



実践課題制作

2019/05/13

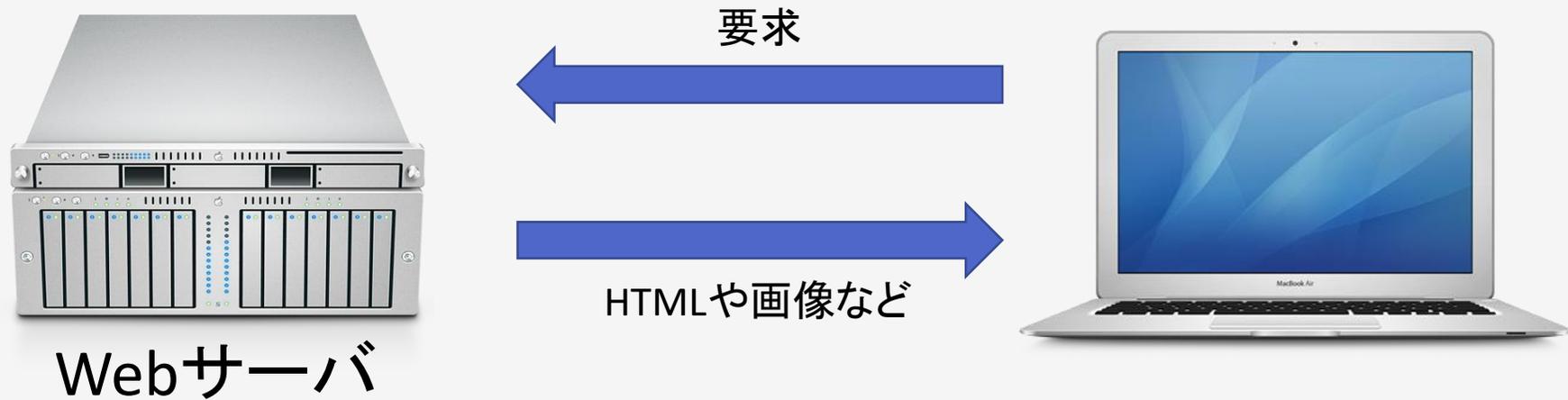
Kazuma Sekiguchi

class@cieds.jp

こんな場合どうするか

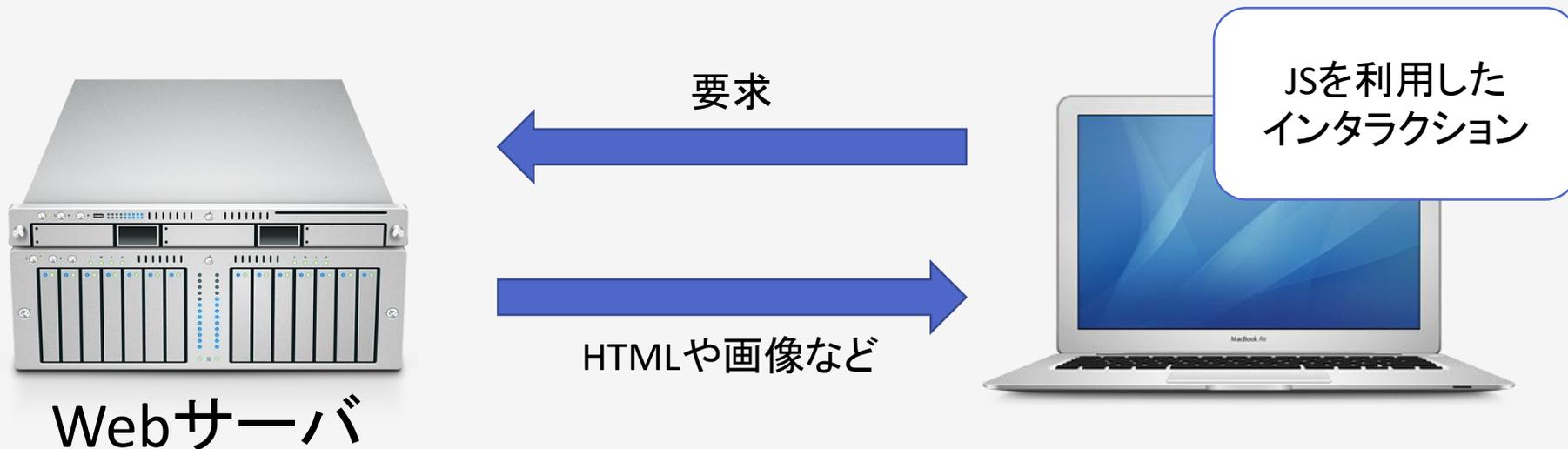
- フォームで入力してもらい確認画面を表示
 - ユーザが入力するものはマチマチなためあらかじめ作成しておくことは不可能
- ページ数が膨大なサイト
 - 表示のフォーマットが決まっているなら、動的に生成して出力してあげた方が効率的
- 検索ページ
 - どんな言葉が検索されるか分からない

