

 無料電子ブック

学習

Jsoup

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#jsoup

.....	1
<b>1: Jsoup</b> .....	<b>2</b>
.....	2
JavaScript.....	2
.....	2
.....	2
.....	3
Examples.....	3
URL.....	3
HTMLURL.....	3
HTML.....	4
<b>2: HTML</b> .....	<b>5</b>
.....	5
.....	5
Examples.....	5
.....	5
<b>3: Javascript</b> .....	<b>7</b>
Examples.....	7
JsoupHtmUnitJavaScript.....	7
<b>4: Jsoup</b> .....	<b>9</b>
Examples.....	9
JsoupPOST.....	9
JsoupPOST.....	9
FormElement.....	10
<b>5: JsoupWeb</b> .....	<b>12</b>
Examples.....	12
.....	12
JsoupJavaScript.....	12
JSoupURL.....	13
<b>6:</b> .....	<b>15</b>
.....	15

Examples.....	16
CSS.....	16
Twitter.....	17
.....	<b>19</b>

---

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [jsoup](#)

It is an unofficial and free Jsoup ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Jsoup.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

# 1: Jsoupをいめる

Jsoupは、JavaのHTMLおよびデータライブラリであり、といやすさにをいています。これは、に「Webスクレイピング」とばれるHTMLページからのデータをしたり、HTMLページのをしたり、できないHTMLをタグやのホワイトリストで「クリーン」するためにできます。

## JavaScriptサポート

JsoupはJavaScriptをサポートしていないため、ページロードにページにされるにされたコンテンツやコンテンツをページからすることはできません。JavaScriptをしてページにされるコンテンツをするがあるは、いくつかのがあります。

- のWebブラウザをしてページをロードするSeleniumやHtmlUnitなど、JavaScriptをサポートするライブラリをします。
- ページがどのようにデータをみむかをリバースエンジニアリングします。、AJAXをしてデータをにロードするWebページでは、ブラウザのデベロッパーツールのネットワークタブをして、データのロードをし、それらのURLをのコードですることができます。は、[AJAXページをスクラップするをしてください](#)。

## ウェブサイトとドキュメンテーション

[Jsoup.org](#)には、[Javadoc](#)、[Jsoupの](#)、[JARダウンロード](#)など、さまざまなJsoupリソースがあります。ソースコード、およびプルリクエストについては、[GitHubリポジトリ](#)をしてください。

## ダウンロード

Jsoupは`org.jsoup.jsoup:jsoup`としてMavenでできます。GradleをしているAndroidスタジオなど、`build.gradle`セクションにのものをすることでプロジェクトにできます。

```
compile 'org.jsoup:jsoup:1.8.3'
```

AntEclipseをしているは、POMセクションにをします。

```
<dependency>
  <!-- jsoup HTML parser library @ http://jsoup.org/ -->
  <groupId>org.jsoup</groupId>
  <artifactId>jsoup</artifactId>
  <version>1.8.3</version>
</dependency>
```

Jsoupは、ののダウンロードなJARとしてもできます。

## バージョン

バージョン	
1.9.2	2016517
1.8.3	2015-08-02

## Examples

リンクのURLとタイトルをする

Jsoupをすると、Webページからすべてのリンクをにできます。この、Jsoupをして、なのリンク、ここではページのh3ヘッダーのみをすることができます。リンクのテキストもできます。

```
Document doc = Jsoup.connect("http://stackoverflow.com").userAgent("Mozilla").get();
for (Element e: doc.select("a.question-hyperlink")) {
    System.out.println(e.attr("abs:href"));
    System.out.println(e.text());
    System.out.println();
}
```

これにより、のがられます。

```
http://stackoverflow.com/questions/12920296/past-5-week-calculation-in-webi-bo-4-0
Past 5 week calculation in WEBI (BO 4.0)?

http://stackoverflow.com/questions/36303701/how-to-get-information-about-the-visualized-
elements-in-listview
How to get information about the visualized elements in listview?

[...]
```

