



プログラミング演習2

XMLとWeb API

中村、小林、橋本、辻野

本日の内容



- Webからの情報取得方法を学ぶ
 - CSV/TSV
 - XML
 - JSON

- Web APIを使って情報を取得してみる



- eXtensible Markup Language
 - W3C (World Wide Web Consortium) で採択されたWeb上でのデータのやりとりに注目した構造化文書記述のためのフォーマット
- XMLの特徴
 - 新しいタグを定義することが可能
 - 構造は任意の形でネスト可能 (繰り返し)
 - XMLはデータ記述言語であり、表示能力は持っていない



- データを機械可読な形に記述可能
 - HTMLは機械での認識が難しい
- 関係データベースで表現できない半構造データを扱うことが可能
 - 生物学のデータ
 - Web上の各種データ

まずは見てみよう！



- 下記のXMLを開いてみよう

- <https://www.city.sabae.fukui.jp/xml/population/population.xml>
- <https://lecture.nkmr.io/2015/fms.xml>



- 要素： XMLの1単位
 - `<population> ... </population>`, `<year> ... </year>`, `<man> ... </man>`
 - 空要素にもなりうる： `<director></director>`
- タグ： `population`, `year`, `man`, `woman`, `age`など
- 開始タグ： `<population>`
- 終了タグ： `</population>`
- 属性： 要素の中で指定する属性
 - `<player position="GK" number="1" ...>`
 - 属性は開始タグの中で指定

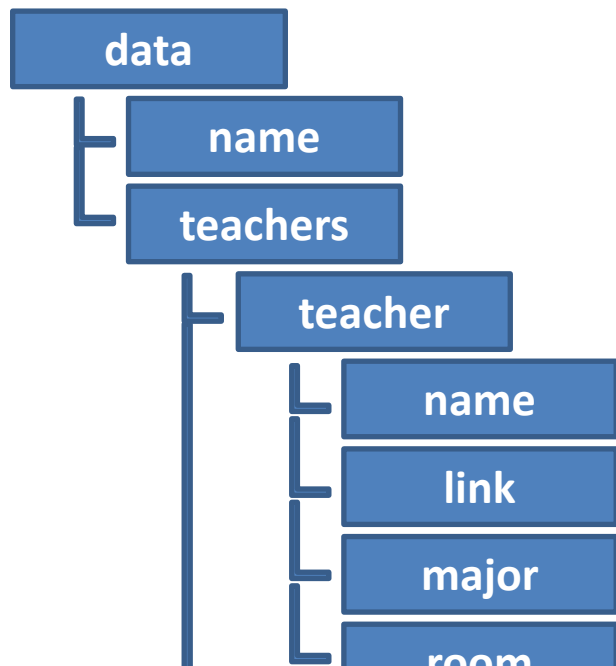


XML処理のAPI

- `getParent()` 親を取得
- `getChildren("name")` **nameの子達のリストを取得**
- `getChild("name")` **nameの子を取得**
- `getName()` 要素の名前を取得
- `hasChildren("name")` nameの子達が居るかチェック
- `getContent()` **要素の内容のテキスト情報を取得**
- `getIntContent()` **要素の内容の整数情報を取得**
- `getFloatContent()` **要素の内容の実数情報を取得**
- `getString("atr")` atr属性を文字列として取得
- `getInt("atr")` atr属性を整数として取得
- `getFloat("atr")` atr属性を実数として取得



- fms.xml について下記に挑戦してみよう
 - 学科名を表示
 - 教員リストを表示
 - 講義名リストを表示



```
<data>
  <name>先端メディアサイエンス学科</name>
  <teachers>
    <teacher>
      <name>宮下 芳明</name>
      <link>http://www.homei.com/</link>
      <major>インタラクシヨ</major>
      <room>1017</room>
      <lectures>
        <lecture>エンタテイメントプログラミング演習</lecture>
        <lecture>コンテンツ・エンタテインメント概論</lecture>
        <lecture>コンテンツメディアプログラミング実習2</lecture>
      </lectures>
    </teacher>
    <teacher>
      <name>小松 孝徳</name>
      <link>http://www.tkomat-lab.com/</link>
      <major>認知心理学</major>
      <room>1114</room>
      <lectures>
        <lecture>プログラミング演習1</lecture>
        <lecture>プログラミング演習2</lecture>
      </lectures>
    </teacher>
    <teacher>
      <name>中村 聡史</name>
      <link>http://snakamura.org/</link>
      <major>ヒューマンインタフェース</major>
      <room>1007</room>
      <lectures>
        <lecture>プログラミング演習1</lecture>
        <lecture>プログラミング演習2</lecture>
        <lecture>コンテンツメディアプログラミング実習2</lecture>
      </lectures>
    </teacher>
    <teacher>
      <name>鈴木 正明</name>
      <link>http://www.ms.u-tokyo.ac.jp/~macky/</link>
      <major>位相幾何学</major>
      <room>1002</room>
      <lectures>
        <lecture>線形代数1</lecture>
        <lecture>線形代数2</lecture>
        <lecture>情報数理基礎</lecture>
      </lectures>
    </teacher>
    <teacher>
      <name>小林 玲</name>