サーバサイドとクライアントサイド



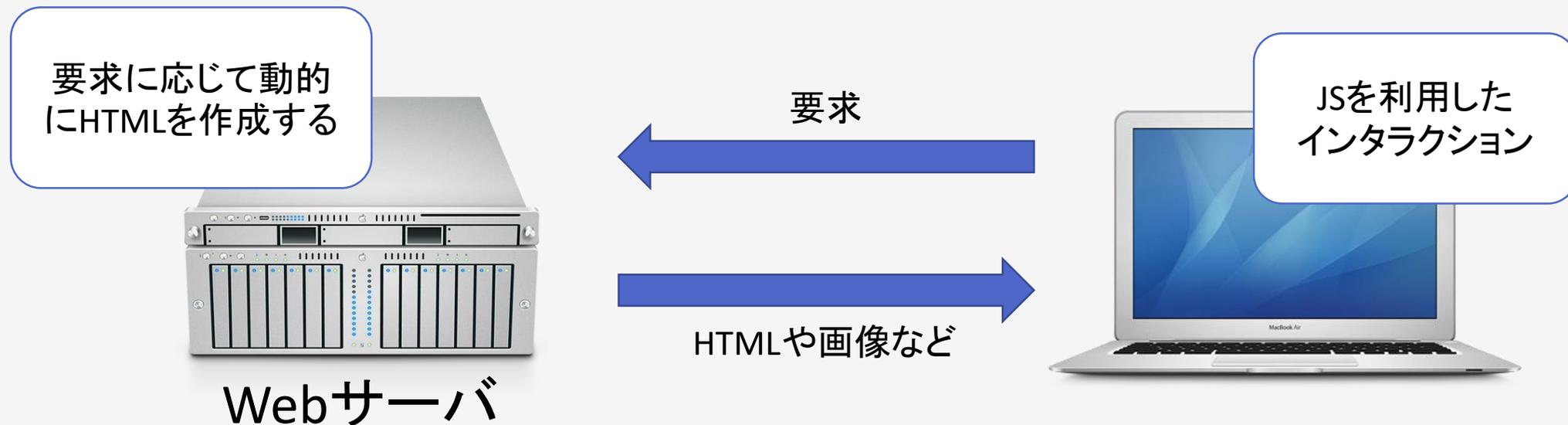
- URLを入力してアクセス
 - WebサーバというPCが要求に応じてファイルをクライアントに送り出す
 - (ファイルを送り出す) サービスをする = サーバー

サーバサイドとクライアントサイド



- ファイルにJSが合った場合、クライアント側で、JSが実行される
 - クライアント側で実行される = クライアントサイドプログラム

サーバサイドとクライアントサイド



- 要求に応じてHTMLを生成する
 - HTMLを生成→何らかのプログラミング言語を用いる
 - サーバー側でHTMLを生成する=サーバーサイド
 - 生成するのはHTMLだけとは限らない

