



プログラミング演習2

XMLとWeb API

中村, 小松, 小林, 鹿喰

XMLを学ぼう！



- 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データフォーマット
- 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(HTMLとの違い. 表示にはCSSなどを使用)

まずは見てみよう！

- 下記のXMLを開いてみよう！
 - <http://nkmr.io/lecture/2014/fms.xml>
 - http://nkmr.io/lecture/2014/fms_syllabus.xml

by Takuya Wada

```
<?xml version="1.0" encoding="UTF-8"?>
<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://homei.com/</link>
      <major>インタラクション</major>
      <room></room>
    </teacher>
  </teachers>
  <lecture>
    <name>エンタテイメントプログラミング演習</name>
    <year>1</year>
    <term>春</term>
  </lecture>
  <lecture>
    <name>コンテンツエンタテイメント概論</name>
    <year>1</year>
    <term>秋</term>
  </lecture>
  <lecture>
    <name>コンテンツメディアプログラミング実習2</name>
    <year>2</year>
    <term>秋</term>
  </lecture>
</lectures>
</teacher>
<teacher>
  <name>小松 孝徳</name>
  <link>http://www.tkomat-lab.com/</link>
  <major>認知心理学</major>
  <lecture>
    <year>1</year>
    <term>春</term>
  </lecture>
</teacher>
</data>
</xml>
```

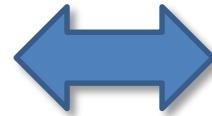


- データを機械可読な形に記述可能
 - HTMLは機械での認識が難しい
- 関係データベースで表現できない半構造データを扱うことが可能
 - 生物学のデータ
 - Web上の各種データ

HTMLとXMLの違い



```
<h1> 日本代表</h1>
<h2> GK </h2>
<ul> <li> 川島 </li> </ul>
<h2> DF </h2>
<ul>
<li> 中澤 </li> <li> 田中 </li>
<li> 長友 </li> <li> 駒野 </li>
</ul>
<h2> MF </h2>
<ul>
<li>遠藤 </li> <li> 阿部 </li>
<li>長谷部 </li> <li> 大久保 </li>
<li>松井 </li>
</ul>
<h2> FW </h2>
<ul> <li> 本田 </li> </ul>
<h2> 監督 </h2>
<ul> <li> 岡田 </li> </ul>
```



```
<Team>
<Name> 日本代表 </Name>
<GK>
<Player> 川島 </Player>
</GK>
<DF>
<Player> 中澤 </Player>
<Player> 田中 </Player>
<Player> 長友 </Player>
<Player> 駒野 </Player>
:
<Player> 大久保 </Player>
<Player> 松井 </Player>
</MF>
<FW>
<Player> 本田 </Player>
</FW>
<Director> 岡田 </Director>
</Team>
```



- 要素: XMLの1単位
 - `<team> ... </team>`, `<player> ... </player>`
 - 空要素にもなりうる: `<director></director>`
- タグ: `team`, `GK`, `MF`, ..., `player`, `director` など
- 開始タグ: `<team>`
- 終了タグ: `</team>`
- 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

様々な表現形式



```
<player position="GK" number="1" >
```

```
  榎崎正剛
```

```
</player>
```

```
<player>
```

```
<name>榎崎正剛</name>
```

```
<position>GK</position>
```

```
<number>1</number>
```

```
</player>
```

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( "http://nkmr.io/lecture/2014/fms.xml" );
```

```
XML [ ] children = xml.getChild("teachers").getChildren("teacher");  
for( int i=0; i<children.length; i++ ){  
    println( children[i].getChild("name").getContent() );  
}
```

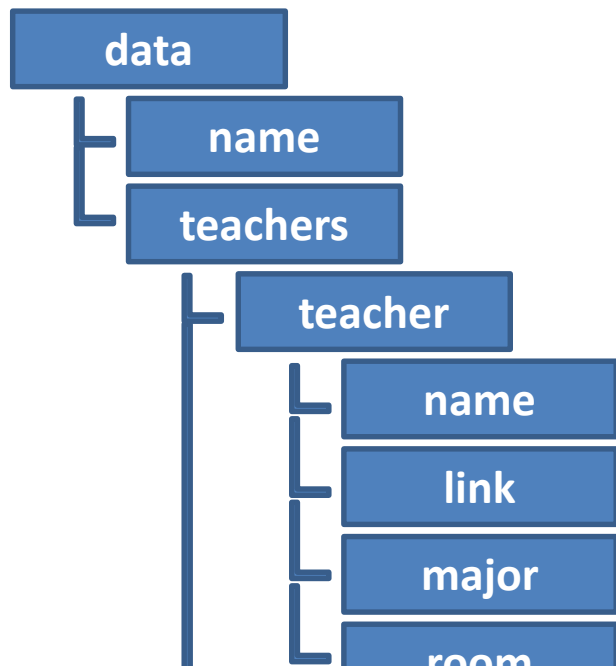
- loadXML で XML を読み込んで、getChildで子どもを取得して、getChildrenでその子どもたちを取得して、getContentで内容を表示！



- getParent() 親を取得
- getName() 要素の名前を取得
- hasChildren("name") nameの子達が居るかチェック
- getChildren("name") nameの子達のリストを取得
- getChild("name") nameの子を取得
- 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
- 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



- 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>インタラクシオン</major>
      <room>1017</room>
      <lectures>
        <lecture>エンタテイン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>