```

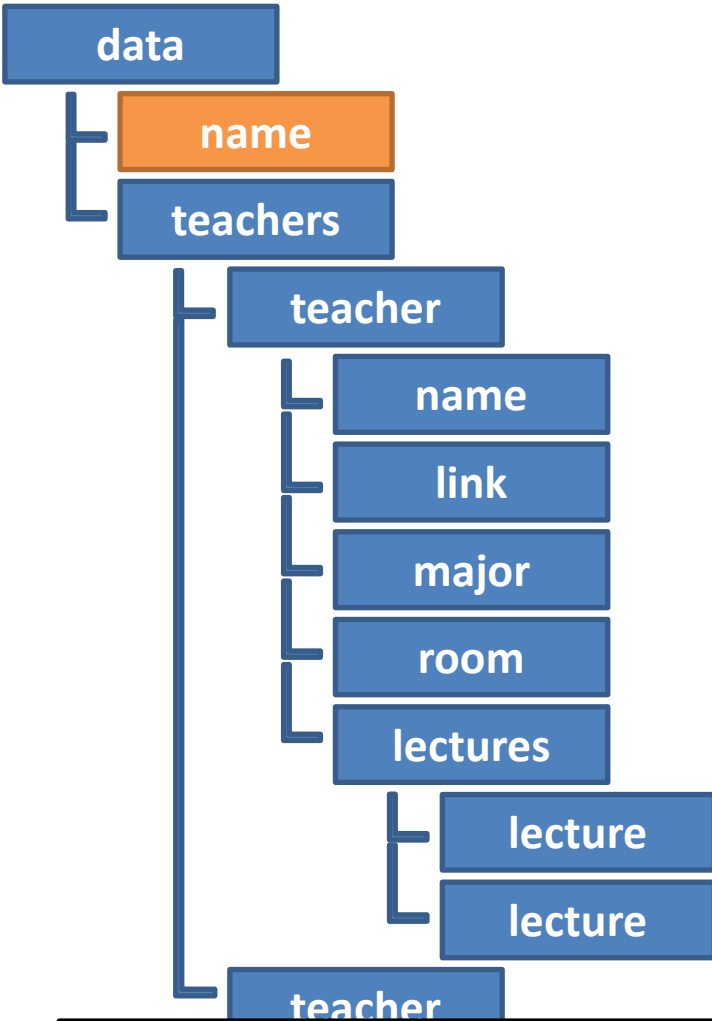
学科名を取得



学科名はオレンジ色のname

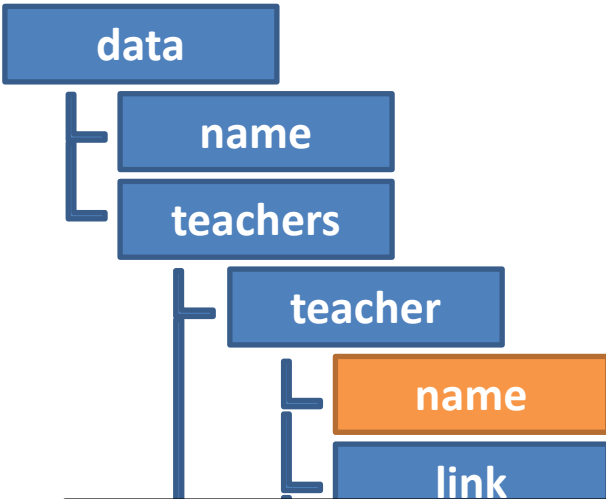
data → name

講義名は、dataの子



```
XML xml = loadXML( "https://lecture.nkmr.io/2015/fms.xml" );  
println( xml.getChild( "name" ).getContent() );
```

教員名を取得



教員名はオレンジ色のname

data → teachers → teacher →

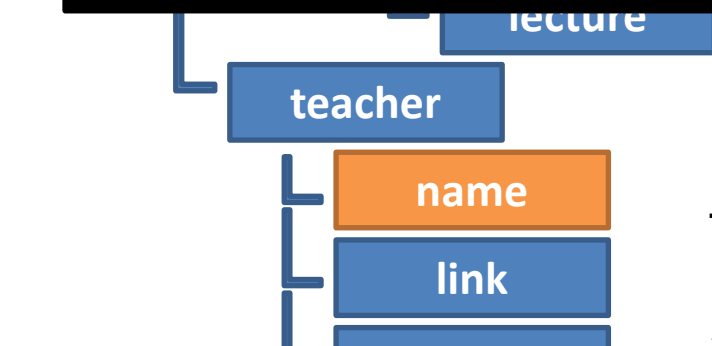
name

教員名はteachersの子である

teacherの子のnameに該当する

```
XML xml = loadXML( "https://lecture.nkmr.io/2015/fms.xml" );
XML teachers = xml.getChild("teachers");
XML[] teacher = teachers.getChildren("teacher");
println( teacher[0].getChild("name").getContent() );
println( teacher[1].getChild("name").getContent() );
```

