)に属する日本の都道府県の一つ。”

URIが同一のデータは同じ行に
まとまって出力されます

(1) シンプルなSPARQLクエリ

```
SELECT ?s ?p ?o WHERE  
{  
    ?s ?p ?o.  
}
```

- 全てのトリプルデータを検索

検索対象

?s	?p	?o
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#label>	"東京都"
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#comment>	“東京都(とうきょうと)は、日本の都道府県の一つである。”
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/山梨県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/千葉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/埼玉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/神奈川県>
<http://ja.dbpedia.org/resource/山梨県>	<http://www.w3.org/2000/01/rdf-schema#label>	“山梨県”
<http://ja.dbpedia.org/resource/千葉県>	<http://www.w3.org/2000/01/rdf-schema#label>	“千葉県”
<http://ja.dbpedia.org/resource/埼玉県>	<http://www.w3.org/2000/01/rdf-schema#label>	“埼玉県”
<http://ja.dbpedia.org/resource/神奈川県>	<http://www.w3.org/2000/01/rdf-schema#label>	“神奈川県”

東京都の隣接県エンドポイント

```
SELECT * WHERE
{
  ?s ?p ?o
}
```

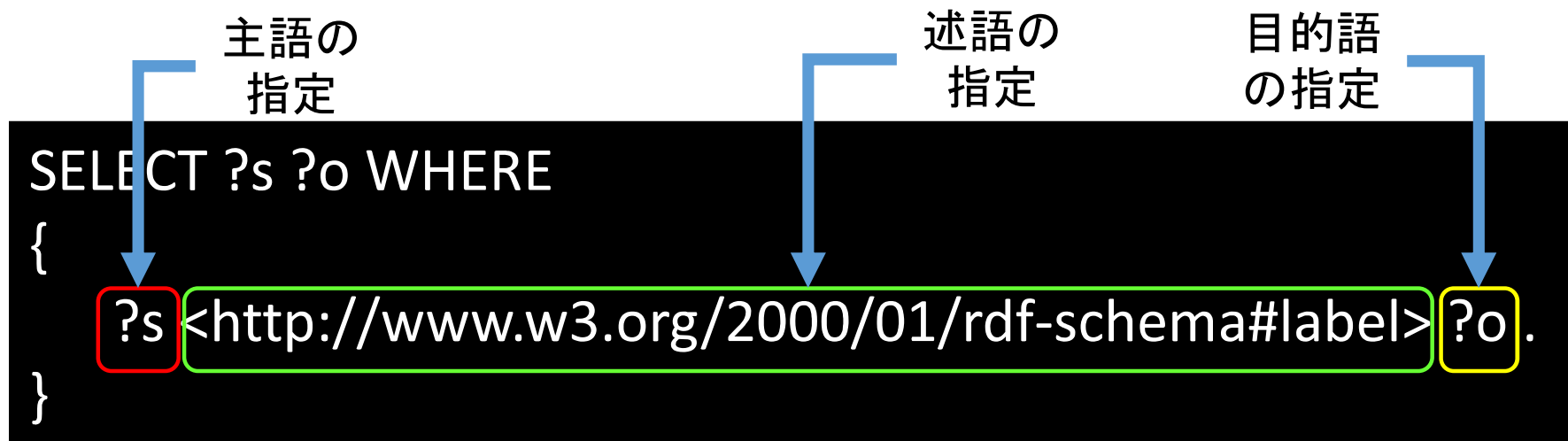
Load

s	p	o
http://ja.dbpedia.org/resource/埼玉県	http://www.w3.org/2000/01/rdf-schema#label	埼玉県
http://ja.dbpedia.org/resource/千葉県	http://www.w3.org/2000/01/rdf-schema#label	千葉県
http://ja.dbpedia.org/resource/神奈川県	http://www.w3.org/2000/01/rdf-schema#label	神奈川県
http://ja.dbpedia.org/resource/東京都	http://www.w3.org/2000/01/rdf-schema#label	東京都
http://ja.dbpedia.org/resource/東京都	http://www.w3.org/2000/01/rdf-schema#comment	東京都(とうきょうと)は、日本の都道府県の一つである。
http://ja.dbpedia.org/resource/東京都	http://ja.dbpedia.org/property/隣接都道府県	http://ja.dbpedia.org/resource/山梨県
http://ja.dbpedia.org/resource/東京都	http://ja.dbpedia.org/property/隣接都道府県	http://ja.dbpedia.org/resource/千葉県
http://ja.dbpedia.org/resource/東京都	http://ja.dbpedia.org/property/隣接都道府県	http://ja.dbpedia.org/resource/埼玉県
http://ja.dbpedia.org/resource/東京都	http://ja.dbpedia.org/property/隣接都道府県	http://ja.dbpedia.org/resource/神奈川県
http://ja.dbpedia.org/resource/山梨県	http://www.w3.org/2000/01/rdf-schema#label	山梨県

<http://uedayou.net/sparql-mashup/test-endpoint/>

(2) 述語が

<http://www.w3.org/2000/01/rdf-schema#label>
であるトリプルを取得



- <http://www.w3.org/2000/01/rdf-schema#label> はデータのラベル(名前)を示す場合によく利用されています。
- 省略形は「rdfs:label」

検索対象

?s	?p	?o
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#label>	"東京都"
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#comment>	"東京都(とうきょうと)は、日本の都道府県の一つである。"
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/山梨県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/千葉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/埼玉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/神奈川県>
<http://ja.dbpedia.org/resource/山梨県>	<http://www.w3.org/2000/01/rdf-schema#label>	"山梨県"
<http://ja.dbpedia.org/resource/千葉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"千葉県"
<http://ja.dbpedia.org/resource/埼玉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"埼玉県"
<http://ja.dbpedia.org/resource/神奈川県>	<http://www.w3.org/2000/01/rdf-schema#label>	"神奈川県"