```

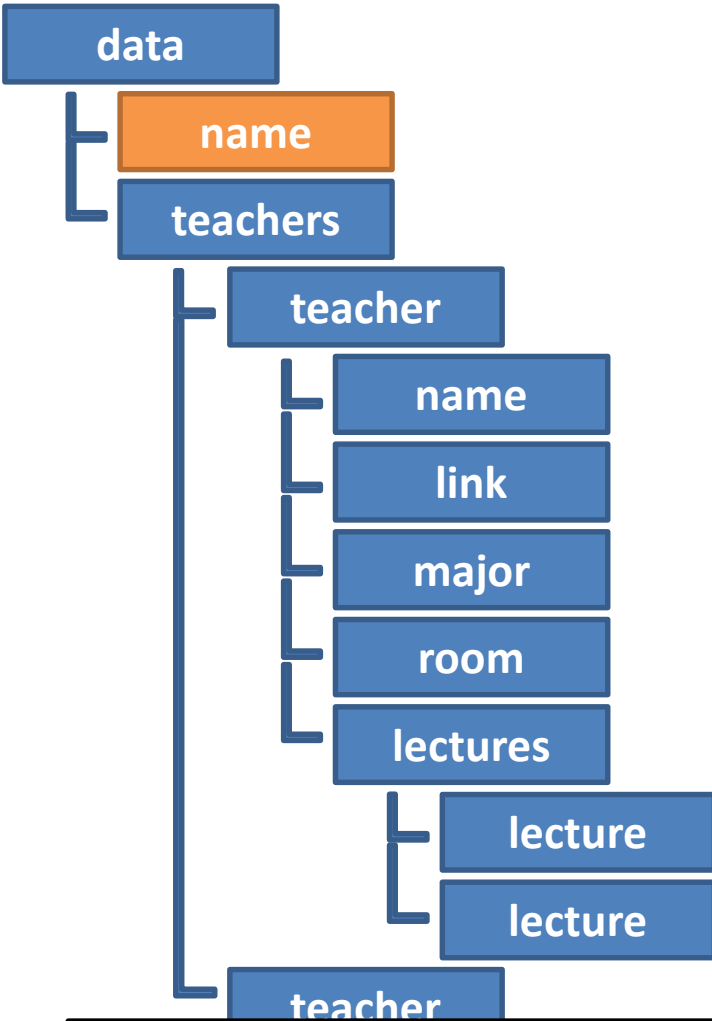
学科名を取得してみる



学科名はオレンジ色のname

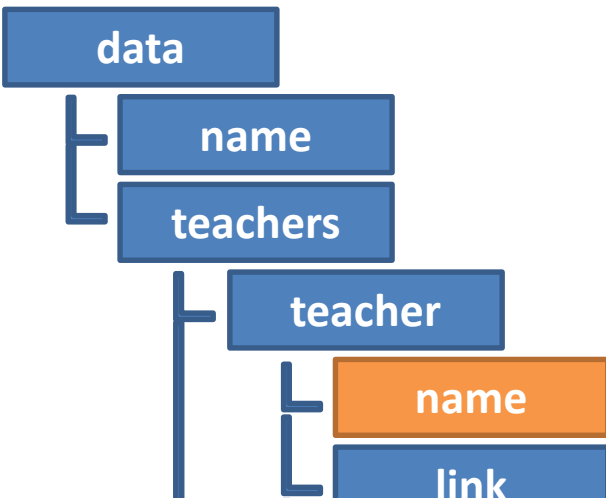
data → name

講義名は, dataの子



```
XML xml = loadXML( "http://nkmr.io/lecture/2014/fms.xml" );  
println( xml.getChild( "name" ).getContent() );
```

教員名を取得してみる

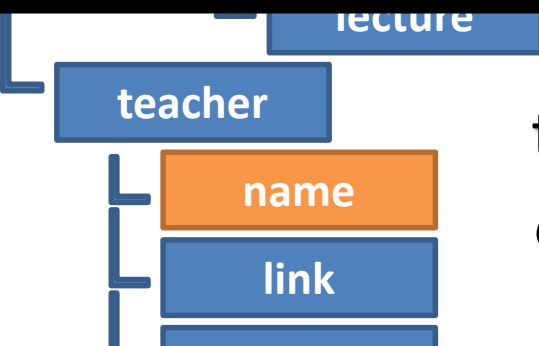


教員名はオレンジ色のname

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( "http://nkmr.io/lecture/2014/fms.xml" );
XML teachers = xml.getChild("teachers");
XML [] teacher = teachers.getChildlen("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

教員名を取得してみる



data

name

教員名はオレンジ色のname

```
XML xml = loadXML( "http://nkmr.io/lecture/2014/fms.xml" );
XML teachers = xml.getChild("teachers");
XML [] teacher = teachers.getChildlen("teacher");
for( int i=0; i<teacher.length; i++ ){
    println( teacher[i].getChild("name").getContent() );
}
```

major

```
XML xml = loadXML( "http://nkmr.io/lecture/2014/fms.xml" );
XML [] teacher = xml.getChild("teachers").getChildlen("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

講義リストを取得してみる



data

name

teachers

teacher

name

link

major

room

lectures

lecture

lecture

teacher

name

link

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

講義リストを取得してみる



```
XML xml = loadXML( "http://nkmr.io/lecture/2014/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() );
```

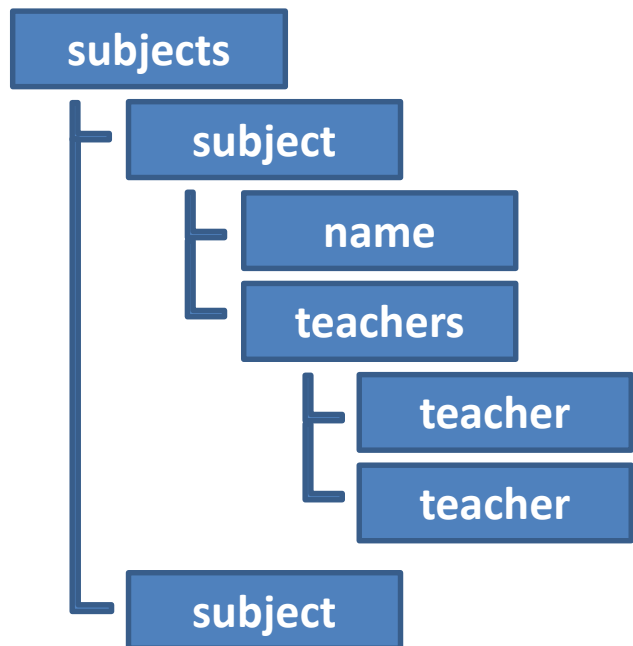
major