:



teacherは同じものが複数存在する
ため子達（配列）として取得する

教員名を取得



data

name

教員名はオレンジ色のname

```
XML xml = loadXML( "https://lecture.nkmr.io/2015/fms.xml" );
XML teachers = xml.getChild("teachers");
XML [] teacher = teachers.getChildren("teacher");
for( int i=0; i<teacher.length; i++ ){
    println( teacher[i].getChild("name").getContent() );
}
```

major

```
XML xml = loadXML( "https://lecture.nkmr.io/2015/fms.xml" );
XML [] teacher = xml.getChild("teachers").getChildren("teacher");
for( int i=0; i<teacher.length; i++ ){
    println( teacher[i].getChild("name").getContent() );
}
```

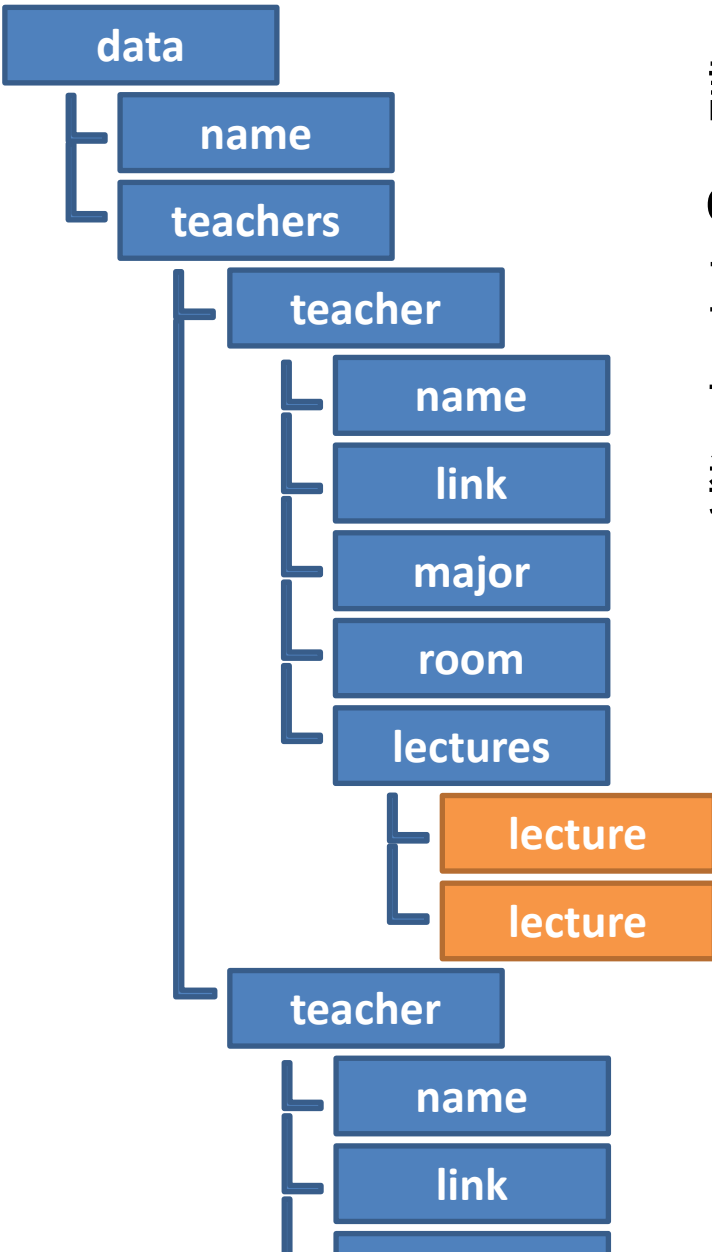
teacher

name

link

teacherは同じものが複数存在する
ため子達（配列）として取得する

講義リストを取得



講義名はオレンジ色のlecture

data → teachers → teacher →
lectures → **lecture**

teacherとlectureは同じものが複数存在するため子達（配列）として取得

講義リストを取得



```
XML xml = loadXML( "https://lecture.nkmr.io/2015/fms.xml" );
XML teachers = xml.getChild("teachers");
XML [] teacher = teachers.getChildren("teacher");
XML lectures = teacher[0].getChild( "lectures" );
XML [] lecture = lectures.getChildren( "lecture" );
println( lecture[0].getContent() );
println( lecture[1].getContent() );
:
```

major

```
XML xml = loadXML( "https://lecture.nkmr.io/2015/fms.xml" );
XML [] teacher = xml.getChild("teachers").getChildren("teacher");
for( int i=0; i<teacher.length; i++ ){
    XML [] lecture = teacher[i].getChild("lectures").getChildren("lecture");
    for( int j=0; j<lecture.length; j++ ){
        println( lecture[j].getContent() );
    }
}
```