東京都の隣接県エンドポイント

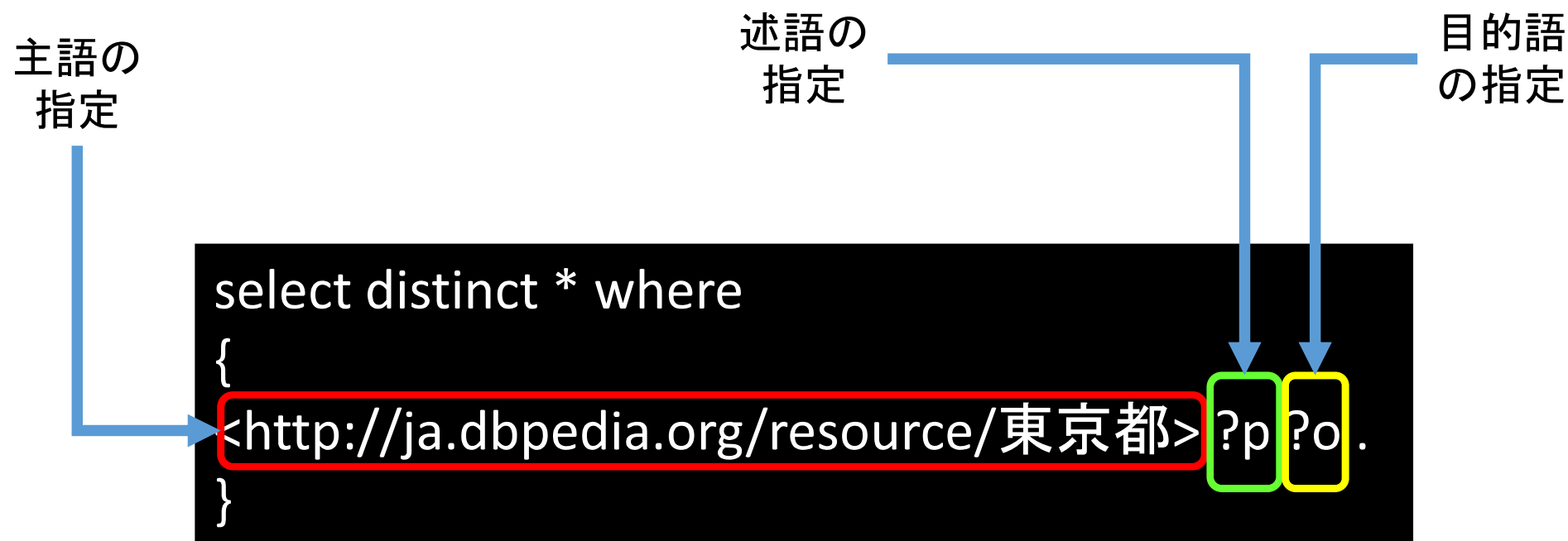
```
SELECT ?s ?o WHERE
{
  ?s <http://www.w3.org/2000/01/rdf-schema#label> ?o .
}
```

Load

s	o
http://ja.dbpedia.org/resource/山梨県	山梨県
http://ja.dbpedia.org/resource/東京都	東京都
http://ja.dbpedia.org/resource/神奈川県	神奈川県
http://ja.dbpedia.org/resource/千葉県	千葉県
http://ja.dbpedia.org/resource/埼玉県	埼玉県

<http://uedayou.net/sparql-mashup/test-endpoint/>

(3)主語に<http://ja.dbpedia.org/resource/東京都>
が指定されているトリプルを取得



- <http://ja.dbpedia.org/resource/東京都>に関するデータ(述語と目的語)全て検索

検索対象

?s	?p	?o
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#label>	"東京都"
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#comment>	"東京都(とうきょうと)は、日本の都道府県の一つである。"
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/山梨県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/千葉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/埼玉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/神奈川県>
<http://ja.dbpedia.org/resource/山梨県>	<http://www.w3.org/2000/01/rdf-schema#label>	"山梨県"
<http://ja.dbpedia.org/resource/千葉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"千葉県"
<http://ja.dbpedia.org/resource/埼玉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"埼玉県"
<http://ja.dbpedia.org/resource/神奈川県>	<http://www.w3.org/2000/01/rdf-schema#label>	"神奈川県"

東京都の隣接県エンドポイント

```
select distinct * where
{
<http://ja.dbpedia.org/resource/東京都> ?p ?o .
}
```

Load

p	o
http://www.w3.org/2000/01/rdf-schema#label	東京都
http://www.w3.org/2000/01/rdf-schema#comment	東京都(とうきょうと)は、日本の都道府県の一つである。
http://ja.dbpedia.org/property/隣接都道府県	http://ja.dbpedia.org/resource/山梨県
http://ja.dbpedia.org/property/隣接都道府県	http://ja.dbpedia.org/resource/千葉県
http://ja.dbpedia.org/property/隣接都道府県	http://ja.dbpedia.org/resource/埼玉県
http://ja.dbpedia.org/property/隣接都道府県	http://ja.dbpedia.org/resource/神奈川県

