

# メディア情報応用 第7週

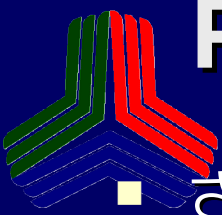
## Advanced Topics in Media Informatics Week 07

金沢工業大学 情報フロンティア学部  
メディア情報学科

山岸 芳夫

Yoshio Yamagishi

Dept. of Media Informatics  
Col. of Informatics and Human Communication  
Kanazawa Inst. Tech.

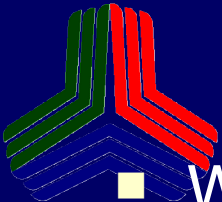


# Python

- さまざまな用途に用いることができるスクリプト系言語。ライブラリが豊富で科学技術計算にも対応しており、CaffeやTensorFlowなどの深層学習プラットフォームも基本的にPythonでプログラミングを行う

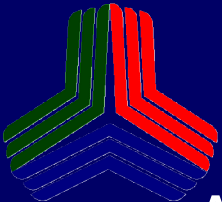
A script language for multi-purpose use. Python has rich libraries and is suitable for scientific calculation. Usually the programming in Deep-Learning platform such as Caffe or TensorFlow is based on Python.

- Webアプリケーションの開発にも用いられることが多い。さらに、**Python自身に Web サーバの機能も実装している**ため、Pythonの実行環境のみでWebアプリケーションを実装することが可能  
Python is often used to develop a Web application. In addition, since the **web server function is implemented in Python itself**, a web application can be constructed by Python executable environment only.



# WSGI

- Web Server Gateway Interface の略。Python で Web アプリケーションを開発する上での共通のインターフェース  
An abbreviation of Web Server Gateway Interface. The common interface for web application development using Python.
- WSGI対応の Web アプリケーションは、**WSGI対応の Webサーバであれば、どのような環境でも実行可能**→現在のPython Webアプリケーションの主流  
WSGI application can be **executable with any WSGI compatible web server**. So currently this is main stream of Python web application.



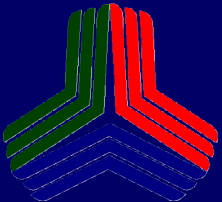
# WSGI

- Apache httpd も従来より CGI や mod\_python モジュールを使って Python アプリケーションを動かすことが出来るが、nginx などの他の Web サーバでそのアプリケーションが動くとは限らない

Python application can be executable with Apache httpd as CGI program or mod\_python module application.  
However, such application can not always be executable with other web servers such as nginx.

- 現在は mod\_wsgi を使って WSGI 対応アプリケーションを動作させるのが主流（今回はやらない）

WSGI application can be executable with Apache httpd including mod\_wsgi (but here we omit it).



# Pythonの文法

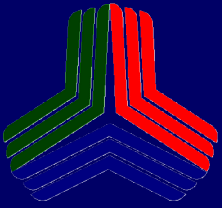
## Syntax of Python

- 行末のセミコロン(";")が要らない

The semicolon (";") at the end of the line is not required.

- ブロックは丸括弧("{ }")ではなくインデントで区切る

The block is not separated by curly bracket ("{}") but indent.



# Hello World

- Moodleの7週目から week07.tar.gz をダウンロードして展開し、hello.py をpythonで実行させる  
Download week07.tar.gz from 7<sup>th</sup> topic of the Moodle course, extract it and then execute hello.py by python.

`python hello.py`

- 実行したら <http://localhost:8000/> にアクセスしてみよう。Hello World が表示されるはず  
Try to access <http://localhost:8000/>. You can see "Hello World".



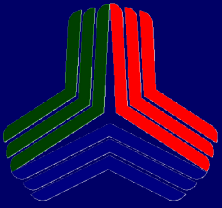
# Python chat

- week07.tar.gz 中にある chat.py をpythonで実行させる

Execute chat.py included in week07.tar.gz with python.