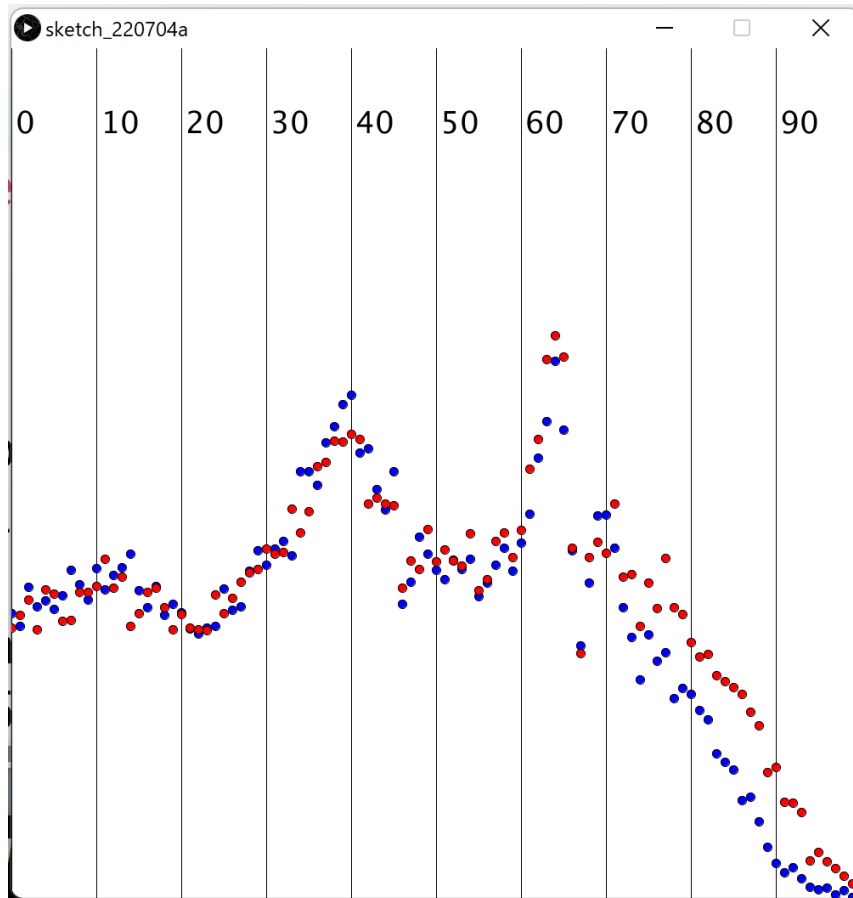
取得

鯖江市の人口XML



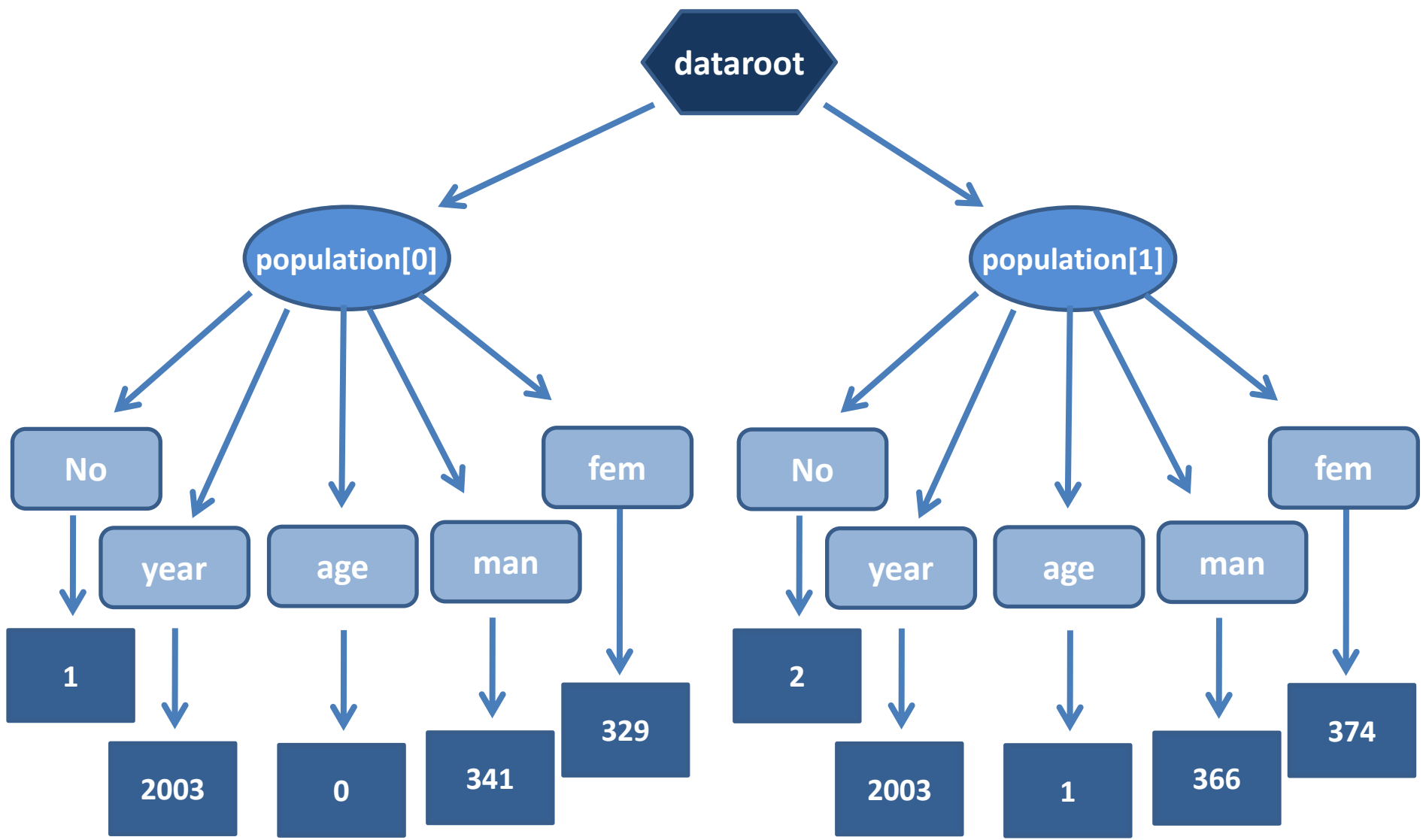
- 鯖江市の人口XMLから、2003年の男女の年齢ごとの人口を可視化してみましょう！