ここがこっているのですか

- まず、されたURLからHTMLをします。また、このコードでは、リクエストのUser Agentヘッダーを「Mozilla」にしているため、はブラウザにされるページがWebサイトにされます。
- に、select(...)とforループをselect(...)、Stack Overflowのへのリンクをすべてします。この、クラスのquestion-hyperlinkます。
- リンクのテキストを.text()でリンクのhrefをattr("abs:href")ます。この、abs:をしてURLをします。ドメインとプロトコルがまれています。

## HTMLからなURLをする

リンクののみをhrefはURLをします。

```
String bodyFragment =
    "<div><a href=\""/documentation\">Stack Overflow Documentation</a></div>";

Document doc = Jsoup.parseBodyFragment(bodyFragment);
String link = doc
    .select("div > a")
    .first()
    .attr("href");

System.out.println(link);
```

```
/documentation
```

`parse`メソッドにベースURIをし、`attr`わりに`absUrl`メソッドをすることで、なURLをすることができます。

```
Document doc = Jsoup.parseBodyFragment(bodyFragment, "http://stackoverflow.com");

String link = doc
    .select("div > a")
    .first()
    .absUrl("href");

System.out.println(link);
```

```
http://stackoverflow.com/documentation
```

## HTMLファイルからデータをする

`Jsoup`は、HTMLをローカルファイルからデータをまたはするためにできます。`filePath`はディスクのファイルのパスです。`ENCODING`がです。セット "Windows-31J"。オプションです。

```
// load file
File inputFile = new File(filePath);
// parse file as HTML document
Document doc = Jsoup.parse(filePath, ENCODING);
// select element by <a>
Elements elements = doc.select("a");
```

オンラインでJsoupをいめるをむ <https://riptutorial.com/ja/jsoup/topic/297/jsoupをいめる>

## 2: HTMLの

### パラメーター

パラメータ	
<code>boolean outline()</code>	アウトラインモードがになっているにします。デフォルトはfalseです。にすると、HTMLメソッドはすべてのタグをブロックとなします。
<code>Document.OutputSettings outline(boolean)</code>	HTMLアウトラインモードをまたはにします。

### Jsoup 1.9.2 API

## Examples

すべてのをブロックとしてする

デフォルトでは、Jsoupはブロックレベルのだけをにしてします。インラインはなしでされます。

インラインをつボディフラグメントをすると

```
<select name="menu">
  <option value="foo">foo</option>
  <option value="bar">bar</option>
</select>
```

### Jsoupでする

```
Document doc = Jsoup.parse(html);

System.out.println(doc.html());
```

```
<html>
  <head></head>
  <body>
    <select name="menu"> <option value="foo">foo</option> <option value="bar">bar</option>
  </select>
  </body>
</html>
```

をブロックとしてうようにをするには、の `OutputSettings outline` オプションをにする `OutputSettings` ます。



```
Document doc = Jsoup.parse(html);  
  
doc.outputSettings().outline(true);  
  
System.out.println(doc.html());
```

```
<html>  
  <head></head>  
  <body>  
    <select name="menu">  
      <option value="foo">foo</option>  
      <option value="bar">bar</option>  
    </select>  
  </body>  
</html>
```

ソース [JSoup - <option>の](#)

[オンラインでHTMLのをむ](https://riptutorial.com/ja/jsoup/topic/5954/html) <https://riptutorial.com/ja/jsoup/topic/5954/html>の

# 3: Javascriptでされたページの

## Examples

JsoupとHtmUnitでJavaScriptページをする

page.html - ソースコード

```
<html>
<head>
  <script src="loadData.js"></script>
</head>
<body onLoad="loadData()">
  <div class="container">
    <table id="data" border="1">
      <tr>
        <th>col1</th>
        <th>col2</th>
      </tr>
    </table>
  </div>
</body>
</html>
```

loadData.js

```
// append rows and cols to table.data in page.html
function loadData() {
  data = document.getElementById("data");
  for (var row = 0; row < 2; row++) {
    var tr = document.createElement("tr");
    for (var col = 0; col < 2; col++) {
      td = document.createElement("td");
      td.appendChild(document.createTextNode(row + "." + col));
      tr.appendChild(td);
    }
    data.appendChild(tr);
  }
}
```

page.html ブラウザにみまれたとき

Col1	Col2
0.0	0.1
1.0	1.1

jsoupをしてcolデータのpage.htmlをする

```
// load source from file
Document doc = Jsoup.parse(new File("page.html"), "UTF-8");

// iterate over row and col
for (Element row : doc.select("table#data > tbody > tr"))

    for (Element col : row.select("td"))

        // print results
        System.out.println(col.ownText());
```

