

Arduino入門勉強会 #4.5

【Webサーバ編（補習）】

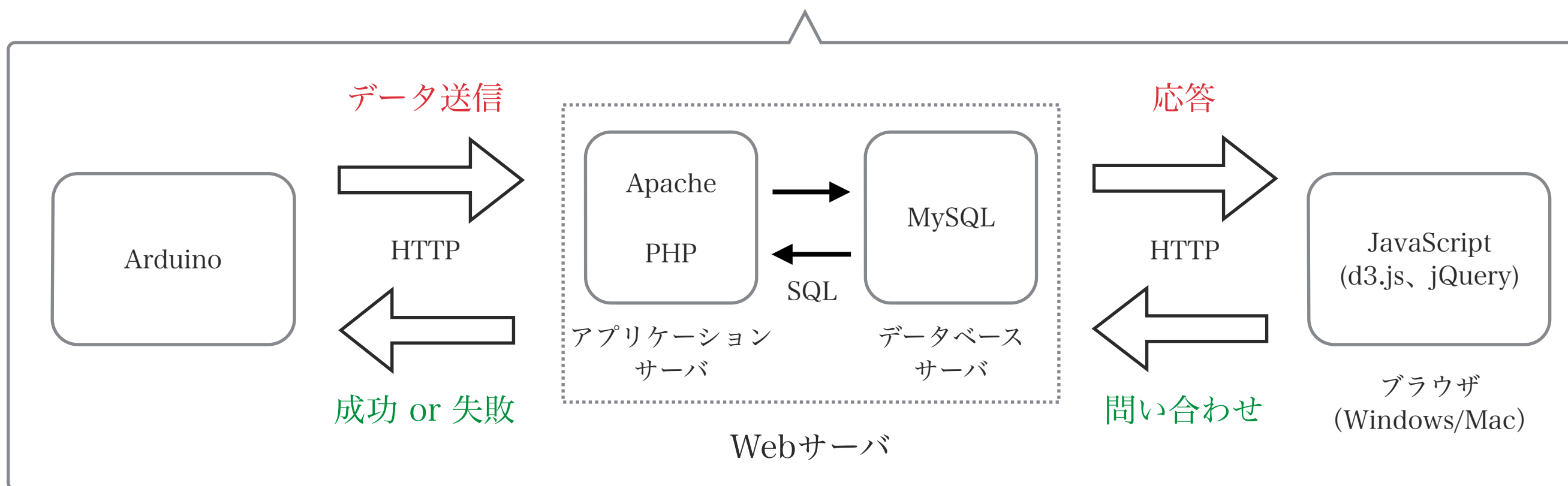
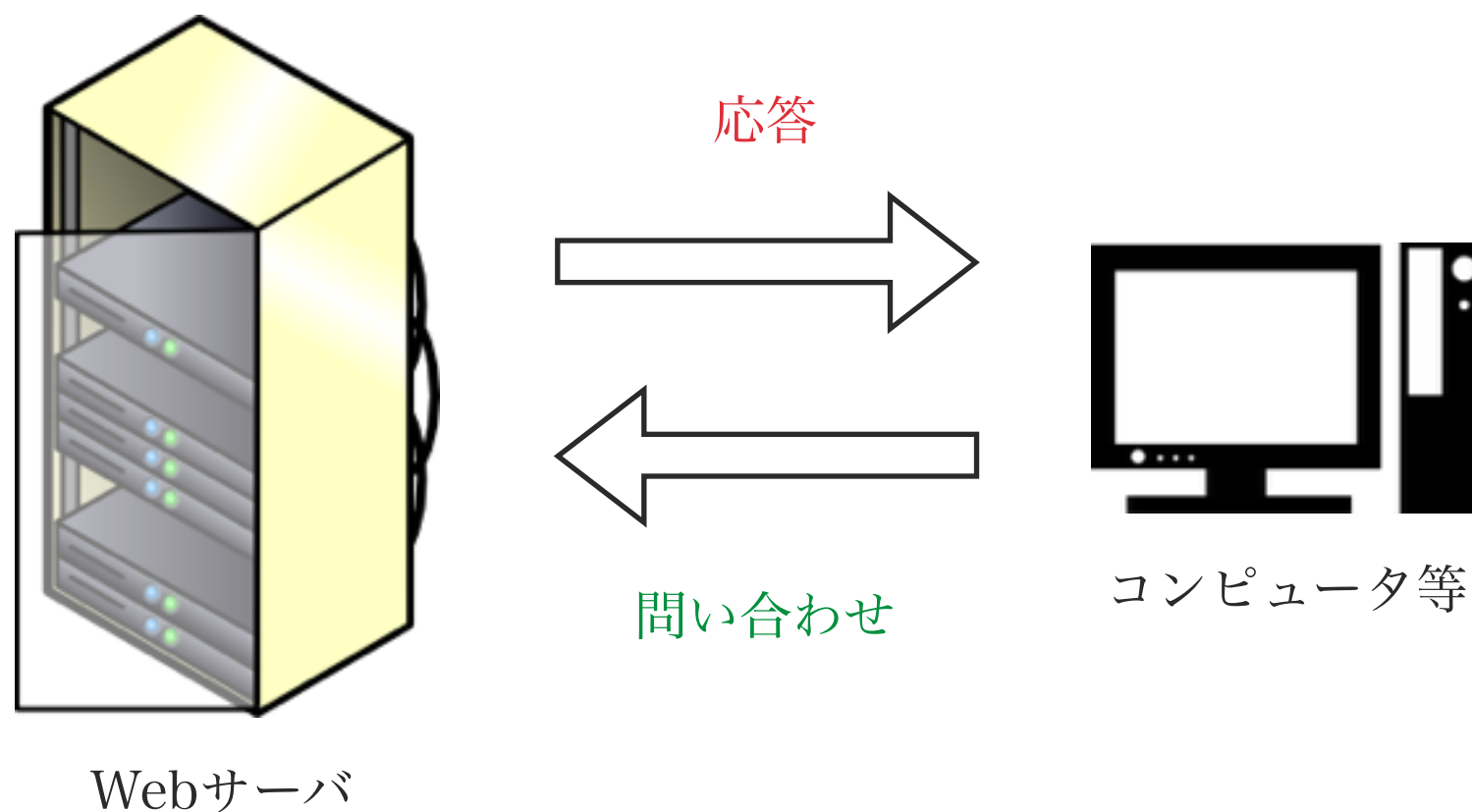
平成27年8月28日