動的と静的

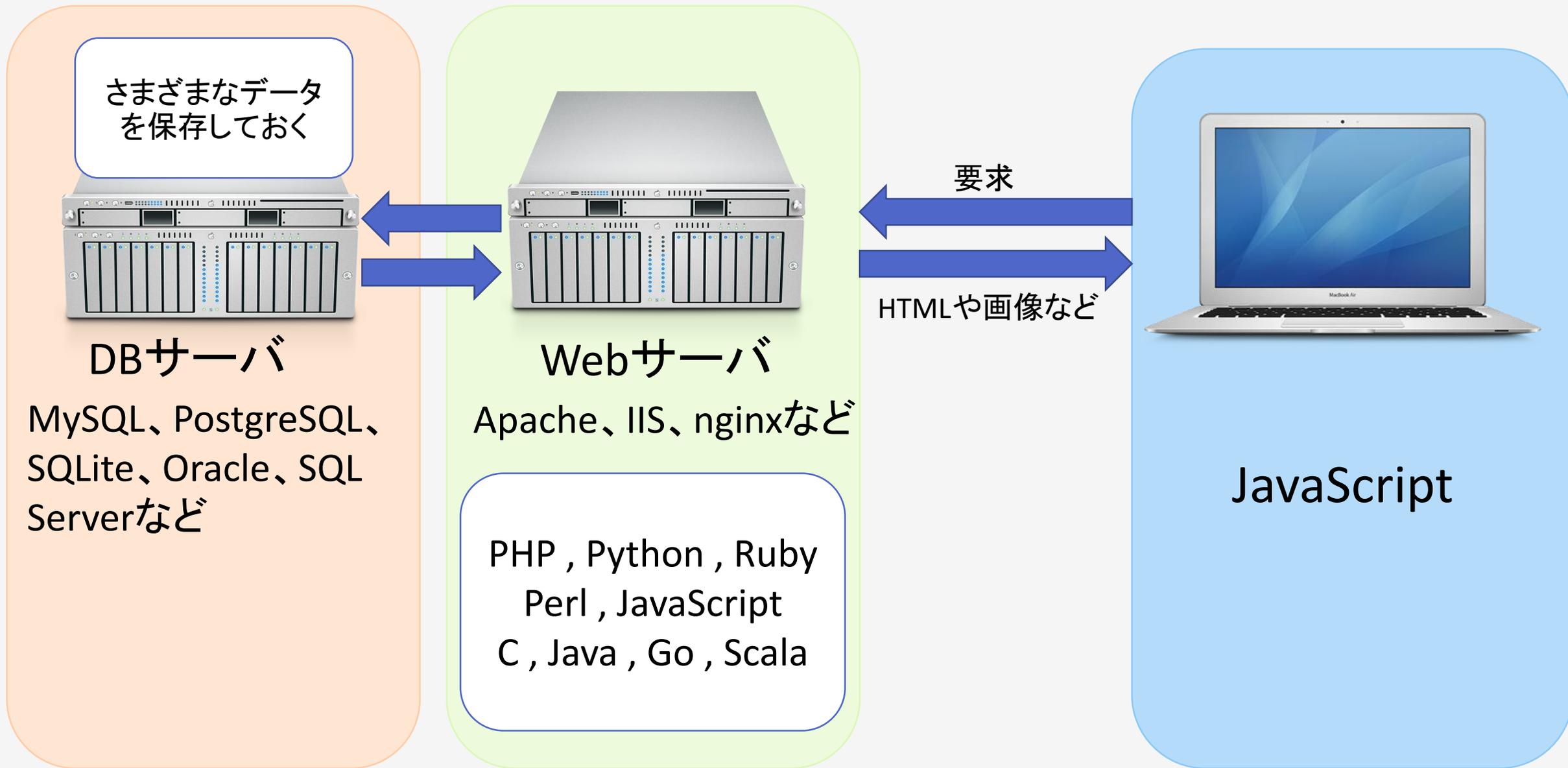
- 動的

- アクセスがあるごとにHTMLをプログラムに基づいて作成し、クライアントに対して送信する

- 静的

- あらかじめPCなどでHTMLを作成し、FTPなどでアップロードをしておき、クライアントのアクセスがあったら、作成済みのHTMLを送信する

サーバサイドとクライアントサイド



動的生成

- ユーザの操作や動きに合わせてその場その場でHTMLを生成する
 - 検索結果画面などを用意しておく必要がない
 - リアルタイムに結果を反映可能
- プログラミング言語を用いて作成する
- 表示するデータは予めデータベースに保存しておく
 - そこから必要なデータを取り出してくる

JSとの比較

	JavaScript	PHP
実行タイミング	ユーザーイベント連動	読み込み時に実行
動作	ブラウザ、Nodeなど	PHP上で実行