の

がこった

Jsoupは、サーバからまたはこのはファイルからロードされてソースコードをします。JavaScriptやCSSのDOMなどのクライアントのアクションはびされません。このでは、とはしてデータテーブルにされません。

ブラウザでレンダリングされたページをするには

```
// load page using HTML Unit and fire scripts
WebClient webClient = new WebClient();
HtmlPage myPage = webClient.getPage(new File("page.html").toURI().toURL());

// convert page to generated HTML and convert to document
doc = Jsoup.parse(myPage.asXml());

// iterate row and col
for (Element row : doc.select("table#data > tbody > tr"))

    for (Element col : row.select("td"))

        // print results
        System.out.println(col.ownText());

// clean up resources
webClient.close();
```

```
0.0
0.1
1.0
1.1
```

オンラインでJavascriptでされたページのをむ <https://riptutorial.com/ja/jsoup/topic/4632/javascript>  
でされたページの

## 4: Jsoupでウェブサイトにログインする

### Examples

#### JsoupでのなPOSTリクエスト

データをしたなPOSTをにします。 `username`と`password`フィールドは、Webサイトによって異なります。

```
final String USER_AGENT = "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36";
Connection.Response loginResponse = Jsoup.connect("yourWebsite.com/loginUrl")
    .userAgent(USER_AGENT)
    .data("username", "yourUsername")
    .data("password", "yourPassword")
    .method(Method.POST)
    .execute();
```

#### JsoupによるよりなPOSTリクエスト

ほとんどのウェブサイトでは、のものよりもはるかになプロセスがです。

ウェブサイトにログインするためのなはのとおりでです。

1. のログインフォームからの`cookie`をします。
2. ログインフォームをべて、リクエストのURLをします
3. ログインフォームをして、ユーザーとパスワードとにするのある`security token`がないかどうかをします。
4. リクエストをします。

は[GitHub](https://github.com)のウェブサイトにログインするリクエストのです

```
// # Constants used in this example
final String USER_AGENT = "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36";
final String LOGIN_FORM_URL = "https://github.com/login";
final String LOGIN_ACTION_URL = "https://github.com/session";
final String USERNAME = "yourUsername";
final String PASSWORD = "yourPassword";

// # Go to login page and grab cookies sent by server
Connection.Response loginForm = Jsoup.connect(LOGIN_FORM_URL)
    .method(Connection.Method.GET)
    .userAgent(USER_AGENT)
    .execute();

Document loginDoc = loginForm.parse(); // this is the document containing response html
HashMap<String, String> cookies = new HashMap<>(loginForm.cookies()); // save the cookies to be passed on to next request

// # Prepare login credentials
```

```

String authToken = loginDoc.select("#login > form > div:nth-child(1) >
input[type=\"hidden\"]:nth-child(2)")
    .first()
    .attr("value");

HashMap<String, String> formData = new HashMap<>();
formData.put("commit", "Sign in");
formData.put("utf8", "e2 9c 93");
formData.put("login", USERNAME);
formData.put("password", PASSWORD);
formData.put("authenticity_token", authToken);

// # Now send the form for login
Connection.Response homePage = Jsoup.connect(LOGIN_ACTION_URL)
    .cookies(cookies)
    .data(formData)
    .method(Connection.Method.POST)
    .userAgent(USER_AGENT)
    .execute();

System.out.println(homePage.parse().html());

