「情報Ⅱ オンライン学習会」

～情報システムを改造してみよう～

明星大学　長慎也

https://bitarrow.eplang.jp/?joho2

前回のあらすじ

- 情報システムとは
- 情報システム作成演習を高校で行うには
- Bit Arrowの宣伝 😅
- 「グループチャット」の作成
  - サーバ側
    - 書き込み
    - 読み出し
  - クライアント（ブラウザ側）
    - 画面
    - 読み出し
    - 書き込み

授業の進め方（一例）

- クライアント側か，サーバ側か，役割を決めよう
- それぞれ作成して，単体テストをしよう
  - どんなテストをすればよいか，考えてみよう
- それぞれのプログラムが連携できるか，結合テストをしよう
  - どんなテストをすればよいか，考えてみよう

- 自分たちでシステムを改造してみよう
  →今回をお楽しみに！

前回のシステムの問題点

- 誰が書いたものなのかわからない
- いつ書いたのかもわからない
- 別人がなりすまして書いていないか？　などなど…

- 書き込み以外の情報をしまう方法
  - 例：「名前」と「書き込み」をしまう
- 今回は，もっとも単純な「タブ区切りのテキストファイル」を使用して作成

前回のグループチャットシステムの構成

- サーバ側
  - 書き込みの追加
  - 書き込みの読み出し
- クライアント（ブラウザ）側
  - 書き込みフォームの表示
    - 入力欄
    - 書き込みボタン
    - 書き込み表示欄
  - 書き込みの表示
  - 書き込み追加の実行

Pythonで作成

HTML+JavaScriptで作成

前回の講演で作ったプログラムを見てみよう

- https://bitarrow3.eplang.jp/bitarrow/
  - **クラス名** `joho2_2402`
  - ユーザ名　*<<前回使ったID>>*
  - パスワード　（空欄）

※講演終了後，こちらのpptxをご覧になりながらでも作業可能です.

※前回のスライドの作業がまだの方はそちらを先に行ってください.