<https://www.city.sabae.fukui.jp/xml/population/population.xml>



```
<?xml version="1.0" encoding="UTF-8" ?>
<dataroot xmlns:od="urn:schemas-microsoft-com:officedata" xmlns:xsi="http://www
  <population>
    <No>1</No>
    <year>2003</year>
    <age>0</age>
    <man>341</man>
    <fem>329</fem>
    <uri>http://www3.city.sabae.fukui.jp/xml/population/#0001</uri>
  </population>
  <population>
    <No>2</No>
    <year>2003</year>
    <age>1</age>
    <man>366</man>
    <fem>374</fem>
    <uri>http://www3.city.sabae.fukui.jp/xml/population/#0002</uri>
  </population>
  <population>
    <No>3</No>
    <year>2003</year>
    <age>2</age>
    <man>360</man>
    <fem>367</fem>
    <uri>http://www3.city.sabae.fukui.jp/xml/population/#0003</uri>
  </population>
  <population>
    <No>4</No>
    <year>2003</year>
    <age>3</age>
    <man>384</man>
    <fem>372</fem>
    <uri>http://www3.city.sabae.fukui.jp/xml/population/#0004</uri>
  </population>
  <population>
    <No>5</No>
    <year>2003</year>
    <age>4</age>
    <man>399</man>
    <fem>351</fem>
    <uri>http://www3.city.sabae.fukui.jp/xml/population/#0005</uri>
  </population>
  <population>
    <No>6</No>
    <year>2003</year>
    <age>5</age>
    <man>380</man>
    <fem>294</fem>
    <uri>http://www3.city.sabae.fukui.jp/xml/population/#0006</uri>
  </population>
  <population>
    <No>7</No>
    <year>2003</year>
    <age>6</age>
    <man>350</man>
    <fem>343</fem>
    <uri>http://www3.city.sabae.fukui.jp/xml/population/#0007</uri>
  </population>
  <population>
    <No>8</No>
    <year>2003</year>
    <age>7</age>
```

XMLの構造 (鯖江市の人口XML)



XML の構造 (鯖江市の)

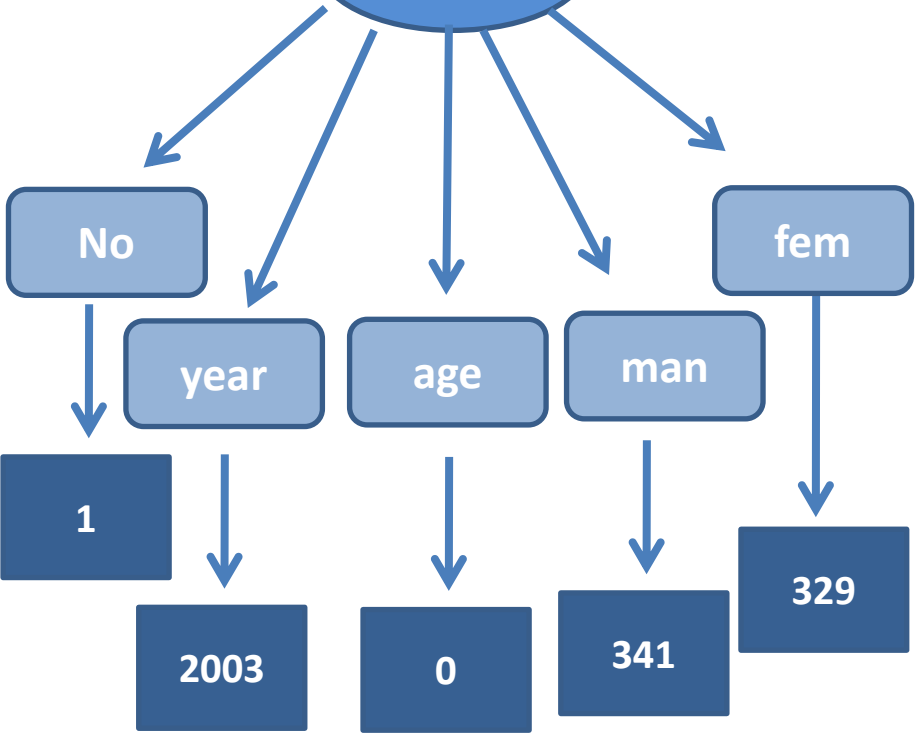
```

▼<dataroot xmlns:od="urn:schemas-microsoft-com:officedata" xmlns:xsi="http://www
▼<population>
  <No>1</No>
  <year>2003</year>
  <age>0</age>
  <man>341</man>
  <fem>329</fem>
  <uri>http://www3.city.sabae.fukui.jp/xml/population/#0001</uri>
</population>
▼<population>
  <No>2</No>
  <year>2003</year>
  <age>1</age>
  <man>366</man>
  <fem>374</fem>
  <uri>http://www3.city.sabae.fukui.jp/xml/population/#0002</uri>
</population>
▼<population>
  <No>3</No>
  <year>2003</year>
  <age>2</age>
  <man>360</man>
  <fem>367</fem>
  <uri>http://www3.city.sabae.fukui.jp/xml/population/#0003</uri>
</population>
▼<population>
  <No>4</No>
  <year>2003</year>
  <age>3</age>
  <man>384</man>
  <fem>372</fem>
  <uri>http://www3.city.sabae.fukui.jp/xml/population/#0004</uri>
</population>
▼<population>
  <No>5</No>
  <year>2003</year>
  <age>4</age>
  <man>399</man>
  <fem>351</fem>
  <uri>http://www3.city.sabae.fukui.jp/xml/population/#0005</uri>
</population>
▼<population>
  <No>6</No>
  <year>2003</year>
  <age>5</age>
  <man>380</man>
  <fem>294</fem>
  <uri>http://www3.city.sabae.fukui.jp/xml/population/#0006</uri>
</population>
▼<population>
  <No>7</No>
  <year>2003</year>
  <age>6</age>
  <man>350</man>
  <fem>343</fem>
  <uri>http://www3.city.sabae.fukui.jp/xml/population/#0007</uri>
</population>
▼<population>
  <No>8</No>
  <year>2003</year>
  <age>7</age>