変数	var , let , constを付けて宣言	\$を付ける。宣言は不要
データ保存	localStorage , IndexedDb	何らかのサーバーDB、ファイル
ユーザーによる改変	可能	不可能

HTMLとPHP

- PHPはHTMLを生成可能
 - PHP構文内で無ければ、PHPファイル内に通常通りHTMLタグを記述していけばOK
- PHP構文内部で利用する場合は、`print()`関数などを用いて、シングルクォートで括ればそのまま出力される
- PHPを利用するためには、実行環境としてPHP（というソフト）が必要
 - 通常はWebサーバと組み合わせて利用する

```
<h1>PHPでのHTML</h1>
```

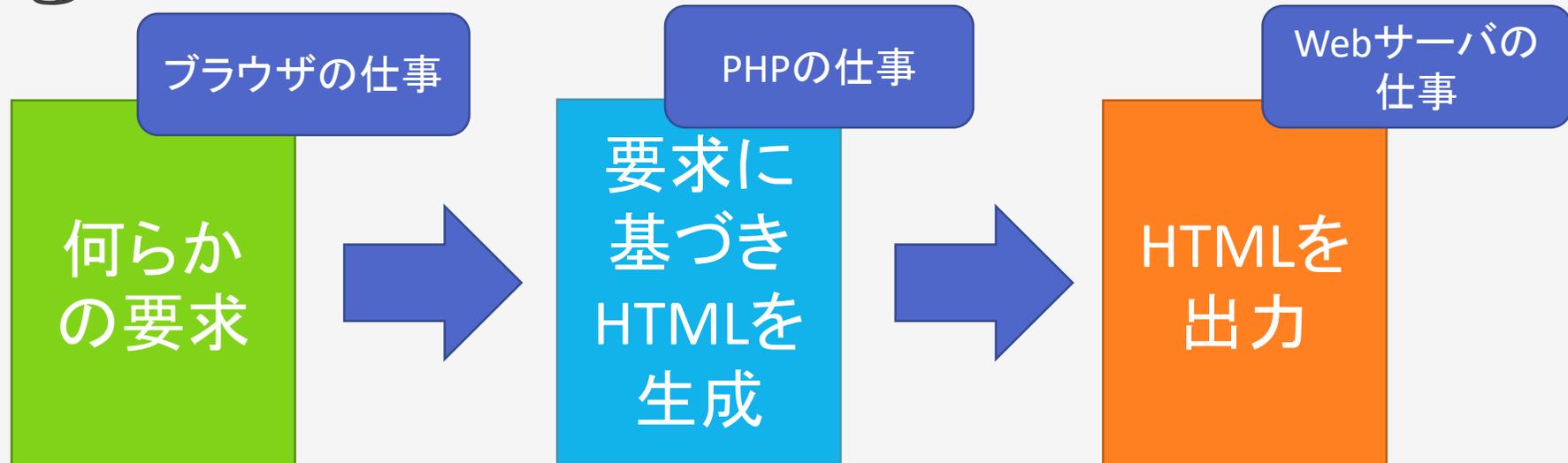
```
<?php
```

```
    print('<h2>ここに書くなら、print()関数を利用する');
```

```
?>
```