ソフトピアジャパン ドリーム・コア1F ネクストコア

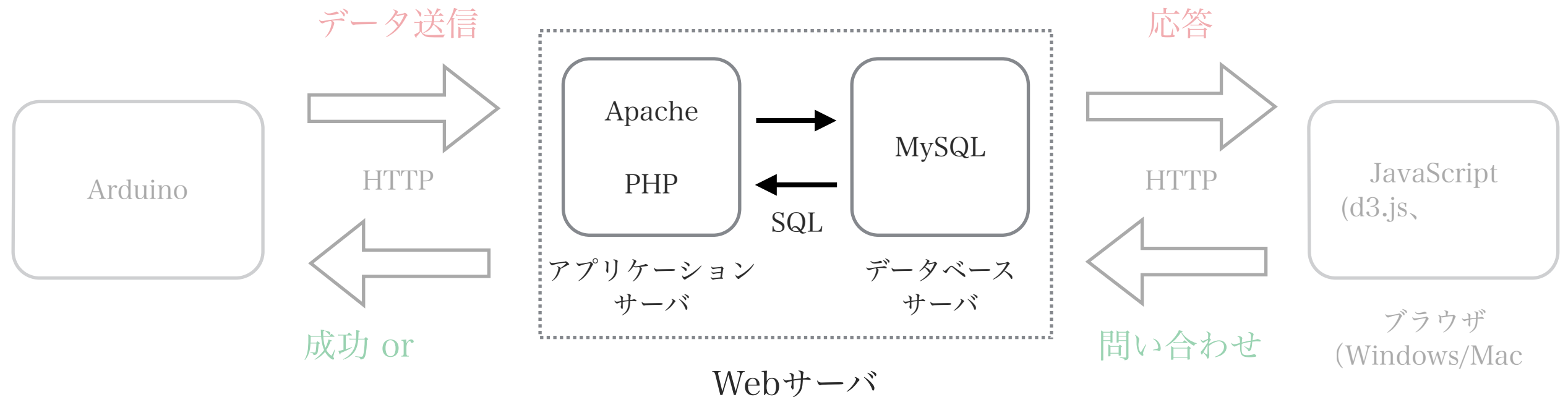
Webサーバ基礎

BASIC

Webサーバ



LAMP



LAMP

データベースを含めた一般的なWebサーバの構成

- **Linux** サーバOS
- **Apache** Webサーバソフトウェア (httpd)
- **MySQL** データベース (DBMS)
- **PHP** サーバ言語