<http://uedayou.net/sparql-mashup/test-endpoint/>

(4) 東京都に隣接する県の名前

1. 東京都のURIから隣接する都道府県を表す
<http://ja.dbpedia.org/property/隣接都道府県>
を述語として持つ目的語を検索する
2. さらに、検索された目的語を主語に指定して、そのrdfs:labelを検索する

```
SELECT DISTINCT ?pref ?label WHERE
{
  <http://ja.dbpedia.org/resource/東京都> <http://ja.dbpedia.org/property/隣接都道府県> ?pref.
  ?pref <http://www.w3.org/2000/01/rdf-schema#label> ?label.
}
```

- 同じ代数(例えば ?pref)を異なるトリプルパターンで記述すると、同じURIを持つものののみ検索される (AND検索)

検索対象

主語	述語	目的語
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#label>	"東京都"
<http://ja.dbpedia.org/resource/東京都>	<http://www.w3.org/2000/01/rdf-schema#comment>	"東京都(?pref は、日本の都道府
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/山梨県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/千葉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/埼玉県>
<http://ja.dbpedia.org/resource/東京都>	<http://ja.dbpedia.org/property/隣接都道府県>	<http://ja.dbpedia.org/resource/神奈川県>
<http://ja.dbpedia.org/resource/山梨県>	<http://www.w3.org/2000/01/rdf-schema#label>	"山梨県"
<http://ja.dbpedia.org/resource/千葉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"千葉県"
<http://ja.dbpedia.org/resource/埼玉県>	<http://www.w3.org/2000/01/rdf-schema#label>	"埼玉県"
<http://ja.dbpedia.org/resource/神奈川県>	<http://www.w3.org/2000/01/rdf-schema#label>	"神奈川県"

「?pref」と同じURIを
主語に持つトリプル
を選択

?label

東京都の隣接県エンドポイント

```
SELECT DISTINCT ?pref ?label WHERE
{
  <http://ja.dbpedia.org/resource/東京都> <http://ja.dbpedia.org/property/隣接都道府県> ?pref.
  ?pref <http://www.w3.org/2000/01/rdf-schema#label> ?label.
}
```

Load

pref

label

<http://ja.dbpedia.org/resource/山梨県>

山梨県

<http://ja.dbpedia.org/resource/千葉県>

千葉県

<http://ja.dbpedia.org/resource/埼玉県>

埼玉県

<http://ja.dbpedia.org/resource/神奈川県>

神奈川県

<http://uedayou.net/sparql-mashup/test-endpoint/>

PREFIXによるURIの省略表記

- URI記述はPREFIXを利用することで省略表記が可能
 - クエリ先頭行に以下のような形式で追加
 - PREFIX **BINDNAME** : <**URI**>
 - 例 : PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 - PREFIX を指定するとURIを省略表記に
 - <http://www.w3.org/2000/01/rdf-schema#label>
- 
- PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 - rdfs:label

PREFIX rdfs: <<http://www.w3.org/2000/01/rdf-schema#>>

PREFIX dbpedia-ja: <<http://ja.dbpedia.org/resource/>>

PREFIX prop-ja: <<http://ja.dbpedia.org/property/>>

主語	述語	目的語
dbpedia-jp:東京都	rdfs:label	"東京都"
dbpedia-jp:東京都	rdfs:comment	“東京都(とうきょうと)は、日本の都道府県の一つである。”
dbpedia-jp:東京都	prop-ja:隣接都道府県	dbpedia-jp:山梨県
dbpedia-jp:東京都	prop-ja:隣接都道府県	dbpedia-jp:千葉県
dbpedia-jp:東京都	prop-ja:隣接都道府県	dbpedia-jp:埼玉県
dbpedia-jp:東京都	prop-ja:隣接都道府県	dbpedia-jp:神奈川県
dbpedia-jp:山梨県	rdfs:label	“山梨県”
dbpedia-jp:千葉県	rdfs:label	“千葉県”
dbpedia-jp:埼玉県	rdfs:label	“埼玉県”
dbpedia-jp:神奈川県	rdfs:label	“神奈川県”

(5)「東京都に隣接する県の名前」 のクエリを省略して書くと...

```
SELECT DISTINCT ?pref ?label WHERE
{
<http://ja.dbpedia.org/resource/東京都> <http://ja.dbpedia.org/property/隣接都道府県> ?pref.
?pref <http://www.w3.org/2000/01/rdf-schema#label> ?label.
}
```



```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpedia-ja: <http://ja.dbpedia.org/resource/>
PREFIX prop-ja: <http://ja.dbpedia.org/property/>

SELECT DISTINCT ?pref ?label WHERE
{
  dbpedia-ja:東京都 prop-ja:隣接都道府県 ?pref .
  ?pref rdfs:label ?label .
}
```

東京都の隣接県エンドポイント

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpedia-ja: <http://ja.dbpedia.org/resource/>
PREFIX prop-ja: <http://ja.dbpedia.org/property/>
```

```
SELECT DISTINCT ?pref ?label WHERE
{
  dbpedia-ja:東京都 prop-ja:隣接都道府県 ?pref .
  ?pref rdfs:label ?label .
}
```

Load

pref

label

<http://ja.dbpedia.org/resource/山梨県>

山梨県

<http://ja.dbpedia.org/resource/千葉県>

千葉県

<http://ja.dbpedia.org/resource/埼玉県>

埼玉県

<http://ja.dbpedia.org/resource/神奈川県>

神奈川県

<http://uedayou.net/sparql-mashup/test-endpoint/>

(6) 同一主語の省略

- 同じ主語から異なる複数の述語を指定する場合、目的語のあとのピリオド「.」をセミコロン「;」にすることで次のトリプルパターンの主語を省略できます