```

## FormElement をしたログイン

ここでは、[FormElement](#) クラスをして [GitHub Web](#) サイトにログインします。

```

// # Constants used in this example
final String USER_AGENT = "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/51.0.2704.103 Safari/537.36";
final String LOGIN_FORM_URL = "https://github.com/login";
final String USERNAME = "yourUsername";
final String PASSWORD = "yourPassword";

// # Go to login page
Connection.Response loginFormResponse = Jsoup.connect(LOGIN_FORM_URL)
    .method(Connection.Method.GET)
    .userAgent(USER_AGENT)
    .execute();

// # Fill the login form
// ## Find the form first...
FormElement loginForm = (FormElement)loginFormResponse.parse()
    .select("div#login > form").first();

checkElement("Login Form", loginForm);

// ## ... then "type" the username ...
Element loginField = loginForm.select("#login_field").first();
checkElement("Login Field", loginField);
loginField.val(USERNAME);

// ## ... and "type" the password
Element passwordField = loginForm.select("#password").first();
checkElement("Password Field", passwordField);
passwordField.val(PASSWORD);

// # Now send the form for login
Connection.Response loginActionResponse = loginForm.submit()
    .cookies(loginFormResponse.cookies())

```

```
        .userAgent (USER_AGENT)
        .execute ();

System.out.println(loginActionResponse.parse().html());

public static void checkElement (String name, Element elem) {
    if (elem == null) {
        throw new RuntimeException ("Unable to find " + name);
    }
}
}
```

すべてのフォームデータは、FormElementクラスによってされますフォームメソッドのさえも。みのConnectionは、FormElementsubmitメソッドをびすときにされます。addionalヘッダークッキー、ユーザーエージェントなどとのこのをしてするだけです。

オンラインでJsoupでウェブサイトにログインするをむ

<https://riptutorial.com/ja/jsoup/topic/4631/jsoupでウェブサイトにログインする>

# 5: JsoupによるWebクローリング

## Examples

メールアドレスとのページへのリンクをする

Jsoupは、Webページからリンクとメールアドレスをするためにすることができます。"Webメールアドレスコレクタボット"まず、このコードはをしてメールアドレスをし、Jsoupがするメソッドをして、ページ。

```
public class JSoupTest {

    public static void main(String[] args) throws IOException {
        Document doc =
Jsoup.connect("http://stackoverflow.com/questions/15893655/").userAgent("Mozilla").get();

        Pattern p = Pattern.compile("[a-zA-Z0-9_+]+@[a-zA-Z0-9-]+\\.([a-zA-Z0-9-]+)");
        Matcher matcher = p.matcher(doc.text());
        Set<String> emails = new HashSet<String>();
        while (matcher.find()) {
            emails.add(matcher.group());
        }

        Set<String> links = new HashSet<String>();

        Elements elements = doc.select("a[href]");
        for (Element e : elements) {
            links.add(e.attr("href"));
        }

        System.out.println(emails);
        System.out.println(links);

    }

}
```

このコードをにして、にそれらのURLにアクセスし、リンクされたページからデータをするともできます。また、のでにのデータをするともできます。

スパマーにならないでください

## JsoupでJavaScriptデータをする

このでは、 `backgroundColor: '#FFF'` をむJavaScriptデータをしします。その、々はの `backgroundColor` のがされます `'#FFF'` `'#ddd'`。このコードでは、 `getWholeData()` および `setWholeData()` メソッドをしてJavaScriptデータをしします。また、 `html()` メソッドをしてJavaScriptのデータをするともできます。

```
// create HTML with JavaScript data
```

```

StringBuilder html = new StringBuilder();
html.append("<!DOCTYPE html> <html> <head> <title>Hello Jsoup!</title>");
html.append("<script>");
html.append("StackExchange.docs.comments.init({");
html.append("highlightColor: '#F4A83D',");
html.append("backgroundColor:'#FFF',");
html.append("});");
html.append("</script>");
html.append("<script>");
html.append("document.write(<style type='text/css'>div,iframe { top: 0; position:absolute;
}</style>");
html.append("</script>\n");
html.append("</head><body></body> </html>");

// parse as HTML document
Document doc = Jsoup.parse(html.toString());

String defaultBackground = "backgroundColor:'#FFF'";
// get <script>
for (Element scripts : doc.getElementsByTag("script")) {
    // get data from <script>
    for (DataNode dataNode : scripts.dataNodes()) {
        // find data which contains backgroundColor:'#FFF'
        if (dataNode.getWholeData().contains(defaultBackground)) {
            // replace '#FFF' -> '#ddd'
            String newData = dataNode.getWholeData().replaceAll(defaultBackground,
"backgroundColor:'#ddd'");
            // set new data contents
            dataNode.setWholeData(newData);
        }
    }
}
System.out.println(doc.toString());