仮想サーバ

VIRTUAL SERVER

仮想サーバ

MAMPとXAMPP

サーバに必要なソフト群（LAMP等）をインストールしなくても、
一時的にサーバ機能を実現できる技術

MAMP

Mac/Apache/MySQL/PHP



XAMPP

X (Windows、Mac、Linux) /
Apache/MySQL/PHP/Perl



仮想サーバ

MAMP

ダウンロードURL

<https://www.mamp.info/en/downloads/>

MAMP

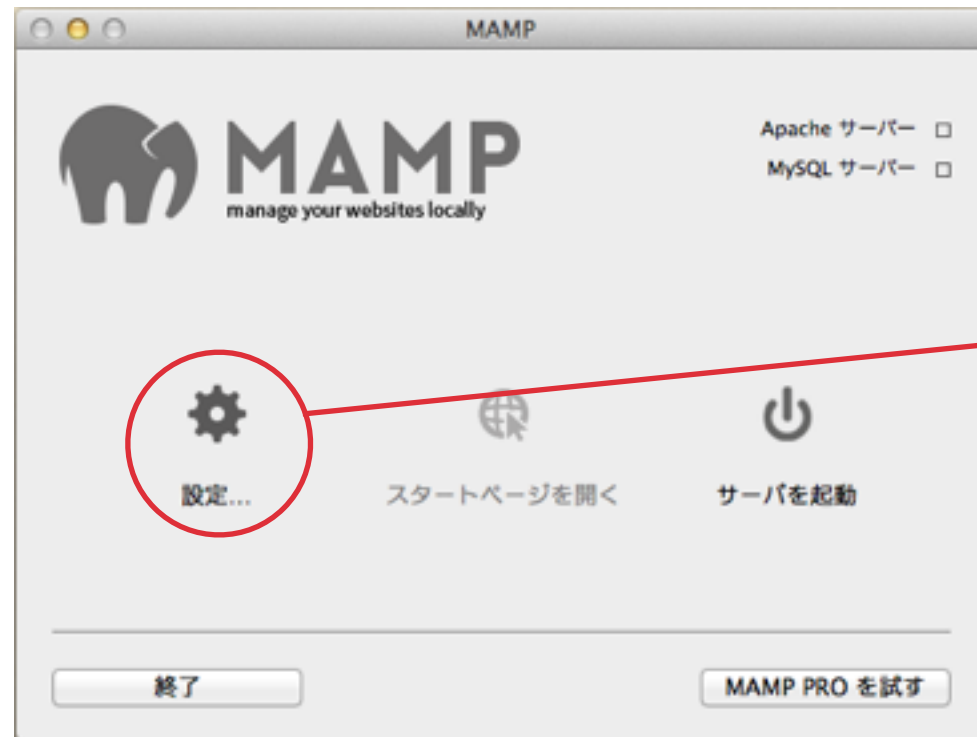


②緑になったら起動

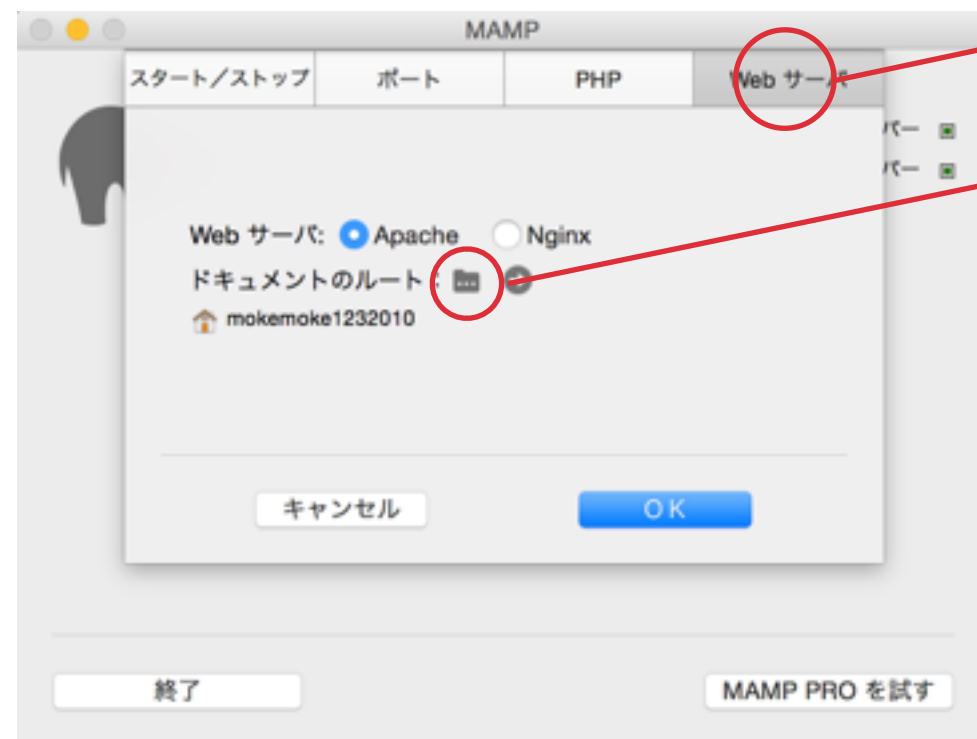
①起動する

仮想サーバ

PHPサンプル①



① 設定を押す



② 「Webサーバ」タブを開く

③ デスクトップを指定する

仮想サーバ

PHPサンプル②

PHPプログラムをMAMPで動かしてみる

Githubのからサンプルプログラム（「/05/Sample.php」）をダウンロードし、デスクトップに配置してブラウザで開く

Github URL

<https://github.com/ichi-3/ArduinoStudyGroup>

<http://localhost:8888/sample.php>

MAMPに設定したポート番号と同じ



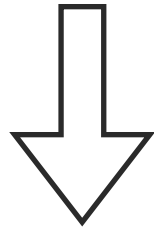
仮想サーバ

PHPサンプル③

- ①PHPサンプルを改良する（〇〇に固有の名前を入れる）
- ②他の人のページにアクセスしてみる

http://localhost:8888/sample.php

localhost=自身のコンピュータ



http://192.168.0.13:8888/sample.php

他の人のIPアドレス
を聞いて入力する

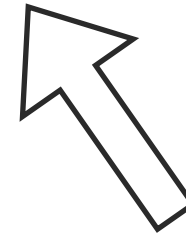
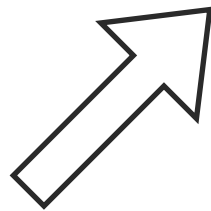
データベース

DATABASE

データベース
MySQL



MySQL (RDBMS)



phpMyAdmin
(GUIツール)