```
SELECT DISTINCT * WHERE {  
  dbpedia-ja:東京都 rdfs:label ?label .  
  dbpedia-ja:東京都 rdfs:comment ?comment .  
}
```



```
SELECT DISTINCT * WHERE {  
  dbpedia-ja:東京都 rdfs:label ?label;  
  dbpedia-ja:東京都 rdfs:comment ?comment.  
}
```

主語を
省略
できる

省略を終了するトリプルパターンの最後は必ずピリオドにすること

東京都の隣接県エンドポイント

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpedia-ja: <http://ja.dbpedia.org/resource/>

SELECT DISTINCT * WHERE {
  dbpedia-ja:東京都 rdfs:label ?label ;
                    rdfs:comment ?comment .
}
```

Load

label	comment
東京都	東京都(とうきょうと)は、日本の都道府県の一つである。

SPARQLの便利な検索機能

検索句	機能
DISTINCT	重複する検索結果を除外
LIMIT	検索結果の上限を設定
OFFSET	検索結果の取得位置を指定
ORDER BY	検索結果の並び順を指定
OPTIONAL	OPTIONAL内は任意検索
FILTER	検索結果のフィルタリングが可能
REGEX	正規表現による検索が可能
BIND	新たな変数への割り当てが可能
CONCAT	文字列の結合が可能
REPLACE	文字列の置き換えが可能(正規表現使用可)
SUBSTR	文字列の切り出しが可能
COUNT	検索件数を表示
GROUP BY	変数のグループ化が可能
HAVING	グループ化した変数の絞込みが可能
各機能の詳しいの説明はSlideShareの資料を参照してください	

SPARQLの 便利な検索機能

DISTINCT

```
SELECT DISTINCT ?s ?p ?o WHERE  
{  
  ?s ?p ?o.  
}
```

- 「SELECT」と代数の間に「DISTINCT」を入れると指定した全ての代数で重複したパターンがあった場合、検索結果から除外される

```
select ?s where {  
  ?s ?p ?o.  
}
```

s

<http://ja.dbpedia.org/resource/埼玉県>

<http://ja.dbpedia.org/resource/千葉県>

<http://ja.dbpedia.org/resource/神奈川県>

<http://ja.dbpedia.org/resource/東京都>

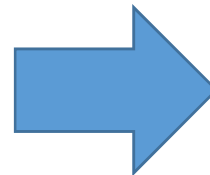
<http://ja.dbpedia.org/resource/東京都>

<http://ja.dbpedia.org/resource/東京都>

<http://ja.dbpedia.org/resource/東京都>

<http://ja.dbpedia.org/resource/東京都>

<http://ja.dbpedia.org/resource/東京都>



```
select distinct ?s where {  
  ?s ?p ?o.  
}
```

s

<http://ja.dbpedia.org/resource/埼玉県>

<http://ja.dbpedia.org/resource/千葉県>

<http://ja.dbpedia.org/resource/神奈川県>

<http://ja.dbpedia.org/resource/東京都>

<http://ja.dbpedia.org/resource/山梨県>

LIMIT

- 検索結果の上限を設定
- WHERE{}の後ろに記述
- 「LIMIT 10」で最大10件取得

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
SELECT DISTINCT ?uri ?label WHERE
```

```
{
```

```
  ?uri rdfs:label ?label .
```

```
}
```

```
LIMIT 10
```

OFFSET

- 検索結果の取得位置を指定
- WHERE{}の後ろに記述
- 「OFFSET 100」は100番目から取得

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

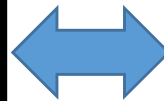
SELECT DISTINCT ?uri ?label WHERE
{
    ?uri rdfs:label ?label .
}
LIMIT 10
OFFSET 100
```

ORDER BY / ORDER BY DESC

- 検索結果の並び順を指定
- WHERE{}の後ろに記述
- 「ORDER BY ?created」で作成日昇順にソート
- 「ORDER BY DESC(?created)」なら作成日降順
- LIMIT、OFFSETがあれば、その前に挿入

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT DISTINCT ?uri ?created WHERE
{
  ?uri dc:created ?created.
}
ORDER BY ?created
LIMIT 10
OFFSET 100
```



```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT DISTINCT ?uri ?created WHERE
{
  ?uri dc:created ?created.
}
ORDER BY DESC(?created)
LIMIT 10
OFFSET 100
```

OPTIONAL

- OPTIONAL{ ... } 内のグラフパターンはパターンに一致しなくても、OPTIONAL外のパターンに一致すれば検索される
- WHERE{ ... } 内に記述

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT DISTINCT ?uri ?label ?description WHERE
{
  ?uri rdfs:label ?label .
  OPTIONAL {
    ?uri dc:description ?description.
  }
}
```


FILTER

- 検索結果のフィルタリングが可能
- WHERE{ ... } 内に記述
- dcterms:issued が 2000(年)以上のものを検索

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcterms: <http://purl.org/dc/terms/>

SELECT DISTINCT ?uri ?label ?issued WHERE
{
    ?uri rdfs:label ?label ;
    dcterms:issued ?issued .
    FILTER ( ?issued >= 2000 )
}
```

REGEX

- 正規表現による検索が可能
- FILTER と組み合わせて利用する
- WHERE{ ... } 内に記述
- 書き方
 - REGEX(検索対象の変数 , 正規表現)

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?uri ?label WHERE
{
    ?uri rdfs:label ?label .
    FILTER ( REGEX ( ?label, "^東京.*$" ) )
}
```

?label を「東京」で前方一致検索

BIND / CONCAT

- BIND: 指定した変数・データを別の名前の変数に割り当てが可能
 - BIND([割り当てるデータ] AS [変数])
- CONCAT: 変数や文字列同士の結合が可能、値はリテラルに
 - CONCAT([変数or文字列] , [変数or文字列] , ...)