`python chat.py`

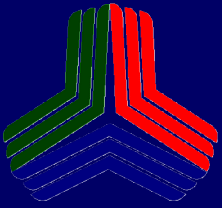
- 実行したら <http://localhost:8000/> にアクセスしてみよう。これまでPHP, サーブレット、JSPで実装してきたものとほぼ同じ仕様のチャットが表示されるはず  
Try to access <http://localhost:8000/>. You can see the chat of which appearance is almost same as PHP, Servlet and JSP chat which have been built by us so far.



# Ajax

- Asynchronous Javascript and XML...非同期の Javascriptによる XML を介した通信  
Asynchronous communication with Javascript via XML.
- 通常の同期通信では、サーバからコンテンツが全てダウンロードされないと処理できない  
Conventional synchronous communication does not allow the browser to render the page until all of the contents are downloaded.
- Ajaxでは、**Javascript によるプログラムがクライアント上で処理を行いながら通信しているため、コンテンツのダウンロードを待たなくてもよい（非同期）**  
The browser can render the Ajax-implemented web page without completion of the contents download because **Javascript executes programs with realtime (asynchronous) communication to the server.**





# Ajax

- **XMLHttpRequest**オブジェクトを用いることで、Javascript が直接サーバからデータをやり取りする事が出来る。データの形式は原則 XML だが、それに限ったわけではない  
The **XMLHttpRequest** object enables Javascript programs to communicate with the web server directly. The data format of XMLHttpRequest communication is recommended to be XML, but the other formats such as plain text or HTML are also available.
- サーバ側にCGIもしくはPHPなど、何らかのプログラムを動かして利用することが多い  
The Ajax application often requires server-side program such as CGI or PHP, etc.

# インクリメンタルサーチ

## Incremental Search

- Ajaxの機能を用いて、入力した文字をchatのログの中で逐次検索（インクリメンタルサーチ）するページを作ってみよう。  
Build an incremental search page of chat with Ajax function.
- week07.tar.gz の中の **search.php** と **search\_ajax.php** を **/var/www/html** にコピーして、  
**http://localhost/search.php** にアクセスしてみよう。通常の検索プログラムが表示される。  
Copy **search.php** and **search\_ajax.php** in week07.tar.gz and then paste them into **/var/www/html**. You can see conventional search program when you access **http://localhost/search.php**

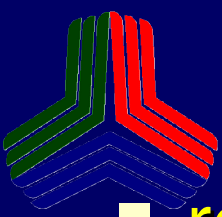
# インクリメンタルサーチ

## Incremental Search

- うまく動作したら、今度は [http://localhost/search\\_ajax.php](http://localhost/search_ajax.php) にアクセスしてみよう。文字入力の途中で検索が行われるインクリメンタルサーチが可能になっている
- If you make sure that the conventional search page works, then access [http://localhost/search\\_ajax.php](http://localhost/search_ajax.php) to check the incremental search function.