```

data

population[0]



XMLをどう読み込むか？



- 試しに下記のプログラムを書いてみよう！

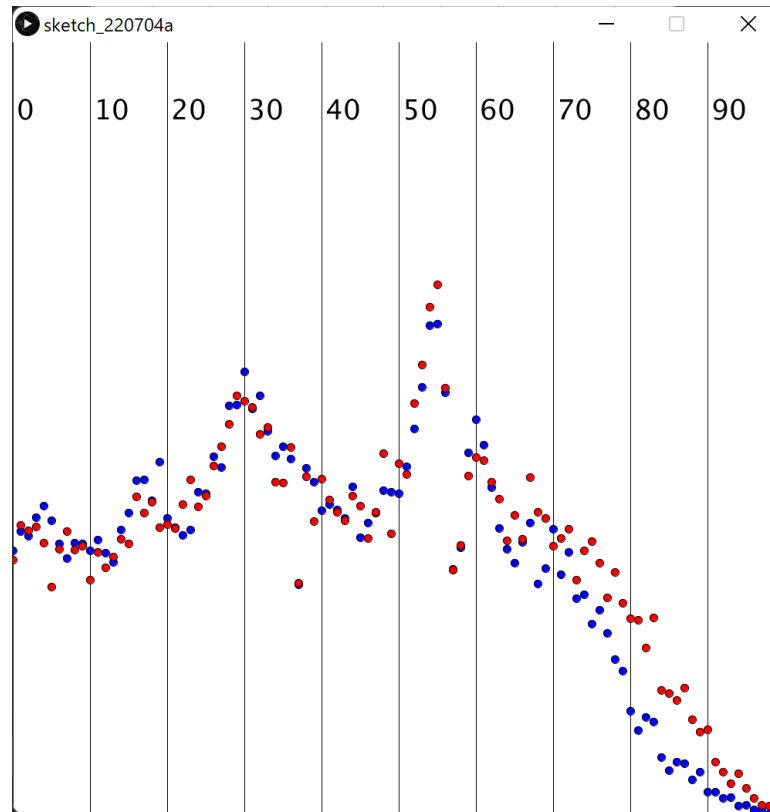
```
XML xml =
    loadXML("https://www.city.sabae.fukui.jp/xml/population/population.xml");

XML[] children = xml.getChildren("population");
for( int i=0; i<children.length; i++ ){
    println( children[i] );
    println( children[i].getChild("year") );
    println( children[i].getChild("year").getIntContent() );
}
```

- loadXMLでXMLを読み込んで、getChildrenでその子どもたちを取得して、getChildでその子どもを取得し、getIntContentでその内容を表示！



- 2003年の年齢ごとの人口を可視化せよ！





- loadXMLをdraw()内部に書いちゃだめ！
 - draw()の度にloadXMLでWebから情報をもってくると、XML提供者に迷惑をかけてしまう
 - loadXMLを行うのは、初期設定setup()か、何らかの処理（mousePressed()など）のときだけにしましょう

XMLを読み込んでみよう



- XMLはよくサイトの更新情報などで活用されている
 - <https://comiqa.com/rss.xml>
 - <https://nkmr-lab.org/feed>
- 情報取得や検索などにも使える
 - Songle
 - <https://widget.songle.jp/api/v1/songs/search.xml?q=miku>



- JavaScript Object Notation

- loadJSONObject(): JSONの読み込み
- parseJSONObject(): 文字列をJSON形式に変換
- getJSONObject(key): keyという子の取得
- getJSONArray(key): keyという子たちの取得
- getString(key): keyに設定された文字列の取得
- getInt(key), getFloat(key): keyに設定された整数・実数の取得
- JSONは
 - { 要素名: 要素の値 } という形で定義
 - [要素1, 要素2] という形で配列を定義