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
SELECT DISTINCT ?uri ?label ?description ?text WHERE
```

```
{  
  ?uri rdfs:label ?label ;  
  dc:description ?description .  
  BIND( CONCAT( "<h1>" , ?label , "</h1><p>" , ?description , "</p>" ) AS ?text )  
}
```

?text は「<h1>[?labelのデータ]</h1><p>[?descriptionのデータ]</p>」になる

REPLACE

- 文字列の置き換えが可能
- 書き方

REPLACE(置換対象の変数 , 置換対象の文字列 , 置換後の文字列)

- BIND を使うと、置き換わった文字列を変数に設定できる

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
SELECT DISTINCT ?uri ?label ?description ?text WHERE
{
  ?uri rdfs:label ?label ;
  dc:description ?description .
  BIND( REPLACE( ?description , “京都” , “大阪” ) AS ?text )
}
```

SUBSTR

- 文字列の切り出しが可能
- 書き方

SUBSTR(切出対象の変数 , 開始位置 , 切り出し文字数)

- BIND を使うと、置き換わった文字列を変数に設定できる

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
SELECT DISTINCT ?uri ?label ?description ?text WHERE
```

```
{
```

```
  ?uri rdfs:label ?label ;
```

```
  dc:description ?description .
```

```
  BIND( SUBSTR( ?description , 10 , 5 ) AS ?text )
```

```
}
```

COUNT

- 変数のデータ総数を計算

```
SELECT COUNT(*) WHERE  
{  
    ?s ?p ?o .  
}
```

全トリプル数を計算

GROUP BY / HAVING

- GROUP BY: 指定の変数でデータを集約(グループ化)する
- HAVING: グループ化されたデータの絞り込み

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
SELECT ?creator (COUNT( ?uri ) AS ?count)
```

COUNT(?uri)の結果が?countに

```
WHERE
```

```
{
```

```
  ?uri dc:creator ?creator .
```

```
}
```

?creator(作者)で集約

```
GROUP BY ?creator
```

```
ORDER BY DESC(?count)
```

異なる主語が5つ以上
ある作者のみ対象

```
HAVING (COUNT(?uri) >= 5)
```

FROM

- トリプルデータには、その全体を現す名前(グラフ名)が指定されています
- FROMを使うとグラフ名ごとに検索が行えます

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?uri ?label
FROM
<http://lod.sfc.keio.ac.jp/challenge2014/show_status.php?id=d001>
WHERE
{
    ?uri rdfs:label ?label .
}
```

<http://lod.sfc.keio.ac.jp/challenge2014/show_status.php?id=d001>
というグラフ名がついているトリプルデータを対象とする

FROM

http://lod.sfc.keio.ac.jp/challenge2014/show_status.php?id=d001

グラフ名ごとに
検索が可能

http://lod.sfc.keio.ac.jp/challenge2013/show_status.php?id=d030

LODチャレンジエンドポイントでは、
エントリーページのURLが
グラフ名になっています

主語	述語	目的語
dbpedia-ja:東京都	rdfs:label	"東京都"
dbpedia-ja:東京都	prop-ja:隣接都道府県	dbpedia-ja:山梨県
dbpedia-ja:東京都	prop-ja:隣接都道府県	dbpedia-ja:千葉県
dbpedia-ja:東京都	prop-ja:隣接都道府県	dbpedia-ja:埼玉県
dbpedia-ja:東京都	prop-ja:隣接都道府県	dbpedia-ja:神奈川県
dbpedia-ja:山梨県	rdfs:label	"山梨県"
dbpedia-ja:千葉県	rdfs:label	"千葉県"
dbpedia-ja:埼玉県	rdfs:label	"埼玉県"
dbpedia-ja:神奈川県	rdfs:label	"神奈川県"

主語	述語	目的語
dbpedia-ja:大阪府	rdfs:label	"大阪府"
dbpedia-ja:大阪府	prop-ja:隣接都道府県	dbpedia-ja:兵庫県
dbpedia-ja:大阪府	prop-ja:隣接都道府県	dbpedia-ja:和歌山県
dbpedia-ja:大阪府	prop-ja:隣接都道府県	dbpedia-ja:京都府
dbpedia-ja:大阪府	prop-ja:隣接都道府県	dbpedia-ja:奈良県
dbpedia-ja:兵庫県	rdfs:label	"兵庫県"
dbpedia-ja:和歌山県	rdfs:label	"和歌山県"
dbpedia-ja:京都府	rdfs:label	"京都府"
dbpedia-ja:奈良県	rdfs:label	"奈良県"

データ型のキャスト

- STR(?data)
 - データを文字列に変換
- URI(?data)
 - データをURIに変換
- xsd:integer(?data)
 - データを整数値型に変換
- xsd:float(?data) / xsd:double(?data)
 - データを浮動小数点型に変換
- xsd:dateTime(?data)
 - データをdateTime型に変換

※ PREFIX xsd: <<http://www.w3.org/2001/XMLSchema#>> があることが前提

エンドポイントの結果(JSON)を プログラムで取得するには？