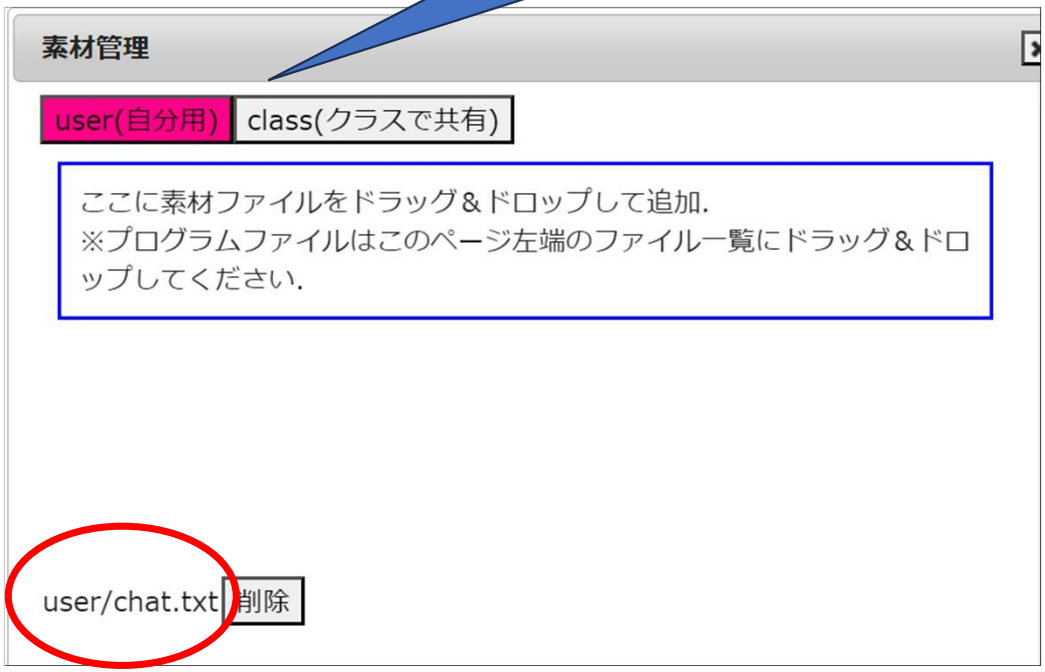
前回のプロジェクトを開いてみよう

- まずはサーバ側のプロジェクトを開いてみよう

前回の書き込みデータを見てみよう

- 「ファイル」→「素材管理」
  - user/chat.txtをクリックして中身を確認

出ない場合は，userとclassを
交互に押して最新情報に更新



前回の書き込みデータ

- 1行ごとに書き込みの内容が記録されている
- 書き込みの内容以外の情報を記載するには「1行を複数の項目に分ける」必要がある

タブ区切りのテキストファイル

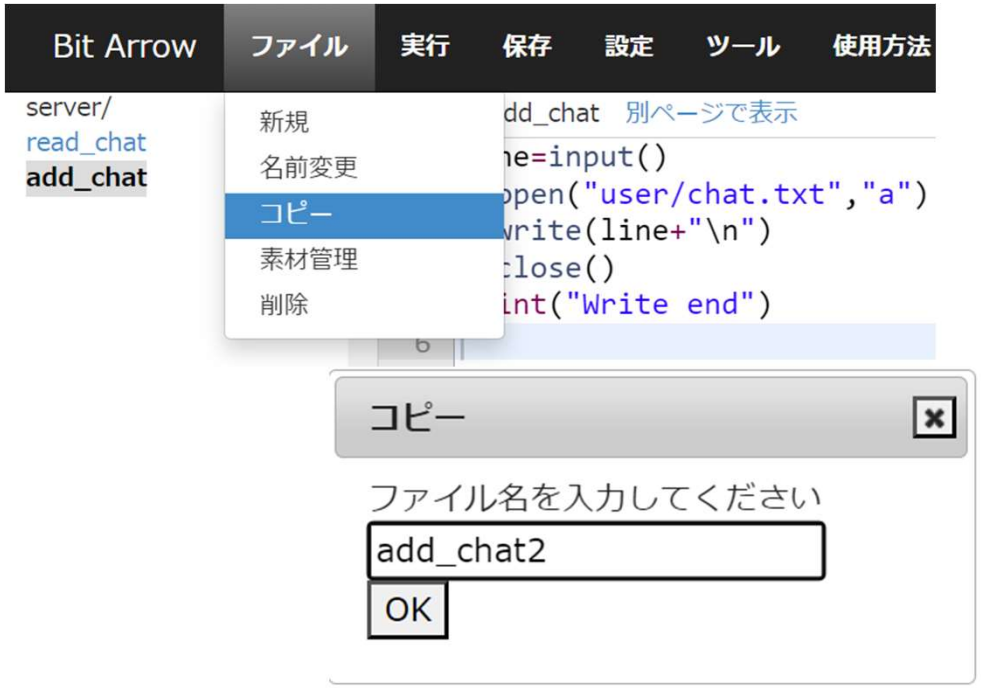
- 例えば，「名前」と「書き込み本体」を格納したい場合
  - 「名前」と「書き込み本体」の間にタブを入れて区別する