```
XML xml = loadXML( "http://nkmr.io/lecture/2014/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() );
    }
}
```

教
取
得

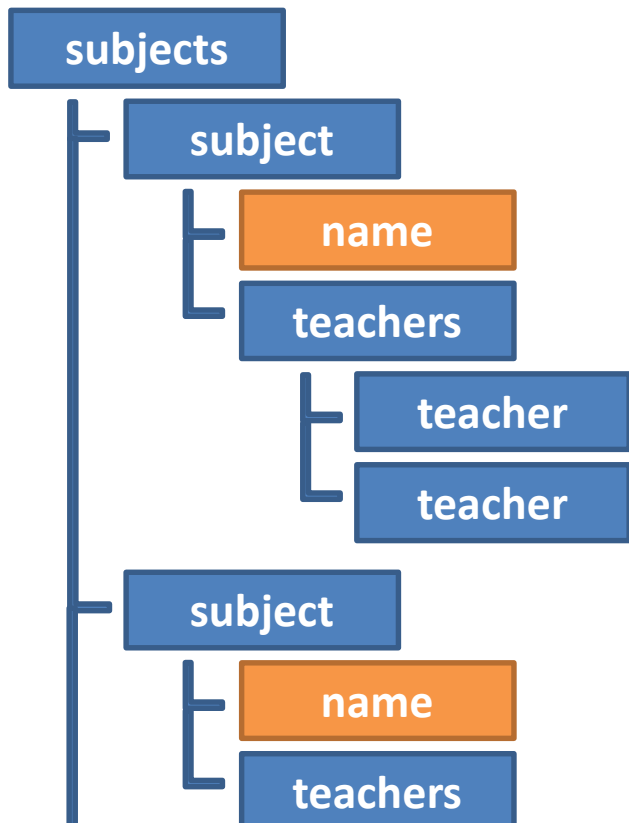
演習

- fms_syllabus.xml について下記に挑戦してみよう
 - 講義名リスト
 - 教員リスト
 - 講義名と教員のリスト



```
<subjects>
  <subject>
    <name>English1A(a,d,h)</name>
    <teachers>
      <teacher>河野円</teacher>
      <teacher>柴崎礼士郎</teacher>
    </teachers>
    <plan>
      <p>オリエンテーションとニーズアセスメント</p>
      <p>Chapter1おおよび速読スキル</p>
      <p>Chapter2おおよび精読スキル</p>
      <p>Chapter2</p>
      <p>Chapter3</p>
      <p>Chapter4おおよびライティング演習1</p>
      <p>Chapter5</p>
      <p>Chapter1から5のまとめ</p>
      <p>英文E-Mailの書き方</p>
      <p>Chapter6</p>
      <p>Chapter7</p>
      <p>Chapter8</p>
      <p>Chapter9おおよびライティング演習</p>
      <p>Chapter10</p>
      <p>Chapter6から10のまとめ</p>
    </plan>
    <text>World of Wonders Shaping Tomorrow</text>
  </reference>
  <evaluation>
    <e>出席と授業参加度10%</e>
    <e>語彙学習と小テスト20%</e>
    <e>提出物(作文やレポート)30%</e>
    <e>試験40%</e>
  </evaluation>
</subject>
  <subject>
    <name>English1A(b,c,e,f,g,i,j)</name>
    <teachers>
      <teacher>河野円</teacher>
      <teacher>柴崎礼士郎</teacher>
      <teacher>友田路</teacher>
      <teacher>原田依子</teacher>
    </teachers>
    <plan>
      <p>オリエンテーションとニーズアセスメント</p>
      <p>Chapter1おおよび速読スキル1</p>
      <p>Chapter1おおよび速読スキル2</p>
      <p>Chapter2おおよび精読スキル1</p>
      <p>Chapter2おおよび精読スキル2</p>
      <p>Chapter3おおよびライティングの基礎</p>
      <p>Chapter3おおよびライティング演習1</p>
      <p>Chapter1から2のまとめ</p>
    </plan>
  </subject>
</subjects>
```

講義名を取得してみる



講義名はオレンジ色のname

subjects → subject → name

講義名は、subjectsの子である
subjectの子に該当する

subject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