```
{ "head": { "vars": ["link", "title", "lat", "long" ] },  
  "results": {  
    "bindings": [  
      { "link": { "type": "uri",  
                  "value": "http://linkdata.org/resource/rdf1s933i#8542" },  
        "title": { "type": "literal", "xml:lang": "ja",  
                   "value": "福島消防署上福島出張所"},  
        "lat": { "type": "typed-literal",  
                 "datatype": "http://www.w3.org/2001/XMLSchema#float",  
                 "value": "34.6936" },  
        "long": { "type": "typed-literal",  
                  "datatype": "http://www.w3.org/2001/XMLSchema#float",  
                  "value": "135.482" }    ]  
  }
```



```
var results = json.results.bindings;  
for ( var i=0 ; i < results.length ; i++ ) {  
  var result = results[i];  
  var title = result.title.value;  
}
```

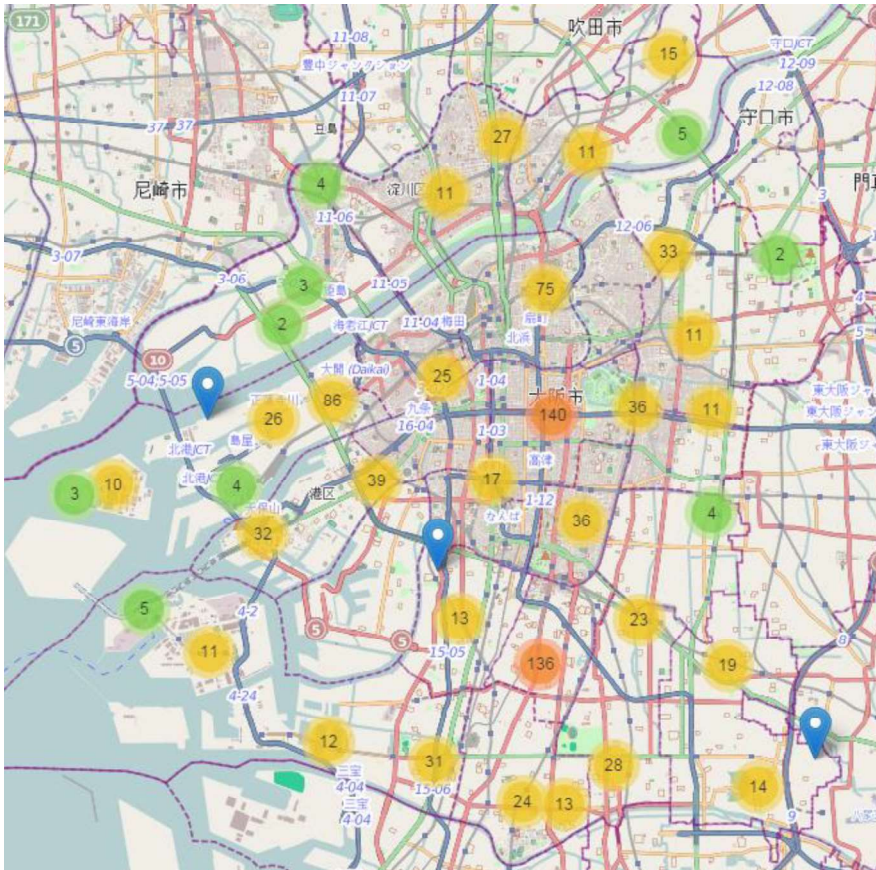
Javascript