山田[ ]こんにちは
鈴木[ ]ごきげんいかが？
田中[ ]よろしくお願いいたします。

タブ
プログラム上は ¥t と表記

サーバ側：書き込みの追加を改造しよう

- サーバ側のプログラム　add_chat をコピー



add_chat2の中身

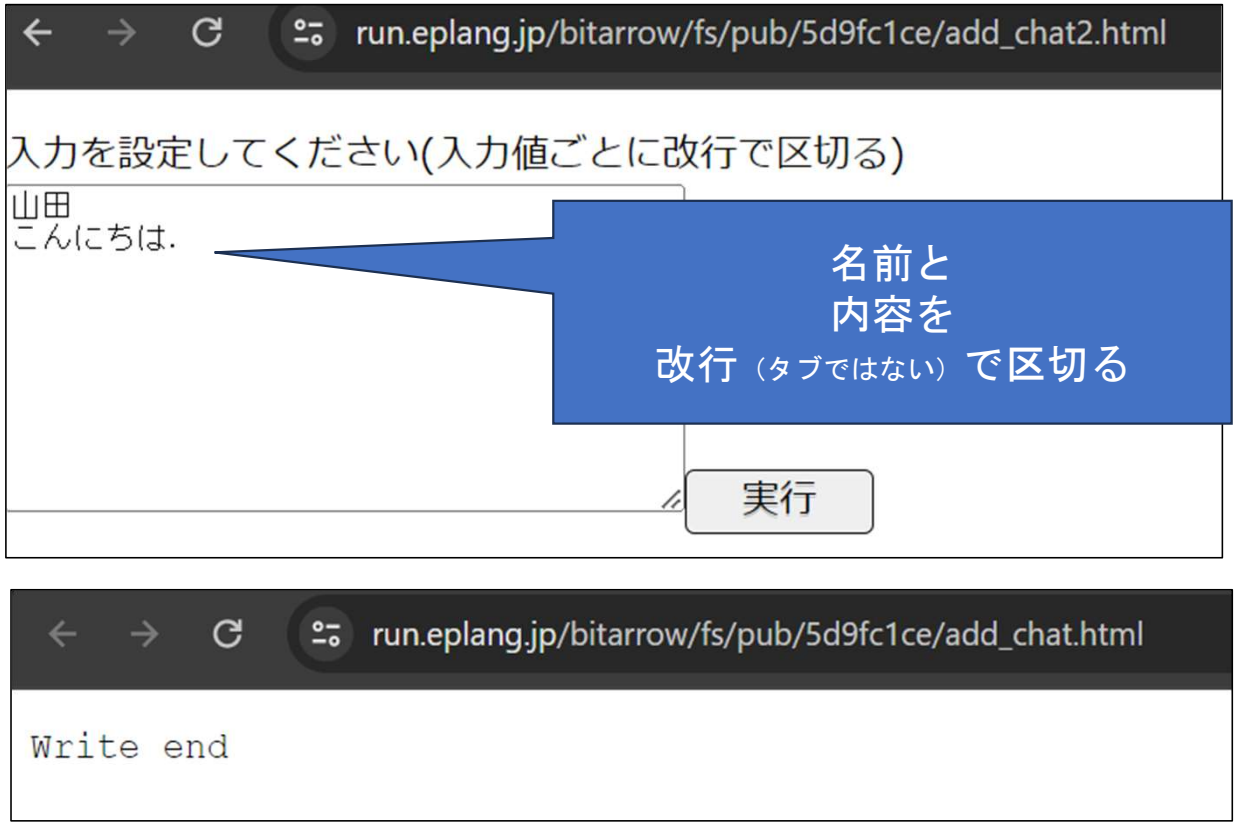
add_chat2

```
name=input()
content=input()
f=open("user/chat2.txt","a")
f.write(name+"¥t"+content+"¥n")
f.close()
print("Write end")
```