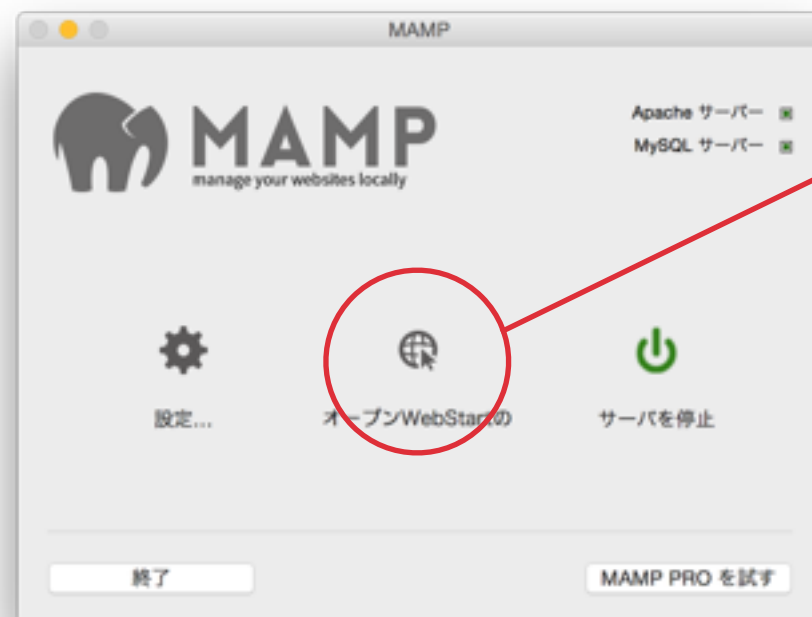
PHP
(プログラム)

データベース

座席表データベースのインポート①

座席表プログラムのデータベースを準備する

①phpMyAdminを開く



①スタート画面を開く

②ツール → phpMyAdmin



データベース

座席表データベースのインポート②

座席表プログラムのデータベースを準備する

②arduino_ws0401.sqlをインポートする

サーバ: localhost:8889

データベース SQL 状態 ユーザ エクスポート **インポート** 設定

現在のサーバへのインポート

インポートするファイル:

ファイルは圧縮されていないもの、もしくは、gzip, bzip2, zip で圧縮されているもの。
圧縮ファイルの名前は、[フォーマット].[圧縮形式]で終わっていること。例: .sql.zip

アップロードファイル: **ファイルを選択** 選択されていません (最長: 32MiB)

ファイルの文字セット: utf-8

部分インポート:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout.
(transactions.)

Skip this number of queries (for SQL) or lines (for other formats), starting from the first

フォーマット:

SQL

フォーマット特有のオプション:

①「インポート」を開く

②sqlファイルを選択する

データベース

座席表プログラムの実行①

①座席表プログラムを実行してみる

「04」フォルダの「**ArduinoWS04_Web01**」をデスクトップにコピー

②MAMPのMySQLポートが8889になっているか確認する



③座席表プログラムにMAMPを通してアクセスする

http://localhost:8888/ArduinoWS04_Web01/index.php

データベース

座席表プログラムの実行②

④ データベースを変更して結果を見してみる

② 「表示」タブをクリック

① 「Seats」をクリック

Server: localhost:8889 * Database: arduino_ws0401 * Table:

+ オプション

				seats_index	seats_on
<input type="checkbox"/>	編集	コピー	削除	1	1
<input type="checkbox"/>	編集	コピー	削除	2	0
<input type="checkbox"/>	編集	コピー	削除	3	0
<input type="checkbox"/>	編集	コピー	削除	4	0
<input type="checkbox"/>	編集	コピー	削除	5	0
<input type="checkbox"/>	編集	コピー	削除	6	0
<input type="checkbox"/>	編集	コピー	削除	7	0
<input type="checkbox"/>	編集	コピー	削除	8	0
<input type="checkbox"/>	編集	コピー	削除	9	1
<input type="checkbox"/>	編集	コピー	削除	10	0
<input type="checkbox"/>	編集	コピー	削除	11	0
<input type="checkbox"/>	編集	コピー	削除	12	0
<input type="checkbox"/>	編集	コピー	削除	13	0
<input type="checkbox"/>	編集	コピー	削除	14	0
<input type="checkbox"/>	編集	コピー	削除	15	0
<input type="checkbox"/>	編集	コピー	削除	16	0
<input type="checkbox"/>	編集	コピー	削除	17	0
<input type="checkbox"/>	編集	コピー	削除	18	0
<input type="checkbox"/>	編集	コピー	削除	19	0
<input type="checkbox"/>	編集	コピー	削除	20	0