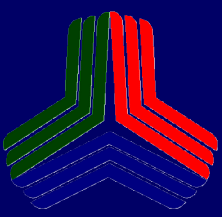
# search\_ajax.phpの要点

## Essentials in search\_ajax.php

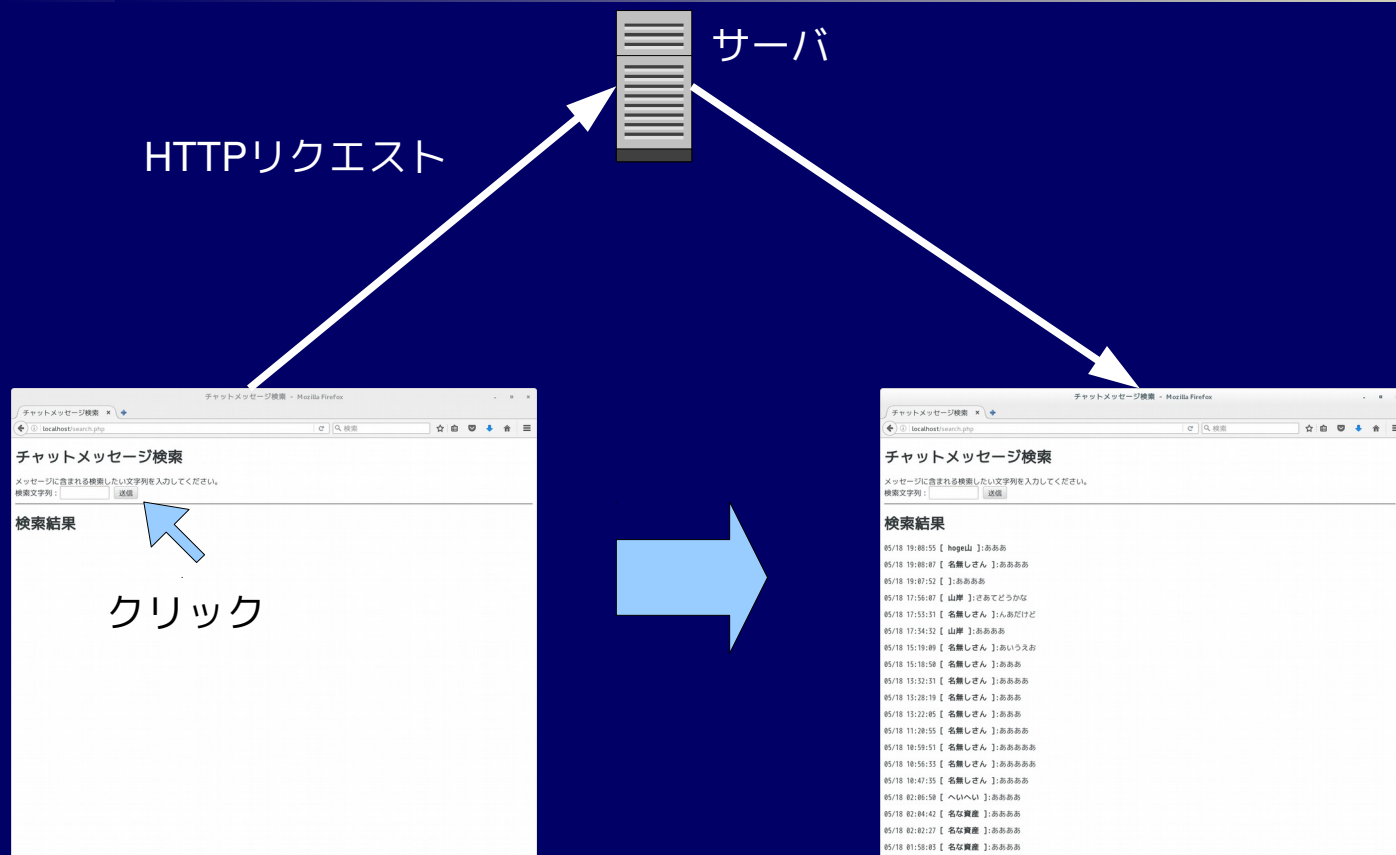
- 
- `req = new XMLHttpRequest();`...XMLHttpRequest オブジェクト req を宣言  
declares XMLHttpRequest object "req".
  - `req.onreadystatechange = show_data;` ... 待ち受け状態(readystate)が変化(change)した場合（即ち、サーバから何かレスポンスがあった場合）に、**コールバック関数**と呼ばれる関数を呼び出す。このプログラムのコールバック関数は `show_data` である。  
calls "callback function" when the readystate is changed(i.e. server returns some responses). At this time the callback function is "show\_data".