名前と内容をタブ(¥t)で連結する

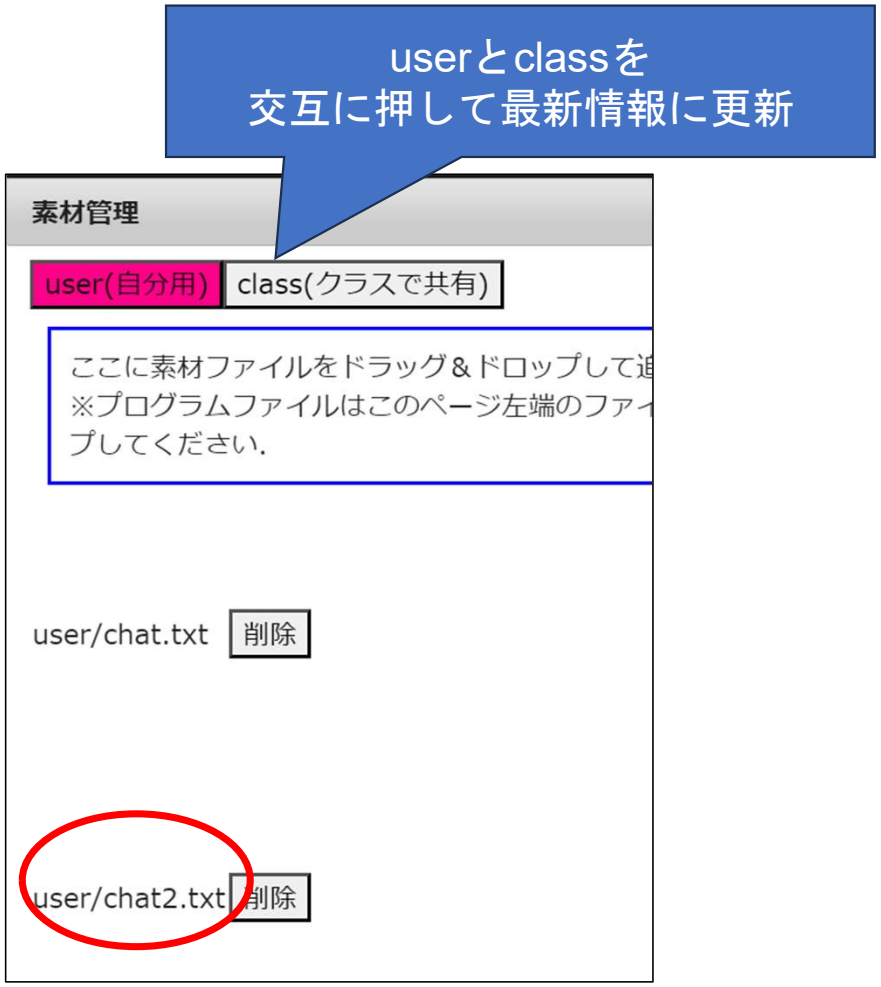
- 名前を入力
- 書き込む内容を入力
- ファイルを追記モードで開く
  - ファイル名が変わっています
- 書き込む内容を追記
- ファイルを閉じる
- メッセージを表示

実行する


run.eplang.jp/bitarrow/fs/pub/5d9fc1ce/add_chat2.html
入力を設定してください(入力値ごとに改行で区切る)
山田
こんにちは.
名前と
内容を
改行（タブではない）で区切る
実行
run.eplang.jp/bitarrow/fs/pub/5d9fc1ce/add_chat.html
Write end

書き込まれた内容の確認

- 「ファイル」→「素材管理」

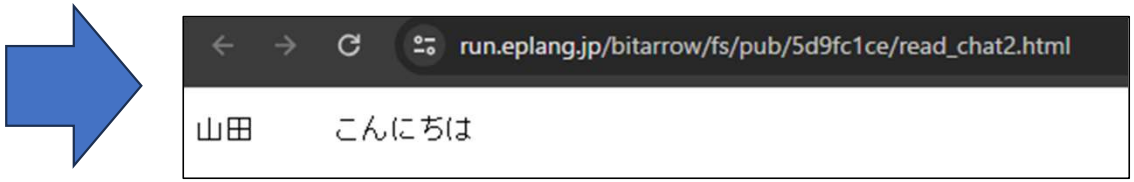


サーバ側：書き込みの読み出しも改造しよう

- サーバ側のプログラム　read_chat をコピー
  - 名前：　read_chat2
  - 読み込むファイル名をchat.txtからchat2.txtに変更するだけ

read_chat2

```
f=open("user/chat2.txt","r")
for i in f:
    print(i,end="")
f.close()
```