③ 値を0 or 1に変更してみる

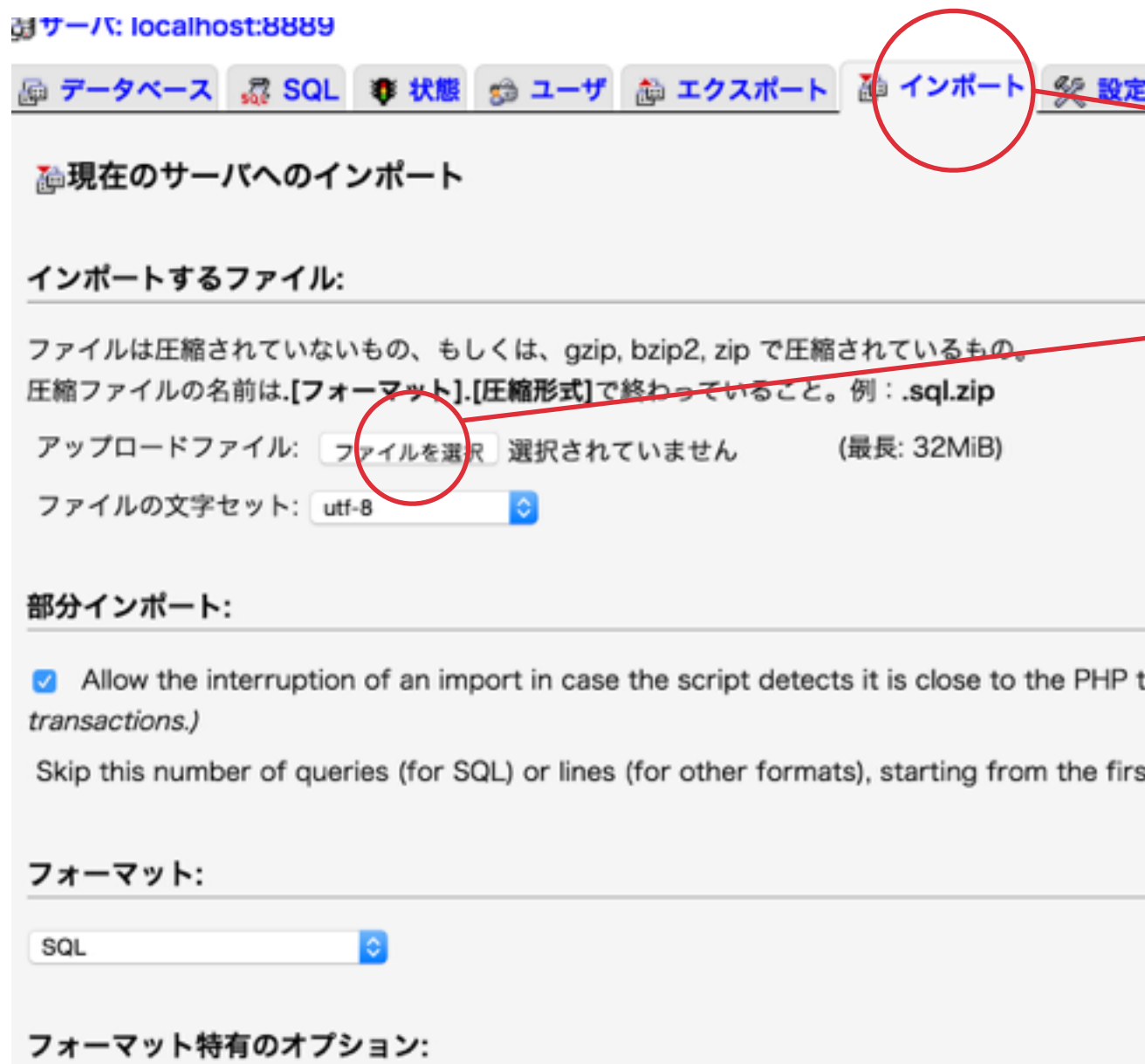
⑤ 座席表プログラムに変化があったか確認する

データベース

室温表示データベースのインポート

温度表示プログラムのデータベースを準備する

②arduino_ws0402.sqlをインポートする



The screenshot shows the phpMyAdmin interface with the 'Import' tab selected. The 'Import' button in the top navigation bar is circled in red, with a green arrow pointing to it from the annotation '①「インポート」を開く'. Below the navigation bar, the 'Import' section is titled '現在のサーバへのインポート'. Under 'Importするファイル:', there is a text box for 'アップロードファイル:' with a red circle around the 'ファイルを選択' button, and a green arrow pointing to it from the annotation '②sqlファイルを選択する'. The 'ファイルの文字セット:' is set to 'utf-8'. The '部分インポート:' section has a checked checkbox for 'Allow the interruption of an import in case the script detects it is close to the PHP timeout.' and a text input for 'Skip this number of queries (for SQL) or lines (for other formats), starting from the first'. The 'フォーマット:' section has a dropdown menu set to 'SQL'. The 'フォーマット特有のオプション:' section is partially visible at the bottom.

サーバ: localhost:8889

データベース SQL 状態 ユーザ エクスポート インポート 設定

現在のサーバへのインポート

インポートするファイル:

ファイルは圧縮されていないもの、もしくは、gzip, bzip2, zip で圧縮されているもの。
圧縮ファイルの名前は、[フォーマット].[圧縮形式]で終わっていること。例：.sql.zip

アップロードファイル: ファイルを選択 選択されていません (最長: 32MiB)

ファイルの文字セット: utf-8

部分インポート:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout.
(transactions.)

Skip this number of queries (for SQL) or lines (for other formats), starting from the first

フォーマット:

SQL

フォーマット特有のオプション:

①「インポート」を開く

②sqlファイルを選択する

データベース