# search\_ajax.phpの要点

## Essentials in search\_ajax.php

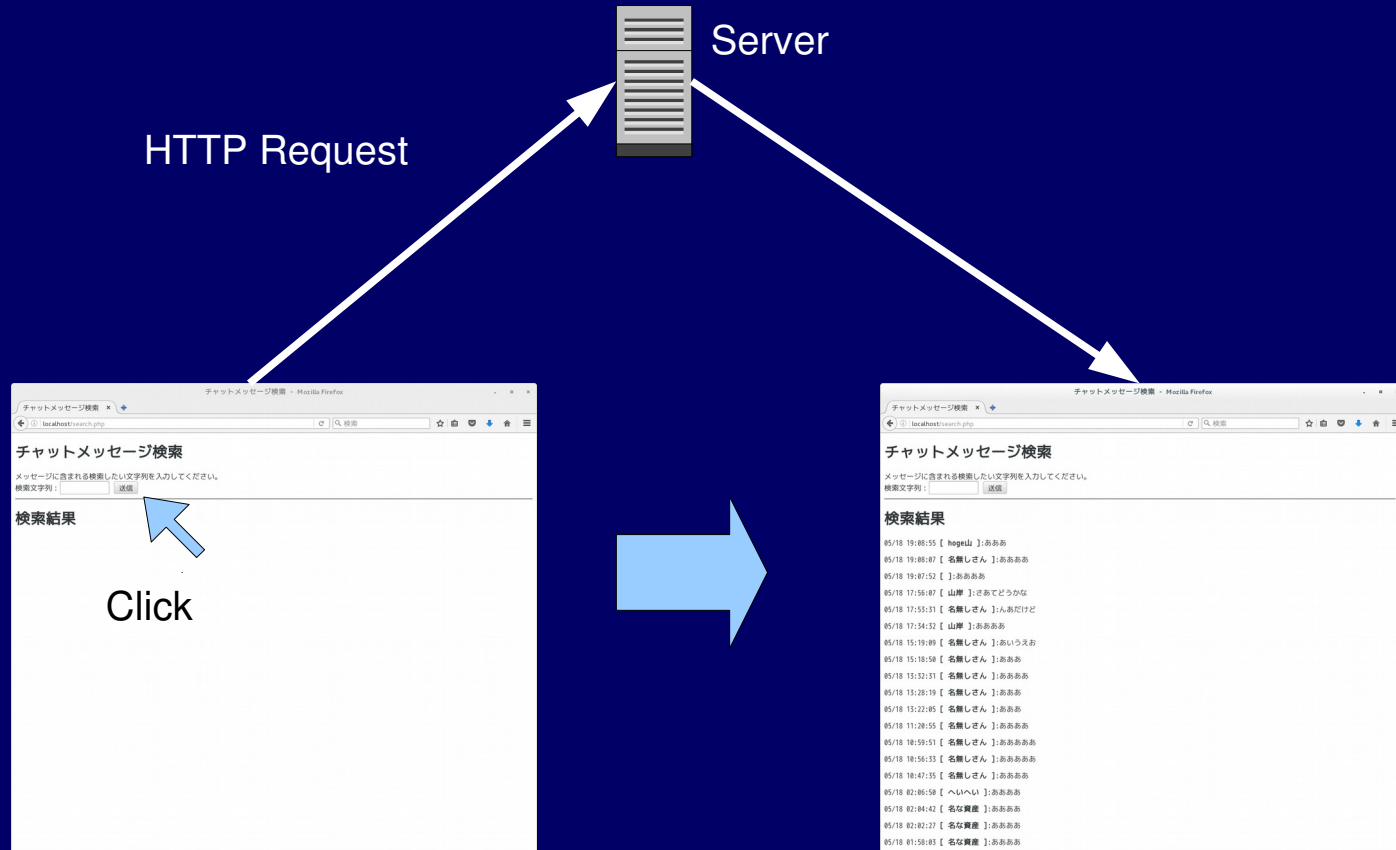
- 
- `onkeyup="load_data()"`...押されていたキーが放された場合(`onkeyup`)に `load_data`関数を呼び出す  
calls "load\_data" function when the pressed key is released(`onkeyup`).