```
XML xml = loadXML( "http://nkmr.io/lecture/2014/fms_syllabus.xml" );
XML [] subject = xml.getChildren("subject");
println( subject[0].getChild("name").getContent() );
println( subject[1].getChild("name").getContent() );
```

:

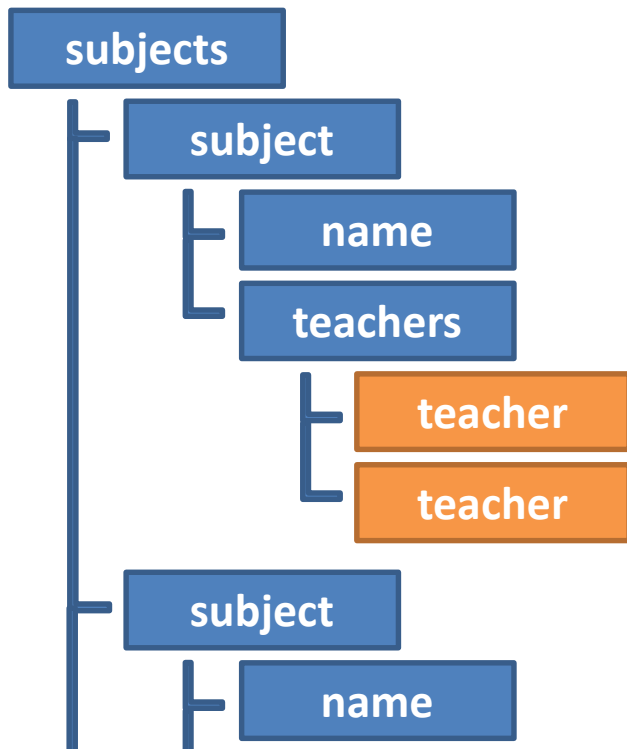
教員名を取得してみる



教員名はオレンジ色のteacher

subjects → subject

→ teachers → **teacher**



```
XML xml = loadXML( "http://nkmr.io/lecture/2014/fms_syllabus.xml" );
XML [] subject = xml.getChildren("subject");
XML teachers = subject[0].getChild( "teachers" );
XML [] teacher = teachers.getChildren( "teacher" );
println( teacher[0].getContent() );
println( teacher[1].getContent() );
```



- loadXMLをdraw()内部に書くのはやめましょう！
 - loadXMLでファイルをダウンロードして利用するものを利用している時, draw()の度にこの処理を行ってしまうと, 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

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



Weather Hacks
ウェザーハックス

- Weather Hack

- http://weather.livedoor.com/weather_hacks/

- 家の近辺の天気予報を表示してみよう！

- RSSは基本的にXMLと同じと解釈しておいてOK

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



 livedoor 天気情報 livedoorが提供するお天気情報

[ヘルプ](#) | [livedoor](#)

天気予報 | 災害情報

[天気予報](#) | [スポット天気](#) | [Weather Hacks](#)


[トップ](#) > [Weather Hacks](#)




Weather Hacks

ウェザーハックス

Weather Hacks (気象データ配信サービス)

 お天気Hack, してみませんか?

livedoor 天気情報では 第三者向け気象データ提供サービス「Weather Hacks」では、WebサービスとRSSフィード、ブログ向けお天気プラグイン、iCalフィードなどのデータを提供しています。

 iCal天気

[[iCal天気詳細](#)]

Weather Hacksでは、iCal方式での今日・明日・週間の天気予報フィードデータの配信を開始しました。iCalデスクトップカレンダーやGoogle Calendarで、お天気チェックしちゃいましょう！ [詳細はこちら](#)

 iCalって何?

iCalとは主にデスクトップカレンダーのスケジュール登録などで利用されているフォーマットです。(仕様は [こちら](#))

appleの [iCalデスクトップカレンダー](#) や [Google Calendar](#) で同フォーマットが採用されています。
(iCal天気を利用した 個別のアプリケーションでの設定についてはお答えできません。あらかじめご了承ください。)

メニュー

- ・ [お天気プラグイン](#)
- ・ [iCal天気](#)
- ・ [RSSフィード一覧](#)
- ・ [お天気Webサービス仕様](#)
- ・ [Weather Hacks用語集](#)
- ・ [Weather Hacks Q&A](#)



livedoorが提供するRSSフィードデータは、予告なく変更・配信を終了する場合があります。あらかじめご了承ください。

[免責事項](#)

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

<http://weather.livedoor.com/forecast/rss/area/130010.xml>

```
String url = "http://weather.livedoor.com/forecast/rss/area/130010.xml";
XML xml = loadXML( url );
XML [] item = xml.getChild("channel").getChildren("item");
for( int i=1; i<item.length; i++ ){
    println( item[i].getChild("title").getContent() );
}
```

[演習]

- 東京以外のどこかの天気予報データを取得しよう

文字列処理と組み合わせる

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



- 最高気温と最低気温をどう取得する？

30日(日)の天気は晴のち曇、最高気温は17°C 最低気温は12°Cでしょう。

– indexOfで場所を指定して，substringで出力

```
String url = "http://weather.livedoor.com/forecast/rss/area/130010.xml";
XML xml = loadXML( url );
XML [] item = xml.getChild("channel").getChildren("item");

for( int i=1; i<item.length; i++ ){
    println( item[i].getChild("description").getContent() );
    String str = item[i].getChild("description").getContent();
    println( str.substring( str.indexOf("最高気温")+5, str.indexOf("°C") ) );
    println( str.substring( str.indexOf("最低気温")+5,
        str.indexOf("°C", str.indexOf("最低気温")) ) );
}
```

[演習] 動画のランキング作成



- デイリーランキングの結果を表示してみよう！
- ニコニコ動画のXML
 - デイリーランキング
 - <http://www.nicovideo.jp/ranking/fav/daily/all>
 - ニコニコ動画API
 - <http://dic.nicovideo.jp/a/ニコニコ動画api>
 - ニコニコ動画のRSS一覧
 - <http://www.nicovideo.jp/rss/>

ニコニコ動画APIいくつか



- getthumbinfo
 - http://ext.nicovideo.jp/api/getthumbinfo/sm*
 - 動画の情報を得ることができる
 - <http://ext.nicovideo.jp/api/getthumbinfo/sm1097445>
- getflv(要ログイン)
 - http://flapi.nicovideo.jp/api/getflv/sm*
 - 指定された動画のFLV保管URLを取得できる
 - threadの情報を取得するのにも利用する
 - <http://flapi.nicovideo.jp/api/getflv/sm1097445>



- JavaScript Object Notation
 - JavaScript用のデータ交換フォーマット
 - 少し特殊なフォーマットを持つ
 - 詳細はCMP2で紹介しますが, XMLと同じように `loadJSONObject()`, `getString()`, `getInt()`, `getFloat()`, `getJSONArray()`, `getJSONObject()` などがあり, データを取得することが可能
 - JSONは
 - { 要素名: 要素の値 } という形で定義
(要素の中に要素を入れることが可能)
 - [要素1, 要素2] という形で配列を定義



- http://nkmr.io/lecture/2014/fms_syllabus.json
を読み込んでみよう
 - 読み込むのは `loadJSONObject("...");`