```
$results = $json["results"]["bindings"];  
foreach ( $results as $result ) {  
  $title = $result["title"]["value"];  
}
```

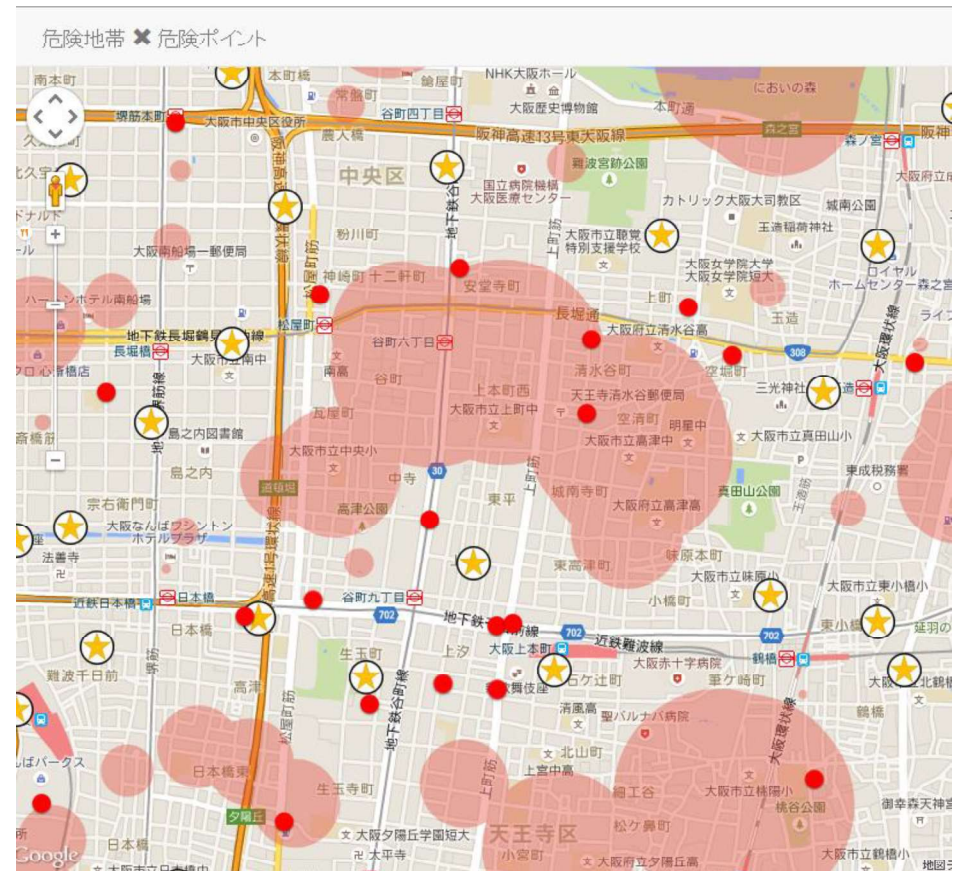
PHP

SPARQLでアプリを
作ってみよう！

SPARQLでアプリ作成



地図アプリ



危険地帯 x 危険ポイント

和歌山県公共施設データ SPARQLエンドポイント

- API URL
 - http://lodcu.cs.chubu.ac.jp/SparqlEPCU/api/wakayama_pf
- データモデル
 - 大阪市 施設情報・防災情報のLODとほぼ同じ
<http://lodosaka.hozo.jp/OsakaCityMapLOD.htm>

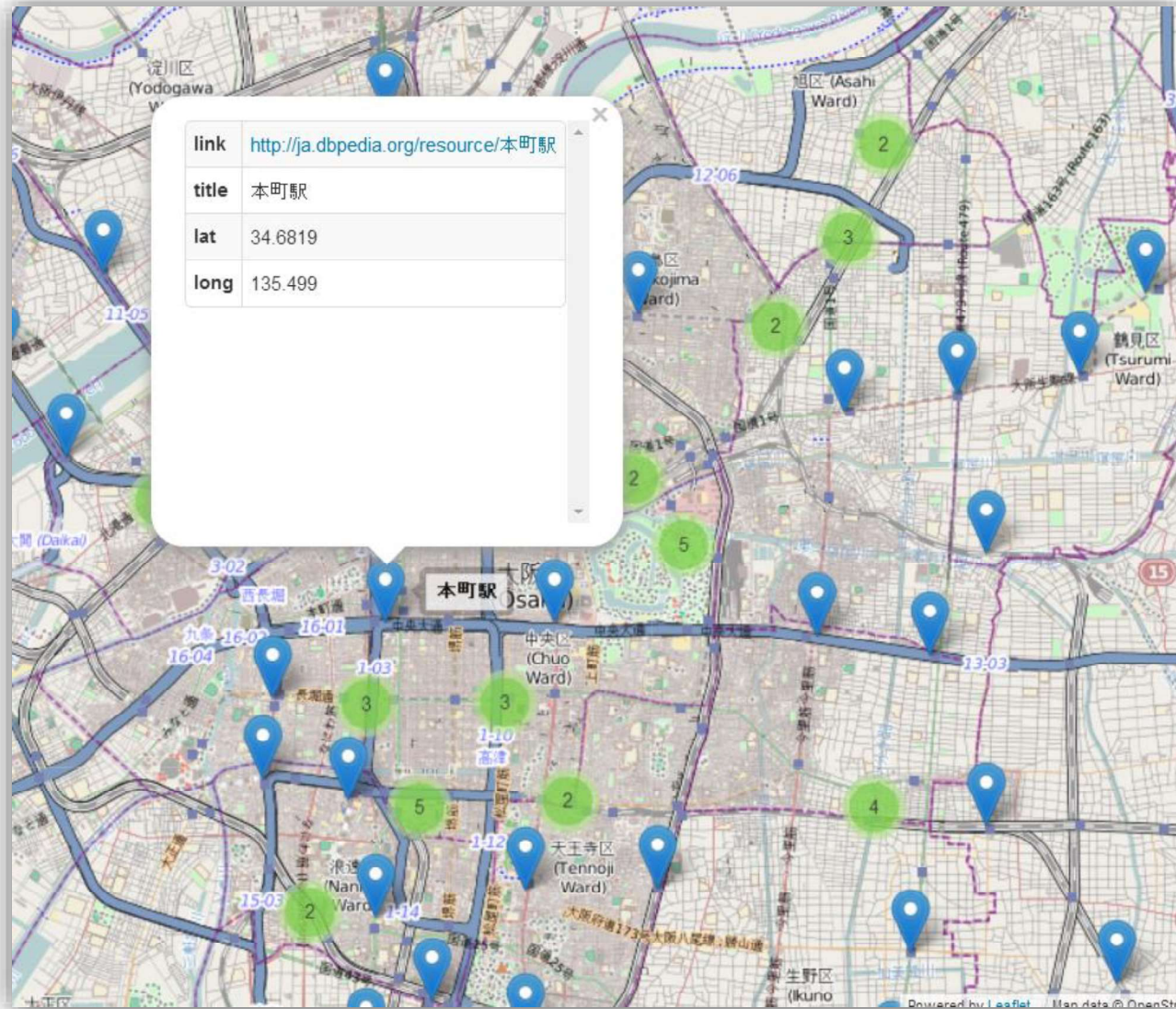
述語URI(プロパティ)	説明
http://schema.org/name	名称
http://www.w3.org/2003/01/geo/wgs84_pos#lat	緯度
http://www.w3.org/2003/01/geo/wgs84_pos#long	経度
http://lodosaka.hozo.jp/category_1	カテゴリ※
http://schema.org/address	住所

※カテゴリは、「大規模小売店舗」「金融機関・郵便局」「文化施設」「トイレ」
「官公庁舎等」「医療施設」「観光関連施設」「宿泊施設」「その他」のいずれかが入っている

http://schema.org/name	名称
http://www.w3.org/2000/01/rdf-schema#label	名称(上と同じ)
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	データ型 (< http://lodosaka.hozo.jp/class/施設情報 >固定)
http://www.w3.org/2003/01/geo/wgs84_pos#lat	緯度
http://www.w3.org/2003/01/geo/wgs84_pos#long	経度
http://lodosaka.hozo.jp/category_1	カテゴリ 「大規模小売店舗」「金融機関・郵便局」 「文化施設」「トイレ」「官公庁舎等」「医療施設」 「観光関連施設」「宿泊施設」「その他」
https://schema.org/telephone	電話番号
https://schema.org/faxNumber	FAX番号
http://lodosaka.hozo.jp/close	閉店・閉館日
https://schema.org/openingHours	開店・開館時間
http://lodosaka.hozo.jp/toilet_info	トイレに関する情報
http://lodosaka.hozo.jp/facility_info	施設に関する情報
http://lodosaka.hozo.jp/facility_guide	施設案内
http://lodosaka.hozo.jp/parking_info	駐車場情報

地図アプリ

- leaflet-simple-sparql
 - <https://github.com/uedayou/leaflet-simple-sparql>



使い方: leaflet-simple-sparql

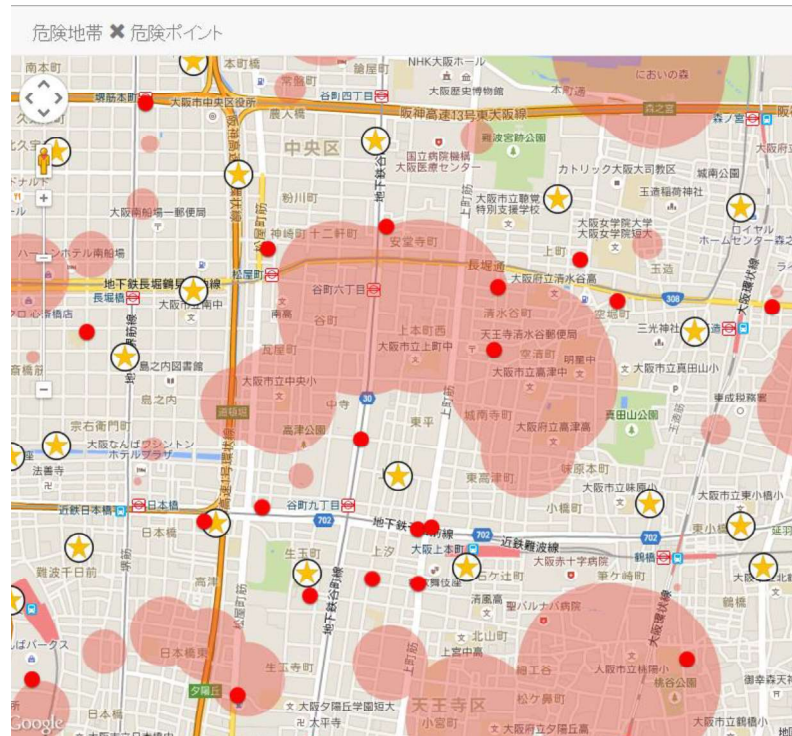
- githubのページから「Download ZIP」ボタンを押して、ソースコード一式をダウンロード
- 解凍して「config.js」にエンドポイントとクエリを記述する
- index.html をブラウザで開く