PHPの利用

- 通常はユーザから送られたデータを元にしてHTMLを動的に生成する



- データが送られてこない場合、静的なHTMLでもOK
 - データに応じて変化させる必要がある場合、PHPなどを利用する

Use PHP

How to make program with PHP



Webサーバ環境構築

- PHPとかMySQLを使うためには環境が必要
 - レンタルサーバで借りるのが手っ取り早いけどテストなどもしないでそのままアップロードは危険
 - セキュリティホールになる可能性がある
 - ローカル環境でテストできるようにしておくことは必要
 - 速度の検証はできない

一番多い構成

- Linux（またはBSD）：OS
 - Apache：Webサーバ
 - PHP：アプリケーション実行環境
 - MySQL：データベース
-
- 大体のレンタルサーバなどもこういう構成
 - 採用例も非常に多いため、情報が多い
 - 安定度も非常に高い
 - ソフトが全て無料で手に入る
 - ある程度古いマシンでも動作する

テスト環境としてお薦め

- XAMPP（ザンプ）などの一括でインストールしてくれるソフトを利用する
 - WindowsなどにApache , PHP ,MySQLをインストールすることは可能
 - ただしかなり面倒（設定が）
 - しかも上手く動かない率が高い
- ローカル環境でウェブ上と同じWebサーバーを動かすことが可能
 - ほとんど無設定で可能

Apacheなどの設定

- Linuxなどで動作するアプリは設定をテキストファイルで行う
 - テキストファイルに所定の書式で記述することで設定完了
 - GUIでも設定できる場合があるが、基本はエディタで設定をしていく
- 設定後は保存をして、ソフトを再起動する
 - 上手く起動しなかったら、ログを見て、エラーを確認する
 - ほとんどが書式エラーか文法エラー
 - 再起動するまで設定は反映されないので注意

サーバー経由での確認

- Xamppなどから確認するためには、Xamppフォルダー内にあるhtdocsフォルダー内に英数字でフォルダーを作り格納する
- 確認時にはブラウザでhttp://127.0.0.1/フォルダー名を入力しアクセスすればOK
 - フォルダーに格納したファイルをダブルクリックして開いてもサーバーを通していないので注意
 - PHPプログラムはサーバーを通さないと実行されない
 - そのままソースが出てくるだけ

MAMP

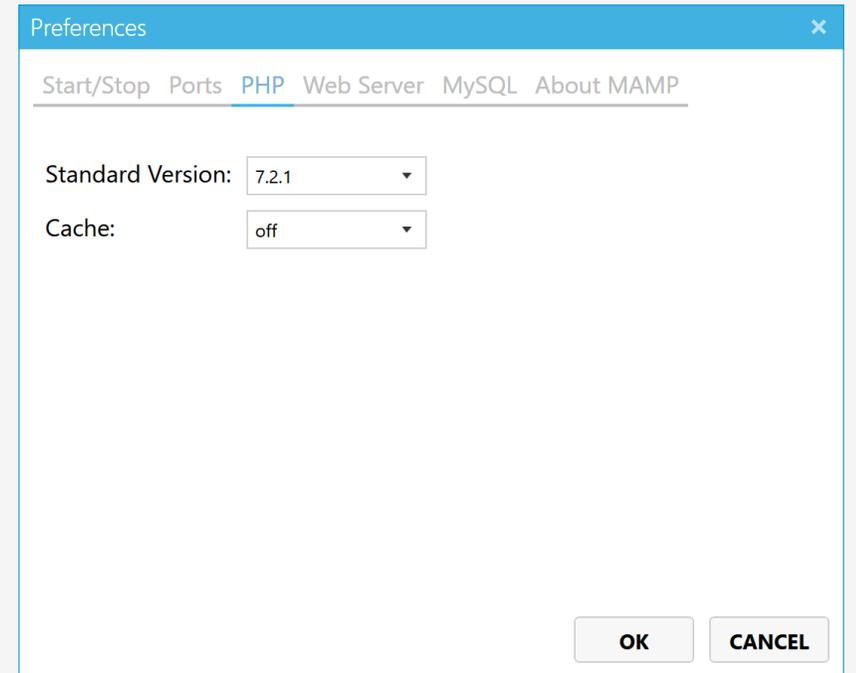
開発環境の設定 (Windows, Mac)