```

```

<script>StackExchange.docs.comments.init({highlightColor:
'#F4A83D',backgroundColor:'#ddd',});</script>

```

## JsoupをしてウェブサイトからすべてのURLをする

ここでは、WebサイトからすべてのWebリンクをします。はのために<http://stackoverflow.com/>をしています。ここではがされ、されたリンクのページは[anchor tag](#)についてされ、そのリンクはびじにされます。

`if(add && this_url.contains(my_site))`はをあなたのドメインにします。

```

import java.io.IOException;
import java.util.HashSet;
import java.util.Set;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.select.Elements;

public class readAllLinks {

    public static Set<String> uniqueURL = new HashSet<String>();
    public static String my_site;

```



```

public static void main(String[] args) {

    readAllLinks obj = new readAllLinks();
    my_site = "stackoverflow.com";
    obj.get_links("http://stackoverflow.com/");
}

private void get_links(String url) {
    try {
        Document doc = Jsoup.connect(url).userAgent("Mozilla").get();
        Elements links = doc.select("a");

        if (links.isEmpty()) {
            return;
        }

        links.stream().map((link) -> link.attr("abs:href")).forEachOrdered((this_url)
-> {
            boolean add = uniqueURL.add(this_url);
            if (add && this_url.contains(my_site)) {
                System.out.println(this_url);
                get_links(this_url);
            }
        });

    } catch (IOException ex) {

    }

}
}

```

プログラムはあなたのウェブサイトに対していくのがかかります。このコードをして、このウェブサイトからデータページのタイトルやテキストやなどをすることができます。はあなたのウェブサイトをscarping するにのをすることをめします。

ここではJSoupライブラリをしてリンクをしています、 `your_url/sitemap.xml` をしてリンクをすることもできます。

オンラインでJsoupによるWebクローリングをむ <https://riptutorial.com/ja/jsoup/topic/319/jsoupによるwebクローリング>

## 6: セレクタ

セレクタは、コンビネータでられたなセレクタのチェーンです。セレクタはとをしません、をむ。

ユニバーサルセレクタ\*は、セレクタがされていないつまり、\*.headerと.headerがしいににされます。

パターン	マッチ	
*	の	*
tag	されたタグをつ	div
ns E	ネームスペースns のEの	fb name finds <fb:name> elements
#id	IDが「id」の	div#wrap, #logo
.class	クラスが"class"の	div.left, .result
[attr]	"attr"のをつ	a[href], [title]
[^attrPrefix]	が"attrPrefix"でまる 。HTML5データセ ットをつのにします	[^data-], div[^data-]
[attr=val]	「attr」というのと 「val」にしいをつ	img[width=500], a[rel=nofollow]
[attr="val"]	「attr」というのと 「val」にしいをつ	span[hello="Cleveland"][goodbye="Columbus"], a[rel="nofollow"]
[attr^=valPrefix]	「attr」というのを つと「valPrefix」で まる	a[href^=http:]
[attr\$=valSuffix]	"attr"というのを つと"valSuffix"でわる	img[src\$=.png]
[attr*=valContaining]	"attr"というのを つと"valContaining" をむ	a[href*=search/]
[attr~=regex]	"attr"というのを つ	img[src~=(?i)\.(png jpe?g)]

パターン	マッチ	
	、およびにする	
	はのでみわせることができます	div.header[title]