JSONの読み込み方



- JSONデータを読み込む（返り値はJSONObject）
 - `loadJSONObject(FilePath);`
- 取得したいJSON配列を取得する
 - `(JSONObject).getJSONArray("name");`
 - JSONArrayからの*i*番目のデータの読み込みは、
`(JSONArray).getJSONObject(i);`
 - このJSONObjectを利用してまた次の情報を得る
- 値自体の取得
 - `(JSONObject).getFloat("name");`
 - `(JSONObject).getInt("name");`
 - `(JSONObject).getString("name");`

天気の情報を取得しよう

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



• 東京の天気予報データを取得してみよう

<https://weather.tsukumijima.net/api/forecast/city/130010>

```
{
  "publicTime": "2023-11-19T17:00:00+09:00",
  "publicTimeFormatted": "2023/11/19 17:00:00",
  "publishingOffice": "気象庁",
  "title": "東京都 東京 の天気",
  "link": "https://www.jma.go.jp/bosai/forecast/#area_type=offices&area_code=130000",
  "description": {
    "publicTime": "2023-11-19T16:33:00+09:00",
    "publicTimeFormatted": "2023/11/19 16:33:00",
    "headlineText": "",
    "bodyText": "  華中には高気圧があって本州付近に張り出しています。\\n\\n  東京地方は、晴れています。\\n\\n  19日は、高気圧に覆われる見込みですが、次第に高気圧に覆われる見込みです。このため、晴れ時々曇りとなるでしょう。\\n\\n【関東甲信地方】\\n  関東甲信地方は、晴れや曇りとなる見込みです。このため、晴れや曇りでしょう。\\n\\n  20日は、前線が本州を南下し西高東低の気圧配置となりますが、次第に高気圧に覆われる見込みの降る所があり、雷を伴う所があるでしょう。\\n\\n  関東地方と伊豆諸島の海上では、うねりを伴い、19日は波が高く、20日はしけとなる見込みです。",
    "text": "  華中には高気圧があって本州付近に張り出しています。\\n\\n  東京地方は、晴れています。\\n\\n  19日は、高気圧に覆われる見込みですが、次第に高気圧に覆われる見込みです。このため、晴れ時々曇りとなるでしょう。\\n\\n【関東甲信地方】\\n  関東甲信地方は、晴れや曇りとなる見込みです。このため、晴れや曇りでしょう。\\n\\n  20日は、前線が本州を南下し西高東低の気圧配置となりますが、次第に高気圧に覆われる見込みの降る所があり、雷を伴う所があるでしょう。\\n\\n  関東地方と伊豆諸島の海上では、うねりを伴い、19日は波が高く、20日はしけとなる見込みです。船舶",
  },
  "forecasts": [
    {
      "date": "2023-11-19",
      "dateLabel": "今日",
      "telop": "晴れ",
      "detail": {
        "weather": "晴れ",
        "wind": "南の風",
        "wave": "0.5メートル"
      },
      "temperature": {
        "min": {
```

天気の情報を取得しよう



- 東京の天気予報データを取得してみよう

<https://weather.tsukumijima.net/api/forecast/city/130010>

```
String url = "https://weather.tsukumijima.net/api/forecast/city/130010";
JSONObject json = loadJSONObject( url );
JSONArray forecasts = json.getJSONArray("forecasts");
for(int i=0; i<forecasts.size(); i++){
    println( forecasts.getJSONObject(i).getString("date") );
    println( forecasts.getJSONObject(i).getString("telop") );
}
```



- ヒーローズリーグ（旧：Mashup Award）などでも色々活用されている
 - <https://heroes-league.net/>

何ができるか？



- 一般的なAPIはメソッドとして用意されており、そこに引数を渡すことで何かの動作を実現する
 - `ellipse(200, 200, 50, 50);`
 - `dist(mouseX, mouseY, 200, 200);`
- Web APIはGETリクエストであるURLに必要な情報を渡すことで何らかの結果を得る
 - `http://nkmr.io/api.php?person=homei`
 - `http://nkmr.io/api.php?x=50&y=30`



`http://snakamura.org/software/index.html`



プロトコル



サーバの場所



サーバ内での場所

- 使える文字は英数字と一部の記号
 - -.~:@!\$&'()
 - 日本語を入力する場合は%エンコーディング
- URI は URL と URN を総称したもの
 - URL は Uniform Resource Locator
 - URN は Uniform Resource Name

リクエストURL



`http://example.jp/search?query=test&area=10&...`

ベースURL

query=test

area=10

`http://example.jp/search`

ベースURL

`query=test`

query=test

`area=10`

area=10

...