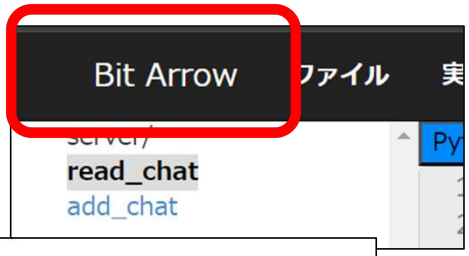


何件か書き込んでみよう（単体テスト）

- add_chat2を実行
- read_chat2を実行して，追記されているかを確認

クライアント側のプログラムを改造

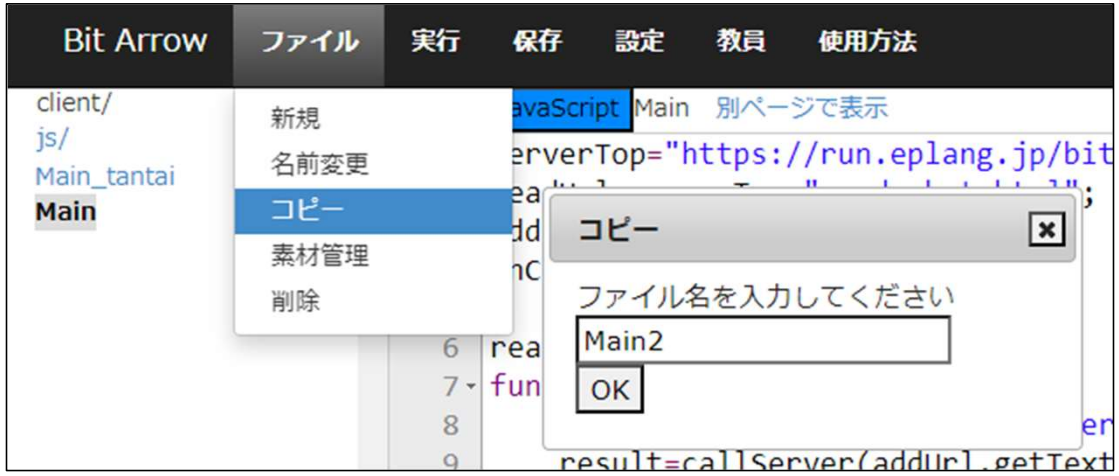
- 最初の画面に戻る
- クライアント側のプロジェクトを開く



クライアントの動作の改造

- 前回作ったファイルを開く
- ファイル→コピー
- 別の名前をつける
  - ここでは前回のファイルをMain，今回改造するものをMain2とします

.



画面（HTML）の内容の変更点

Main （HTML）

```
<html>
<h1>掲示板</h1>
<input name="text"/>
<button name="write">Write</button>
<br/>
<div name="result"></div>
<pre name="hist"></pre>
</html>
```

Main2 （HTML）

```
<html>
<h1>掲示板</h1>
名前<input name="name"/><br/>
内容<input name="content"/><br/>
<button name="write">Write</button>
<br/>
<div name="result"></div>
<pre name="hist"></pre>
</html>
```

入力欄がnameと
contentの2つに

動作（JavaScript）の変更点

Main （JavaScript）

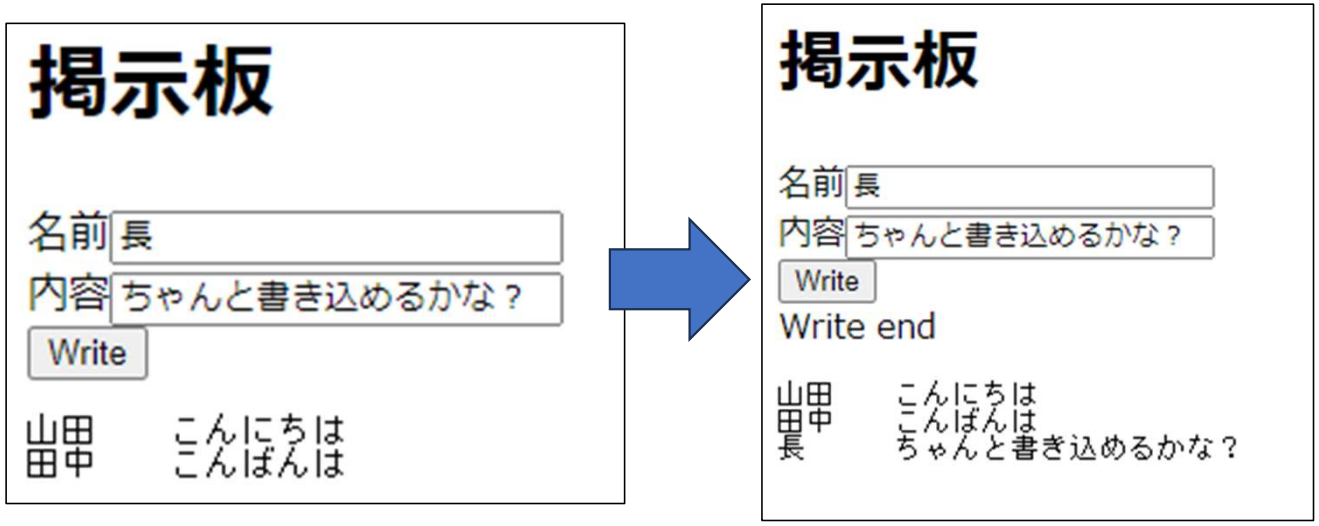
```
serverTop="(略)";
readUrl=serverTop+"read_chat.html";
addUrl=serverTop+"add_chat.html";
onClick("write",write);
read();
function write() {
    setText("result","Running at Server...");
    result=callServer(addUrl,getText("text"));
    setText("result",result);
    read();
}
function read() {
    r=callServer(readUrl);
    setText("hist",r);
}
```