```
String url = "http://nkmr.io/lecture/2014/fms_syllabus.json";  
JSONObject json = loadJSONObject( url );  
println( json );
```

- 講義名リストを作成してみよう
- 講義名 + 教員名リストを作成してみよう

```
String url = "http://nkmr.io/lecture/2014/fms_syllabus.json";
JSONObject json = loadJSONObject( url );
println( json );
JSONArray subjects = json.getJSONArray( "subject" );
for( int i=0; i<subjects.size(); i++ ){
    JSONObject subject = subjects.getJSONObject(i);
    println( subject.getString("name") );
}
```

```
String url = "http://nkmr.io/lecture/2014/fms_syllabus.json";
JSONObject json = loadJSONObject( url );
println( json );
JSONArray arraySubject = json.getJSONArray( "subject" );
for( int i=0; i<arraySubject.size(); i++ ){
    JSONObject subject = arraySubject.getJSONObject(i);
    println( subject.getString("name") );
    JSONObject jsonTeachers = subject.getJSONObject("teachers");
    JSONArray arrayTeacher = jsonTeachers.getJSONArray("teacher");
    for( int j=0; j<arrayTeacher.size(); j++ ){
        println( " - " + arrayTeacher.getString(j) );
    }
}
```




- 例えばこんな感じ

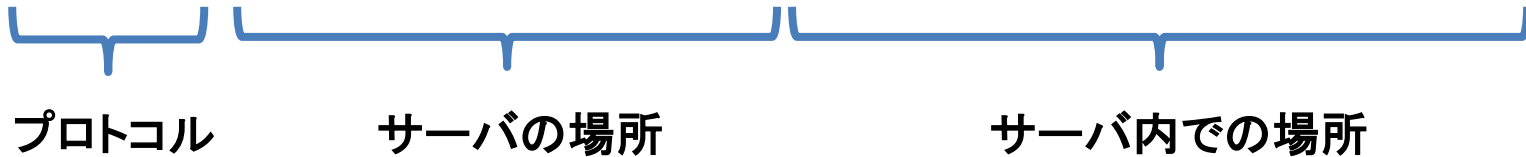
- <http://ma9.mashupaward.jp/apis>

- <http://www.find-job.net/startup/api-2013>





<http://snakamura.org/software/index.html>



- 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

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

XML

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

JSON

何ができるか？

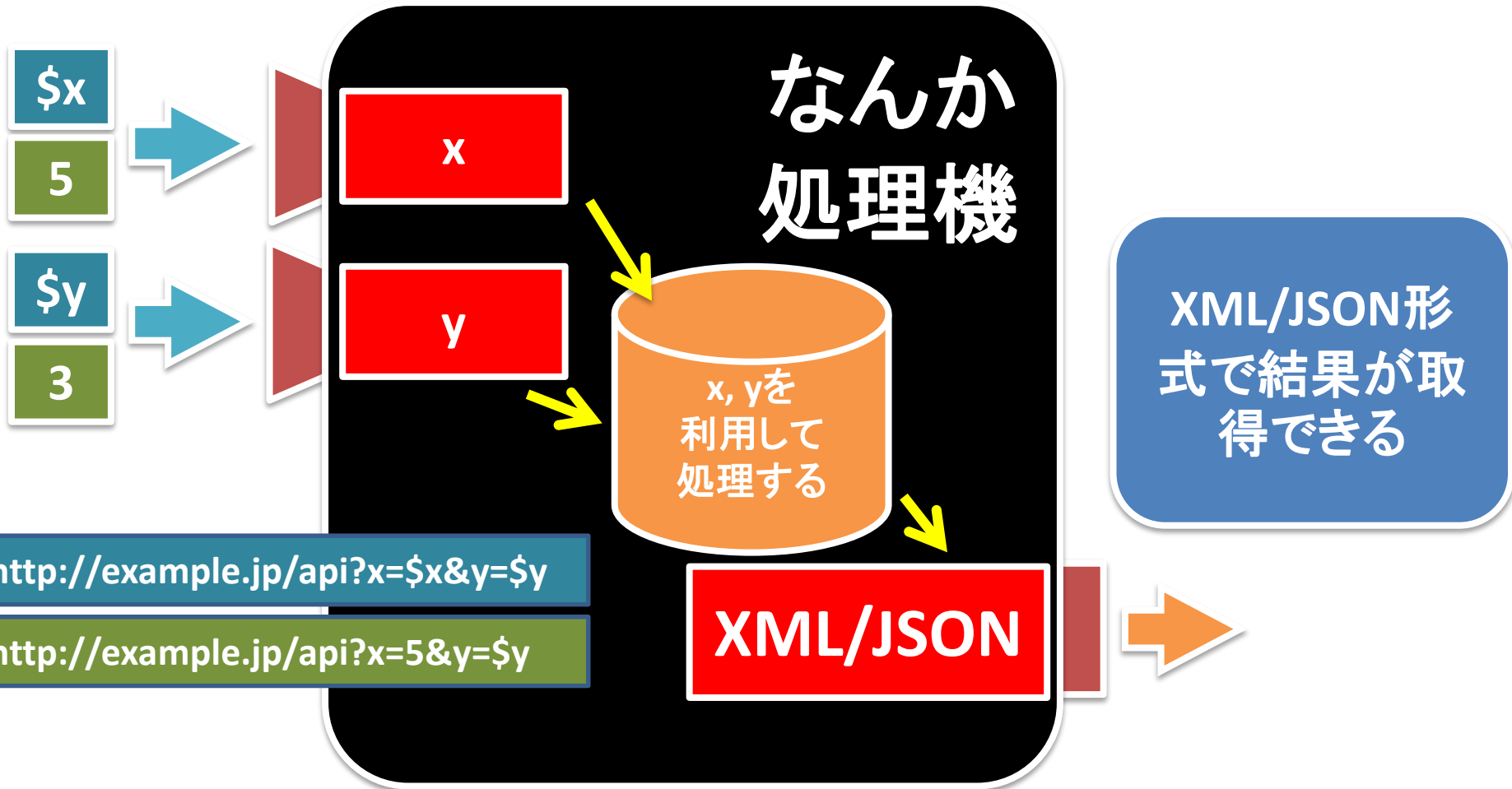


- 一般的なAPIはメソッドとして用意されており, そこに引数を渡すことで何かの動作を実現する
 - `ellipse(200, 200, 50, 50);`
 - `dist(mouseX, mouseY, 200, 200);`
- Web APIはGETリクエストであるURLに必要な情報を渡すことで何らかの結果を得る
 - `http://nkmr.io/api.php?person=homei`
 - `http://nkmr.io/api.php?x=50&y=30`

Web APIの内部処理



- 複数の引数を受け取ることが可能



[演習] フォト蔵APIを使う



- フォト蔵とは、インターネット上の画像を集め、検索可能とするサービス
 - https://api.photozou.jp/rest/search_public.xml?key=word=cat