セレクターのなりファレンス

## Examples

CSSセレクタをしたの

```
String html = "<!DOCTYPE html>" +
    "<html>" +
    "<head>" +
    "    <title>Hello world!</title>" +
    "</head>" +
    "<body>" +
    "    <h1>Hello there!</h1>" +
    "    <p>First paragraph</p>" +
    "    <p class=\"not-first\">Second paragraph</p>" +
    "    <p class=\"not-first third\">Third <a href=\"page.html\">paragraph</a></p>"
+
    "</body>" +
    "</html>";

// Parse the document
Document doc = Jsoup.parse(html);

// Get document title
String title = doc.select("head > title").first().text();
System.out.println(title); // Hello world!

Element firstParagraph = doc.select("p").first();

// Get all paragraphs except from the first
Elements otherParagraphs = doc.select("p.not-first");
// Same as
otherParagraphs = doc.select("p");
otherParagraphs.remove(0);

// Get the third paragraph (second in the list otherParagraphs which
// excludes the first paragraph)
Element thirdParagraph = otherParagraphs.get(1);
// Alternative:
thirdParagraph = doc.select("p.third");

// You can also select within elements, e.g. anchors with a href attribute
// within the third paragraph.
Element link = thirdParagraph.select("a[href]");
// or the first <h1> element in the document body
Element headline = doc.select("body").first().select("h1").first();
```

サポートされているセレクタのなは[こちら](#)でございました。

## Twitterマークアップをする

```
// Twitter markup documentation:
// https://dev.twitter.com/cards/markup
String[] twitterTags = {
    "twitter:site",
    "twitter:site:id",
    "twitter:creator",
    "twitter:creator:id",
    "twitter:description",
    "twitter:title",
    "twitter:image",
    "twitter:image:alt",
    "twitter:player",
    "twitter:player:width",
    "twitter:player:height",
    "twitter:player:stream",
    "twitter:app:name:iphone",
    "twitter:app:id:iphone",
    "twitter:app:url:iphone",
    "twitter:app:name:ipad",
    "twitter:app:id:ipad",
    "twitter:app:url:ipadt",
    "twitter:app:name:googleplay",
    "twitter:app:id:googleplay",
    "twitter:app:url:googleplay"
};

// Connect to URL and extract source code
Document doc = Jsoup.connect("http://stackoverflow.com/").get();

for (String twitterTag : twitterTags) {

    // find a matching meta tag
    Element meta = doc.select("meta[name=\"" + twitterTag + "\"]').first();

    // if found, get the value of the content attribute
    String content = meta != null ? meta.attr("content") : "";

    // display results
    System.out.printf("%s = %s\n", twitterTag, content);
}
```

```
twitter:site =
twitter:site:id =
twitter:creator =
twitter:creator:id =
twitter:description = Q&A for professional and enthusiast programmers
twitter:title = Stack Overflow
twitter:image =
twitter:image:alt =
twitter:player =
twitter:player:width =
twitter:player:height =
twitter:player:stream =
twitter:app:name:iphone =
twitter:app:id:iphone =
twitter:app:url:iphone =
twitter:app:name:ipad =
```

```
twitter:app:id:ipad =  
twitter:app:url:ipadt =  
twitter:app:name:googleplay =  
twitter:app:id:googleplay =  
twitter:app:url:googleplay =
```

オンラインでセレクトをむ <https://riptutorial.com/ja/jsoup/topic/515/セレクト>

## クレジット

S. No		Contributors
1	Jsoupをいめる	<a href="#">Alice</a> , <a href="#">Community</a> , <a href="#">Jeffrey Bosboom</a> , <a href="#">JonasCz</a> , <a href="#">Zack Teater</a>
2	HTMLの	<a href="#">Zack Teater</a>
3	Javascriptでされたページの	<a href="#">Zack Teater</a>
4	Jsoupでウェブサイトログインする	<a href="#">Joel Min</a> , <a href="#">JonasCz</a> , <a href="#">Stephan</a>
5	JsoupによるWebクローリング	<a href="#">Alice</a> , <a href="#">JonasCz</a> , <a href="#">r_D</a> , <a href="#">RamenChef</a>
6	セレクタ	<a href="#">JonasCz</a> , <a href="#">Stephan</a> , <a href="#">still_learning</a> , <a href="#">Zack Teater</a>