Main2 （JavaScript）

```
serverTop="(略)";
readUrl=serverTop+"read_chat2.html";
addUrl=serverTop+"add_chat2.html";
onClick("write",write);
read();
function write() {
    setText("result","Running at Server...");
    result=callServer(addUrl,getText("name")+"¥n"+getText("content"));
    setText("result",result);
    read();
}
function read() {
    r=callServer(readUrl);
    setText("hist",r);
}
```

nameとcontentの内容を改行(¥n)で区切って送信

実行してみると....



- テキストファイルの中身をただ表示しているだけなので，書き込みの表示がシンプルすぎる
- 名前と書き込みを区別して表示するには？

read関数を改造

Main2　（JavaScript）の一部

```
function read() {
    var r=callServer(readUrl);
    var lines=r.split("¥n");
    setText("hist","");
    for (var line of lines) {
        var data=line.split("¥t");
        if (data.length<2) continue;
        var name=data[0];
        var content=data[1];
        addText("hist","名前: "+name+"<br/>");
        addText("hist","内容: "+content+"<br/>");
        addText("hist","<hr/>¥n");
    }
}
```

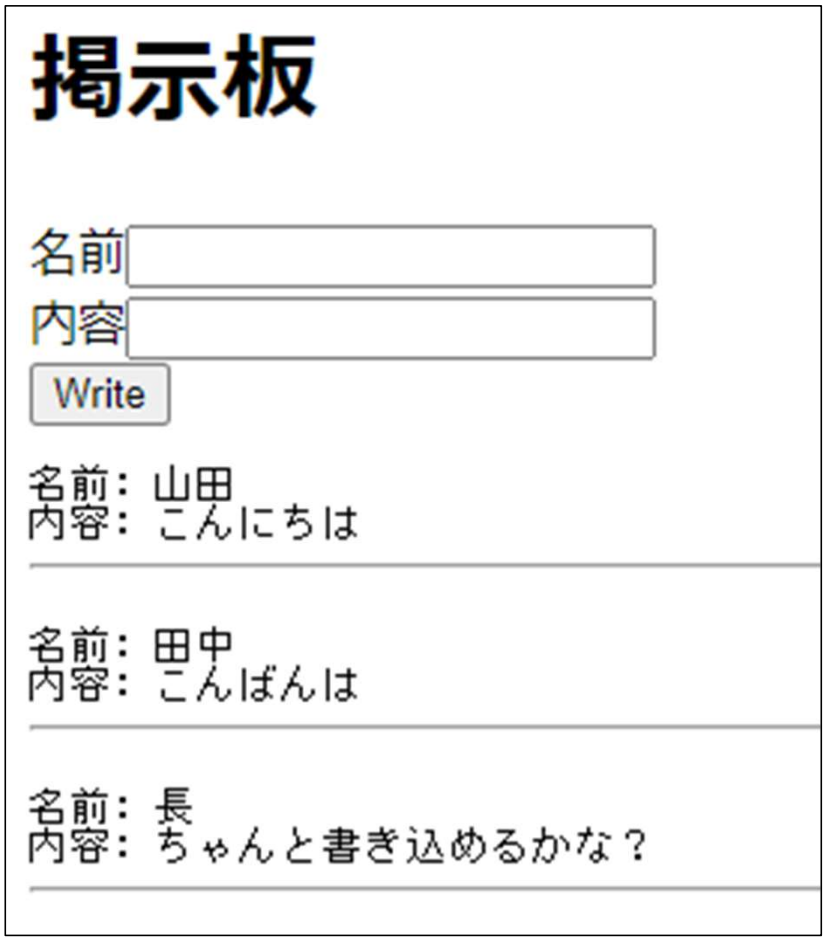
- 書き込みを読み込む
- 行ごとに分割
- 書き込み表示欄を空にする
- 行ごとに
  - タブ(¥t)で分割する
    - (何も書いていない行は飛ばす)
  - nameとcontentを取り出す
  - それぞれ「名前:」「内容: 」という見出しをつけて表示
  - 区切り線<hr/>を表示