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rsp stat="ok">
  <info>
    <photo_num>100</photo_num>
    <photo>
      <photo_id>215118418</photo_id>
      <user_id>2760264</user_id>
      <album_id>7115355</album_id>
      <photo_title>@mikumikul230...</photo_title>
      <favorite_num>0</favorite_num>
      <comment_num>0</comment_num>
      <view_num>0</view_num>
      <copyright>normal</copyright>
      <original_height>640</original_height>
      <original_width>480</original_width>
      <date>2014-11-30</date>
      <regist_time>2014-11-30T01:15:14+09:00</regist_time>
      <url>http://photozou.jp/photo/show/2760264/215118418</url>
      <image_url>
        http://art21.photozou.jp/pub/264/2760264/photo/215118418.jpg
      </image_url>
      <original_image_url>
        http://art21.photozou.jp/pub/264/2760264/photo/215118418_org.jpg
      </original_image_url>
      <thumbnail_image_url>
        http://art21.photozou.jp/pub/264/2760264/photo/215118418_thumbnail.jpg
      </thumbnail_image_url>
      <large_tag>
        <![CDATA[
          <a href="http://photozou.jp/photo/show/2760264/215118418"></a><br><a href="http://photozou.jp/photo/show/2760264/215118418">@mikumikul230...</a> posted by <a
              href="http://photozou.jp/user/top/2760264">(C)1122_maki</a>
          </large_tag>
        </medium_tag>
        <![CDATA[
          <a href="http://photozou.jp/photo/show/2760264/215118418"></a><br><a href="http://photozou.jp/photo/show/2760264/215118418">@mikumikul230...</a> posted by <a
              href="http://photozou.jp/user/top/2760264">(C)1122_maki</a>
          </medium_tag>
        </photo>
      </photo>
      <photo>
        <photo_id>215117670</photo_id>
        <user_id>197452</user_id>
        <album_id>8134717</album_id>
        <photo_title>ニャンですかぁ〜！？</photo_title>
```