# 通常のHTTPアクセス



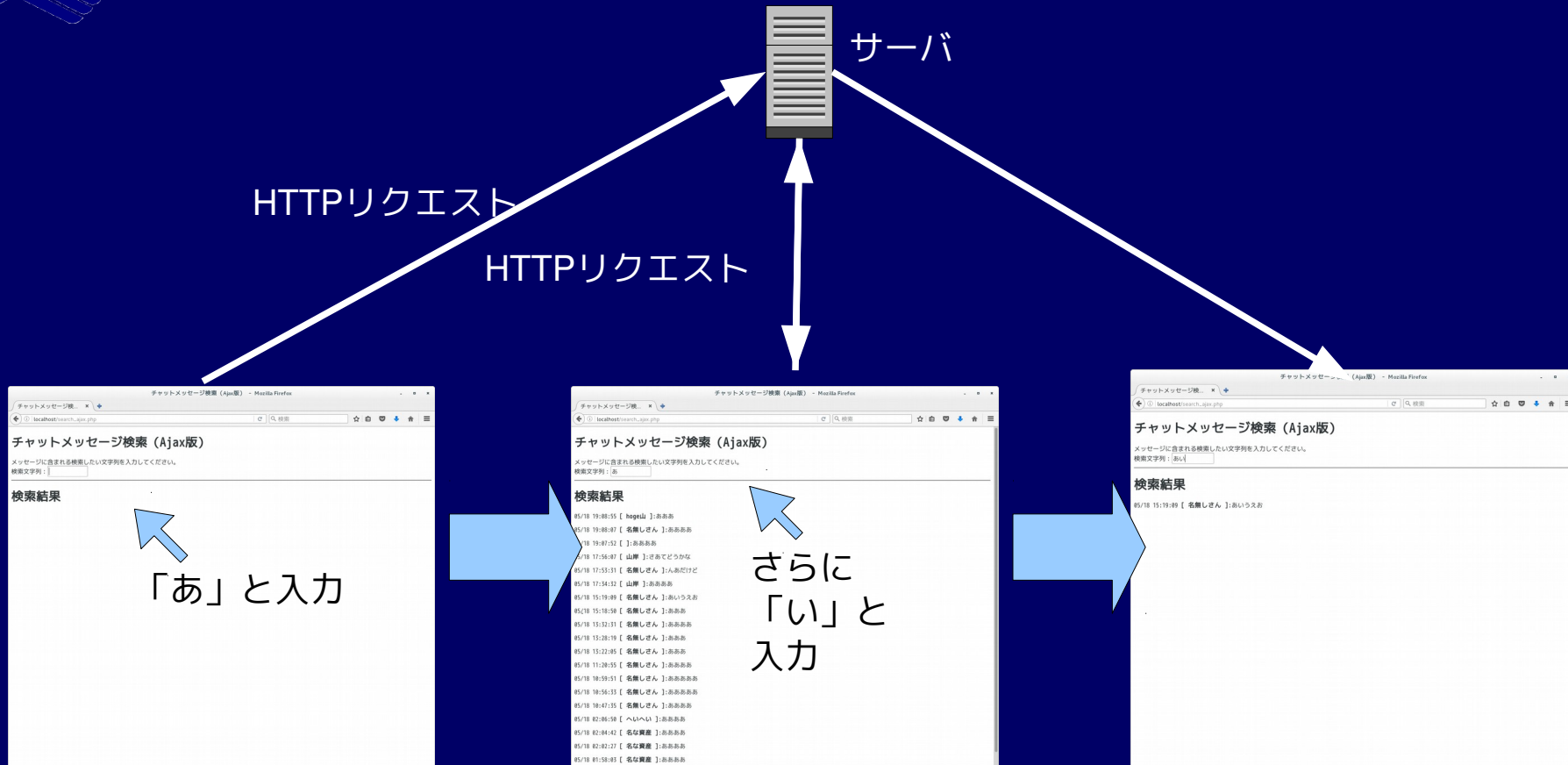
送信ボタンを押す / リンクをクリックすると、  
次の画面に移行する

# Conventional HTTP Access



**When a link or submit button is clicked,  
the transition to next contents occurs.**

# Ajax によるHTTPアクセス

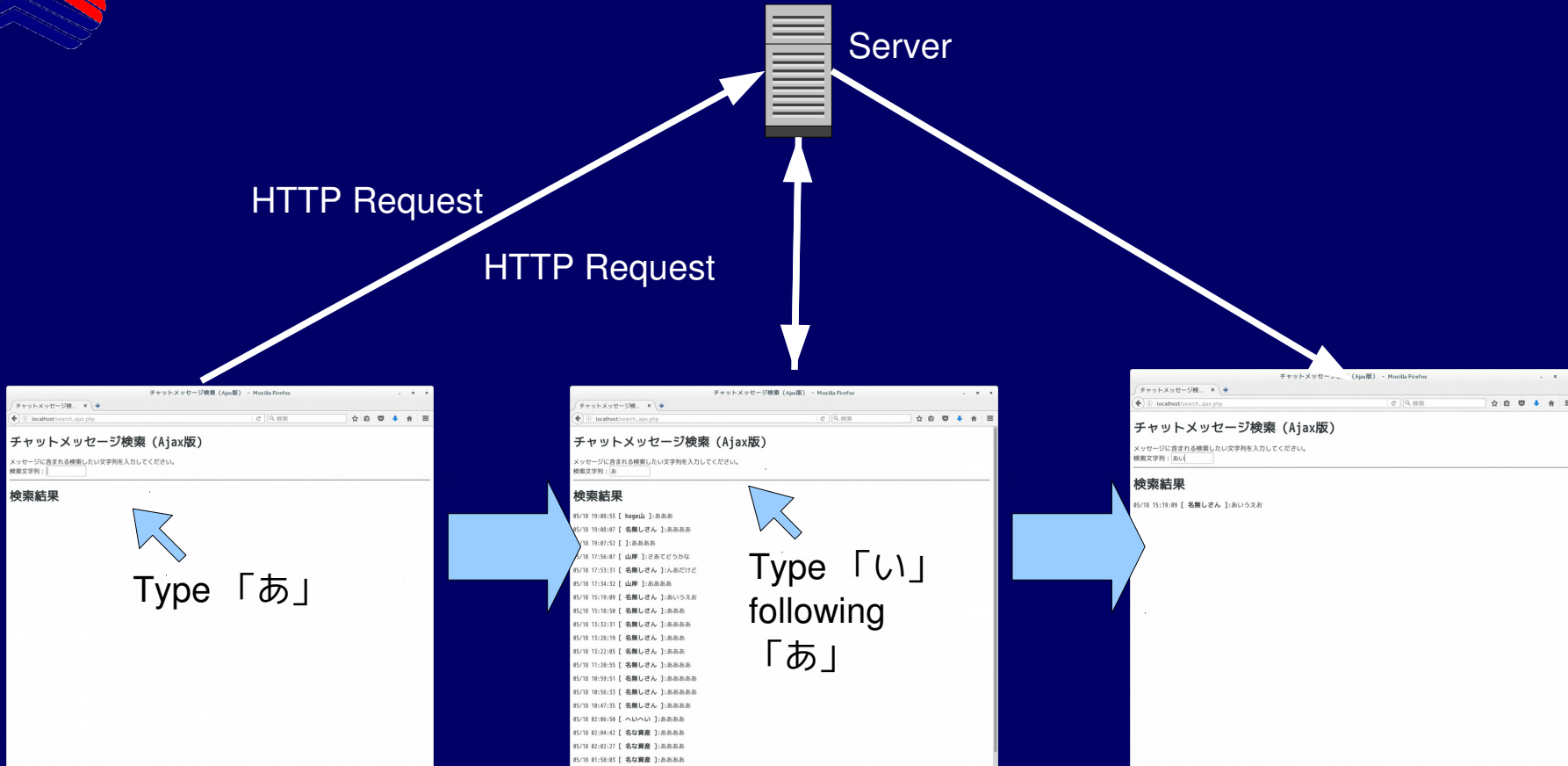


画面は移行せず、その中身が動的に変化する

(その裏では状態の変化に応じて次々とHTTPリクエストが発行されている)



# Ajax HTTP Access



**The Contents are changed dynamically without transition  
(In background, the HTTP requests are transmitted  
at each moment of state change)**