室温表示データベースにデータを追加する

②「挿入」をクリック

③追加するデータを入力する

Recent Favorites

- New
- airmeeting
- airmeeting_
- airmeeting
- arduino
- arduino_ws0401
- arduino_ws0402
- New
- SeatTemps**
- information schema

カラム	データ型	関数	NULL	値
seattemps_index	int(11)			
seattemps_seatid	int(11)			1 座席番号
seattemps_temp	float			245 実際の温度の10倍を入力する
seattemps_date	date			日付を選択する

☒ 無視

カラム	データ型	関数	NULL	値
seattemps_index	int(11)			
seattemps_seatid	int(11)			

実行

①「SeatTemps」をクリック

④「実行」をクリック

室温表示プログラムを実行してみる

- 「04」フォルダの「**ArduinoWS04_Web02**」をデスクトップにコピー
- http://localhost:8888/ArduinoWS04_Web02/index.php

通信と認証

REQUEST AND CERTIFICATION

GETとPOST

HTTP通信で値を送信する2種類の方法

- GETリクエスト

- データをURLに付加して送る

例) http://localhost:8888/index.php?seat_id=0

- 文字数制限あり

- POSTリクエスト

- テキストだけでなくバイナリも送信できる
- 文字数制限なし

BASIC認証とDIGEST認証

リクエストにパスワード認証をかける

- BASIC認証

- ページリクエストにIDとパスワードを設定する

- DIGEST認証

- IDとパスワードをハッシュ化して送るため、
BASIC認証よりも安全性は高い

BASIC認証

「.htaccess」 ファイル

```
AuthUserFile / (フルパス) /.htpasswd  
AuthGroupFile /dev/null  
AuthName "Please enter your ID and password"  
AuthType Basic  
Require valid-user
```

「.htpasswd」 ファイル ※生成サイトを利用すると便利

ID:パスワード (ハッシュ化したもの)

.htpasswd生成サイト

<http://www.luft.co.jp/cgi/htpasswd.php>

URLにBASIC認証を埋め込む

http://ユーザ名:パスワード@www.hoge.com/

http://arduino:sample@localhost:8888/sample.php

ユーザ名 パスワード

補足資料

SUPPORT DOCUMENTATION

LAMP以外の構成

- サーバOS

OS X Server、Windows Server

- Webサーバソフトウェア (httpd)

Nginx、Node.js

- データベース

SQL (SQL Server、Oracle Database、PostgreSQL)

NoSQL (MongoDB)

- サーバ言語

Python、Perl、Ruby、JavaScript