[演習] フォト蔵APIを使う



[https://api.photozou.jp/rest/search_public.xml?](https://api.photozou.jp/rest/search_public.xml?keyword=ramen)

問い合わせの基本的なURL

keyword=ramen

オプション(この場合はキーワードにramenを指定している)

結果を表示して遊ぼう！



- 検索結果の画像を表示してみよう！（10枚）

```
PImage [] img;
void setup() {
  size( 1200, 800 );
  String url = "https://api.photozou.jp/rest/search_public.xml?keyword=ramen";
  XML xml = loadXML( url );
  XML [] photo = xml.getChild("info").getChildren("photo");
  img = new PImage [photo.length];
  for ( int i=0; i<10; i++ ) {
    println( photo[i].getChild("image_url").getContent() );
    img[i] = loadImage(photo[i].getChild("image_url").getContent() );
  }
  background( 0 );
  for( int i=0; i<10; i++ ){
    image( img[i], random(width-300),random(height-300) );
  }
}
void draw(){
}
```

日本語で検索するには...



- 右記のプログラムを別タブで作成
- urlのところを変更
+URLEncode("猫");

```
import java.io.UnsupportedEncodingException;
String URLEncode(String string) {
    String output = new String();
    try {
        byte[] input = string.getBytes("UTF-8");
        for (int i=0; i<input.length; i++) {
            if (input[i]<0){ output += '%' + hex(input[i]); }
            else if (input[i]==32){ output += '+'; }
            else { output += char(input[i]); }
        }
    } catch(UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    return output;
}
```

結果を表示して遊ぼう！



・ 猫というキーワードで検索

// ここに前のページの定義を追加～

```
PImage [] img;
void setup() {
  size( 1200, 800 );
  String url = "https://api.photozou.jp/rest/search_public.xml?keyword="+urlencode("猫");
  XML xml = loadXML( url );
  XML [] photo = xml.getChild("info").getChildren("photo");
  img = new PImage [photo.length];
  for ( int i=0; i<10; i++ ) {
    println( photo[i].getChild("image_url").getContent() );
    img[i] = loadImage(photo[i].getChild("image_url").getContent() );
  }
  background( 0 );
  for( int i=0; i<10; i++ ){
    image( img[i], random(width-300),random(height-300) );
  }
}

void draw(){
}
```

日本語はそのまま処理できないので%XXの形に要変換



- 適当なWeb APIまたはRSSなどを探し，そこから情報を取得して，情報提示するようなシステムを作ってみよう！

－（例）

- 1週間の最高気温，最低気温をグラフで表示
- 現在のニコニコ動画のランキングをサムネイル画像とセットで表示
- Yahoo! オークションの商品をランダムに表示

Yahoo! APIを使ってみよう

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



developer.yahoo.co.jp

YAHOO! JAPAN デベロッパーネットワーク IDでもっと便利に新規取得
ログイン

Yahoo! JAPAN ヘルプ

子ども社会の「大人不在」が、あらゆる子どもの問題を生む

アプリケーションの管理 サービス一覧 サポート ブログ

お知らせ ショッピング ボックス オークション YConnect アプリ販売

50本のAPIをフル公開!

出品APIをつかって、ビジネスをはじめよう
Yahoo! ショッピングAPI連携デベロッパー募集!

詳細はこちら >

いいね! シェア 263人が「いいね!」と言っています。「いいね!」をクリックして、友達に知らせましょう。

ツイート 134

Yahoo!デベロッパーネットワークを活用しよう

- 1 探そう 使いたいAPIやサービスを探そう
- 2 作ろう アプリケーションIDを登録して開発しよう
- 3 つながろう 開発者同士のコミュニティを活用しよう

ブログエントリー 一覧を見る

2014年5月26日 2014年5月19日 2014年5月13日

法人の方へ
WebAPIの商用利用について

Yahoo!サービス
Yahoo!プロモーション広告
ヤフーの検索結果に広告掲載
Yahoo!ウェブオーナーセンター
広告を掲載してサイトの収益アップ!
oo!ビジネスセンター
サイトビジネスをサポート

<http://developer.yahoo.co.jp/>

Yahoo! Japan IDでログイン

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



YAHOO! JAPAN デベロッパーネットワーク IDでもっと便利に[新規取得](#)
[ログイン](#)

[Yahoo! JAPAN - ヘルプ](#)

[デベロッパーネットワークトップ](#) > [アプリケーションの管理](#)

アプリケーションの管理

ログイン

Yahoo! JAPAN Web APIのご利用には、アプリケーションを登録し[アプリケーションID](#)を発行する必要があります。

アプリケーションIDは、ア
Yahoo! JAPAN IDに対し

アプリケーションIDは、ユ
為を防ぐために使用され

で、複数のアプリケーションに固有のIDには、アプリケーションIDに互換性がない。

アプリケーションIDは、第三者に開示したり、第三者が参照できる場所に記載しないでください。

**必要に応じて
アカウントを取得しよう！**

続けるには、Yahoo! JAPAN IDでログインする必要があります。

[ログイン](#)

Yahoo! JAPAN IDをお持ちでない方は[こちら](#)

[>>Yahoo! JAPAN IDを取得\(無料\)](#)

※ Yahoo! JAPANにおける個人情報の取り扱いについては、[プライバシーポリシー](#)をご覧ください。

 [ご利用ガイド](#)

Web APIとその他のサービス

API

[▶ ボックス](#)

[▶ ショッピング](#)

[▶ オークション](#)

[▶ YOLP\(地図\)](#)

[▶ テキスト解析](#)

[▶ 知恵袋](#)

[▶ ニュース](#)

[▶ カテゴリ](#)

[▶ 震災関連情報](#)

[▶ Social API](#)

その他

[▶ YConnect](#)

[▶ YUI](#)

[▶ マーケット](#)

[▶ メール](#)

新しいアプリケーションを開発

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



YAHOO! JAPAN デベロッパーネットワーク ようこそ、todoviewer さん
[ログアウト](#)

[Yahoo! JAPAN](#) - [ヘルプ](#)

[デベロッパーネットワークトップ](#) > [アプリケーションの管理](#)

アプリケーションの管理

[ご利用ガイド](#)

アプリケーション一覧

[新しいアプリケーションを開発](#)

[Show Bar \(サーバーサイド\)](#) 作成日: 2014年5月28日
[編集](#) [削除](#)
[利用状況](#)

[test](#) 作成日: 2012年7月24日
[編集](#) [削除](#)
[利用状況](#)

[SyncRerank](#) 作成日: 2009年10月22日
[編集](#) [削除](#)
[利用状況](#)

作成日: 2007年9月27日
[編集](#) [削除](#)
[利用状況](#)

Web APIとその他のサービス

API

- [ボックス](#)
- [ショッピング](#)
- [オークション](#)
- [YOLP\(地図\)](#)
- [テキスト解析](#)
- [知恵袋](#)
- [ニュース](#)
- [カテゴリ](#)
- [震災関連情報](#)
- [Social API](#)

その他

- [YConnect](#)
- [YUI](#)
- [マーケット](#)
- [メール](#)
- [OAuth1.0](#)



必要な情報を入力！

* アプリケーションの種類

サーバーサイド

サーバー上など、秘密鍵を安全に保管できる場所からWeb APIを使いたい場合に選択してください。OAuth 2.0 Authorization Codeフローによるユーザー属性情報が利用できます。

クライアントサイド

スマートフォンのネイティブアプリなど、クライアントアプリから直接Web APIを使いたい場合に選択してください。OAuth 2.0 Implicitフローによるユーザー属性情報が利用できます。