ベースのURLのあと「？」が入り以降はオプション
複数のオプションは「&」でつなぐ
オプションは「=」で繋ぎ変数名と変数の値を指定

返り値はXMLやJSON



- 返り値はあるデータフォーマット
 - JSONやXMLなどの形式

XML

```
<staffs>
  <staff>
    <name>宮下芳明</name>
    <position>教授</position>
    <room>1018</room>
  </staff>
  <staff>
    <name>中村聡史</name>
    <position>教授</position>
    <room>1007</room>
  </staff>
</staffs>
```

JSON

```
{
  "staffs": {
    "staff": [
      {
        "name": "宮下芳明",
        "position": "教授",
        "room": "1018"
      },
      {
        "name": "中村聡史",
        "position": "教授",
        "room": "1007"
      }
    ]
  }
}
```

参考：Songle API



- OngaCRESTの成果（産総研後藤グループ）
 - <https://widget.songle.jp/docs/v1>
 - 音楽検索、音楽のコード進行、ビートパターンの取得など様々なことが可能に！
 - 例えばこんなのを作ることが可能
 - <https://justune.net/>

検索してみよう



- キーワード検索結果取得 XML/JSON
 - <https://widget.songle.jp/api/v1/songs/search.xml?q=keyword>
 - <https://widget.songle.jp/api/v1/songs/search.json?q=keyword>
- コード進行を取得してみよう！
- 音楽に合わせて何か動かしてみよう！

ChatGPTと対話しよう



- 配布する chatWithChatGPT.zip を解凍し、プログラムをまずは開こう
- apiKey にSlackに貼り付けるKeyをコピーしよう！
 - 120ドル上限をかけてるので、他の人が使うためにも無茶な使い方をしないでね！
- 下記の日本語の部分を差し替えて、showResponseで表示内容を切り替えよう！

```
chatWithChatGPT(apiKey, "日本語でジョークをお願いします");
```



DALLE3でも遊べるよ

- 配布する generateImageWithDalle3.zip を使ってみよう！
- 結果が JSON で返ってくるので、そのJSONから画像のURLを



- basic_loadXML
 - landprice.xml を利用して地価が設定されている場所を点で表現せよ





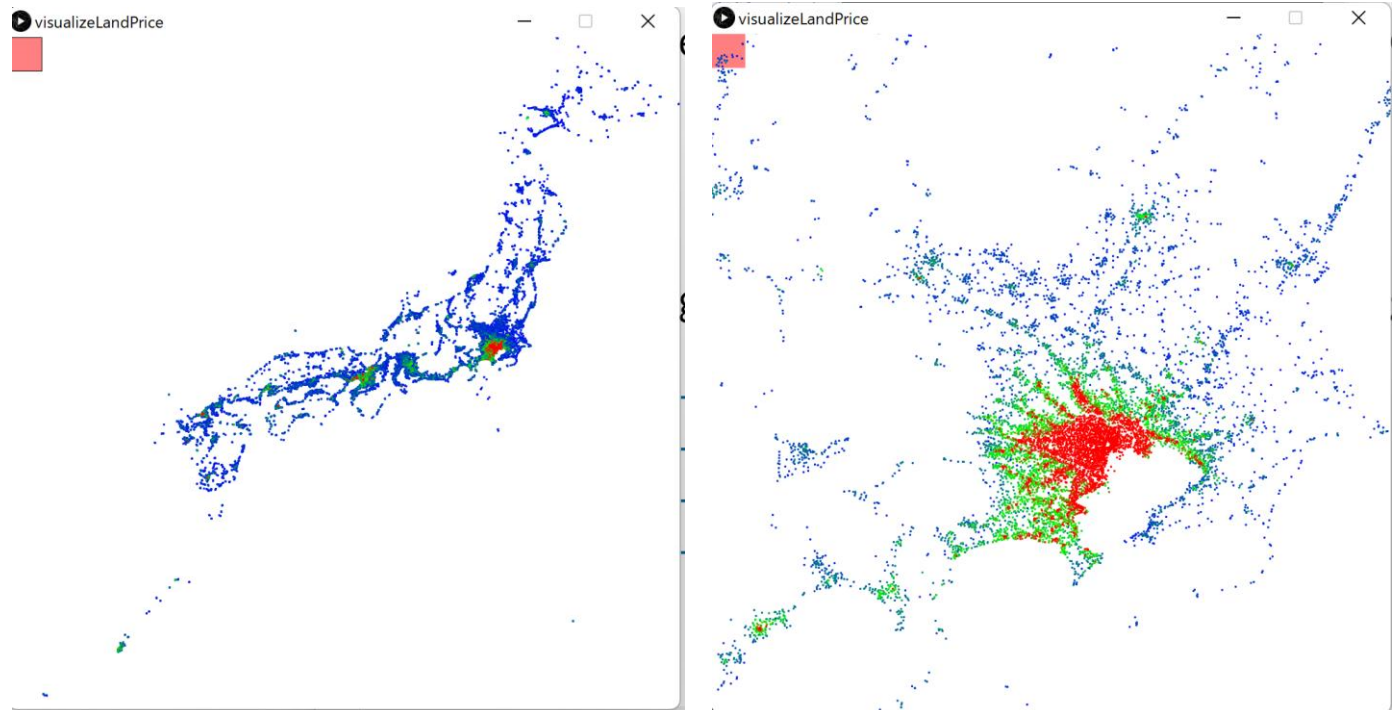
- `basic_loadJSON`
 - `landprice.json` を利用して地価が設定されている場所を点で表現せよ





- hw_showLandPrice

- JSON/XMLどちらかを使って地価の違いを色を用いて表現せよ
- もし余裕があれば拡大などもしてみよう！





- hw_sabaePopulation

- 鯖江の人口データを用いて、2003～2013年の経年変化をアニメーション表示せよ