```
var endpoint = "http://ja.dbpedia.org/sparql";
var query = (function () {/*
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select * where {
    ?link rdfs:label ?title;
    geo:lat ?lat;
    geo:long ?long.
}
limit 1000
*/}).toString().match(/¥n([¥s¥S]*)¥n/)[1];
```

危険地帯 x 危険ポイント

- 「大阪市 警察署 x 犯罪発生」を汎用的に使えるようにしたアプリ
- 任意の2種類の位置情報が含まれるデータ群を地図上に重ね合わせることが可能

<https://github.com/uedayou/dangerzone-x-dangerpoint-sparql>



使い方: 危険地帯 x 危険ポイント

- githubのページから「Download ZIP」ボタンを押して、ソースコード一式をダウンロード

<https://github.com/uedayou/dangerzone-x-dangerpoint-sparql>

- 解凍して「config.js」にエンドポイントとクエリを記述する
- index.html をブラウザで開く

```
// 危険地帯用SPARQLエンドポイントを指定
var dz_endpoint = "http://db.lodc.jp/sparql";
// 危険地帯用SPARQLクエリを指定
// ?latitude , ?longitude, ?label を必ず指定してください
var dz_query = (function () {/*
SELECT DISTINCT *
WHERE{
    ?uri <http://lodosaka.hozo.jp/category_2> "警察・交番"@ja ;
    <http://schema.org/name> ?label;
    <http://www.w3.org/2003/01/geo/wgs84_pos#lat> ?latitude ;
    <http://www.w3.org/2003/01/geo/wgs84_pos#long> ?longitude .
} */}).toString().match(/¥n([¥s¥S]*)¥n/)[1];
// 危険ポイント用SPARQLエンドポイントを指定
var dp_endpoint = "http://sparql.endpoint/sparql";
// 危険ポイント用SPARQLクエリを指定
// ?latitude , ?longitude, ?label を必ず指定してください
var dp_query = (function () {/*
select * where {
    ?uri geo:lat ?latitude;
    geo:long ?longitude;
    <http://linkdata.org/property/rdf1s2565i#category> ?label.
    filter(xsd:float(?latitude)>34.586018936001786)
}
*/}).toString().match(/¥n([¥s¥S]*)¥n/)[1];
// 中心位置を指定
var initial_latitude =34.68206400648744;
var initial_longitude =135.49816131591797;
```