※テキスト

クライアントサイドを選択

しても利用でき

アプリケーションの基本情報

Yahoo! JAPAN ID

todoviewer

* 連絡先メールアドレス

nakamura.satoshi@gmail.com

[メールアドレスを追加・編集する](#)

Yahoo! JAPAN IDの登録情報で確認済みの登録メールアドレスと、Yahoo!メールアドレスから選択できます。連絡先メールアドレスは、Yahoo!デベロッパーネットワークから重要なお知らせなどをお送りするためのメールアドレスとなります。

* アプリケーション名
あと40文字入力できます

processing



(50文字まで)

開発

アプリケーション名はお好きにどうぞ

画面にて表示される

アプリケーション名になります。

サイトURL
あと181文字入力できます

http://example.com/



(半角英数字200文字まで)

利用者同意画面にて表示されます。

サイトURLはそのままOK

アプリケーションの説明

アプリケーションIDを取得

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



YAHOO! デベロッパーネットワーク ようこそ, todoviewer さん
JAPAN ログアウト

Yahoo! JAPAN

デベロッパーネットワークトップ > アプリケーションの管理 > 登録完了

新しいアプリケーションを開発

登録完了

登録が完了し、アプリケーションIDおよびシークレットが発行されました。

アプリケーションID:

`dj0zaizpPU45R29mZjYidUhtNiZzPWNvbnN1bWYyc2VjcmY0Jng9NDI-`

シークレット:

このアプリケーションIDをコピーして利用！

さっそくアプリケーションを作ってみましょう！

1. ブラウザーから直接リクエストする

ショッピング商品検索APIで「テレビ」カテゴリ内の売れ筋商品を検索するには、以下のURLにアクセスします。

http://shopping.yahooapis.jp/ShoppingWebService/Y1/itemSearch?appid=dj0zaizpPU45R29mZjYidUhtNiZzPWNvbnN1bWYyc2VjcmY0Jng9NDI-&category_id=635&sort:

2. プログパーツにしてWebサイトに埋め込めるようにする

地図を表示するには？



アプリケーションの管理

サービス一覧▼

サポート▼

ブログ

デベロッパーネットワークトップ > YOLP(地図)

YOLP(地図)

Yahoo! Open Local Platformで、地図や地域情報サイトを開発

[Yahoo! Open Local Platform\(YOLP\)](#)は、Yahoo! JAPANがデベロッパー向けに提供する地図・地域情報のAPI・SDKです。ウェブページの制作はもちろん、スマートフォン向けアプリケーションの開発においても、多彩な地図表示、店舗・施設検索、ジオコーディング、ルート検索、標高地取得などの豊富な機能をご活用いただけます。

お知らせ

2014年4月8日 [Yahoo!ローカルサーチAPIのレスポンス変更のお知らせ](#)

2014年4月2日 [【重要なお知らせ】「YOLPカセットギャラリー」サービス終了のお知らせ](#)

2014年3月17日 [【重要なお知らせ】YOLP\(地図\)旧APIおよびSilverlightマップAPI,SDK サービス終了について](#)

ウェブサービスで地図を使う

Yahoo! JavaScriptマップAPI NEW!

マウスでドラッグできる地図をウェブページに自由に貼り付けて利用できます。雨雲レーダー情報を重ねて表示できます。

Yahoo! スタティックマップAPI NEW!

地図の画像(PNG形式やJPEG形式など)をウェブページに自由に貼り付けて利用できます。

はじめよう

アプリケーションを開発

法人の方へ

WebAPIの商用利用について

ツイート 104

いいね! 148

開発のヒント

- Web APIの使い方(GET)
- サンプルコード集
- Web APIライブラリ集
- 爆速JSONP
- ドキュメント
- 開発者コミュニティ
- Web APIを利用したサービス
- Yahoo! JAPANが提供するRSS一覧
- クレジットの表示
- 用語集
- エラーメッセージおよびコード
- 利用制限について

▼開発のヒント



APIキーは自分のを利用しましょう！

```
PImage mapImage;  
void setup() {  
  size( 600, 400 );  
  String baseURL = "http://map.olp.yahooapis.jp/OpenLocalPlatform/V1/static?";  
  String option = "appid=dj0zaiZpPWljVIFVRW85TWlnSyZzPWNvbnN1bWVyc2VjcmV0Jng9Njg-";  
  option += "&lat=35.706962&lon=139.659547&z=16&width=600&height=400&pointer=on";  
  mapImage = loadImage( baseURL + option, "png" );  
}  
  
void draw() {  
  image( mapImage, 0, 0 );  
}
```

```

import java.io.UnsupportedEncodingException;
String URLEncode(String string) {
    String output = new String();
    try {
        byte[] input = string.getBytes("UTF-8");
        for (int i=0; i<input.length; i++) {
            if (input[i]<0){ output += '%' + hex(input[i]); }
            else if (input[i]==32){ output += '+'; }
            else { output += char(input[i]); }
        }
    } catch(UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    return output;
}

void setup() {
    String baseURL = "http://jlp.yahooapis.jp/MAService/V1/parse?";
    String option = "appid=dj0zaiZpPWljVIFVRW85TWlnSyZzPWNvbnN1bWVyc2VjcmV0Jng9Njg-";
    option += "&sentence=" + URLEncode( "庭には二羽鶏がいる" );
    XML xml = loadXML( baseURL + option );
    println( xml );
    XML [] word = xml.getChild("ma_result").getChild("word_list").getChildren("word");
    for( int i=0; i<word.length; i++ ){
        println( word[i].getChild("surface").getContent(), word[i].getChild("pos").getContent() );
    }
}
}

```