実行結果

- 表示を工夫してみましょう

```
addText("hist","名前: "+name+"<br/>");
```

- 追加したい内容を，HTMLを表す文字列で記述
- "…." の中はそのまま出力
- name やcontentは名前や書き込みをあらわす変数



他にもこんなことができます

- 書き込みに含まれる属性を増やす
  - 投稿された時刻を記録・表示する
  - URLをつけて画像を表示させる
- データファイルを増やす
  - ユーザ認証
    - ユーザ名とパスワードを記録したファイルを用意
    - 入力されたユーザ名とパスワードが正しいか確認
  - いいね！
    - 書き込みに対して「いいね！」されたことを記録するファイル
    - 書き込み表示時に，「いいね！」の数を集計

他にも作れそうな情報システム

- ファイルを保存・読み出しするもの
- あるユーザが保存→別のユーザが読み出す
- リアルタイム性
  - サーバとの通信：場合によるが，３０秒に一回くらい
  - サーバとの通信のない，クライアント側の書き換え（アニメーションなど）は１秒に何十回でもできる
    - 「いつサーバとの通信が起きるか？」ということを考察してもらうのも大切
- 生徒にアイデア出ししてもらう（次ページ参照）
- 教員のみなさんだったらどんなシステムを作りたいですか？

作品計画書

- サーバ側担当　　：　山田花子
- クライアント側担当：　田中次郎

- 作品のタイトル

なんとかシステム

- 機能・使い方（想定ユーザ）

○ ○に困っている人向けに，× ×を支援する

なんとか機能

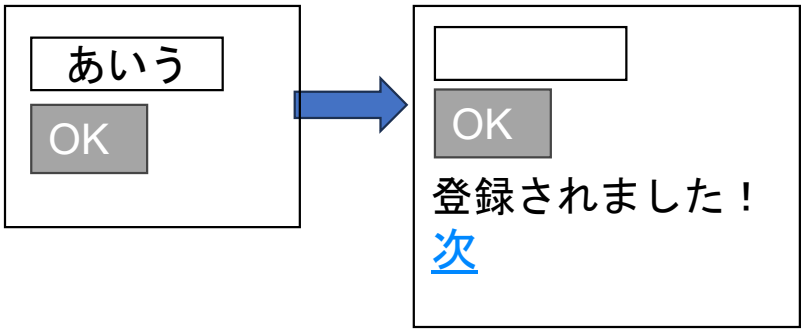
なんとかを登録する

かんとか機能

かんとかを確認する

- データの構造
  - a.txt
    - 属性１
    - 属性２
  - b.txt
    - 属性３
    - 属性４
    - 属性１（a.txtのデータと連携

- 画面構成（図を書いて）



学習のための情報源

- Python
  - https://bitarrow.eplang.jp/index.php?python
  - 今回紹介したのは「サーバ側で実行」するプログラムです.
  - 一部に使えない関数などがあります.
  - 書籍なども参考になります
- JavaScript
  - https://bitarrow.eplang.jp/index.php?javascript
  - Bit ArrowのJavaScript タブに書くプログラムは，実際のJavaScriptとは異なります.
  - 本物のJavaScriptを書きたい場合，HTMLタブに`<script>...</script>`を使って書くこともできます.