設定変更 (Macの場合)

- メニューバーからPreferencesを選択
- Portsを選択し、「Set Web & MySQL ports to 80 & 3306」をクリック
- 「Start Servers」をクリックし、Apache ServerとMySQL Serverの横に緑がともればOK

設定変更

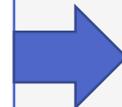
- Macの場合
「PHP」をクリックし、
「Standard Version」で7.1.8を
選択、「Cache」で「off」が選択さ
れていることを確認してOKをクリッ
ク



php.iniの設定(Macの場合)

472行目付近

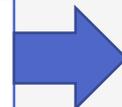
`display_errors = Off`



`display_errors = On`

665行目付近

`post_max_size = 8M`



`post_max_size = 32M`

データ量として受け入れる最大サイズ。ファイルサイズではなく、パケットの総合計サイズになる

910行目付近

`date.timezone = "Europe/Berlin"`



`date.timezone = Asia/Tokyo`

時間を出すときに利用される。日本であれば、GMT+9なので、タイムゾーンを変えないと変な時間が使われる

php.iniの設定 (Macの場合)

1597行目付近以下

文字コードの設定。変更しないと文字化けする

```
;mbstring.language = Japanese
```

```
mbstring.language = Japanese
```

```
;mbstring.internal_encoding = EUC-JP
```

```
mbstring.internal_encoding = UTF-8
```

```
;mbstring.http_input = auto
```

```
mbstring.http_input = pass
```

```
;mbstring.http_output = SJIS
```

```
mbstring.http_output = pass
```

```
;mbstring.encoding_translation = Off
```

```
mbstring.encoding_translation = Off
```

```
;mbstring.detect_order = auto
```

```
mbstring.detect_order = UTF-8,SJIS,EUC-  
JP,JIS,ASCII
```

php.iniの設定 (Macの場合)

- 変更をしたら上書きして保存する
- ファイルを保存したらMAMPのパネルから「Stop Servers」を押してサーバーを停止
- 「Start Servers」を押して再度サーバーを起動
 - 問題なく起動すればOK
 - 起動しない場合、設定をどこか間違えている可能性があるため、再度確認する（セミコロンを外す、残しておくとか、スペルなど）

PHP

PHP Programming

ABC



PHP (1)

- PHP Hypertext Preprocessorの略
- HTML生成に特化したプログラミング言語
 - 作成したプログラムはほとんどブラウザで確認する
- ほとんどのサーバOS上で動作する
- 比較的簡単に書ける
- データベースとの連携が簡単
- 採用例が多い



PHP (2)

- デメリット

- セキュリティが脆弱になりやすい
 - プログラミングの際に注意が必要
- 関数の命名規則が統一されていない
 - 似たような関数が全然違う名前だったりする
- ほかにいろいろなと批判も多い

PHP (3)

- 変数 (配列を含む)
- 関数
- 制御
- クラス

の4つから構成される

書式

- <?phpから始まり?>で終わる
 - HTMLで言う<body>タグと同じ
- 空の行などは一切意味がない
 - 見やすいように適宜改行を行って良い
 - スクリプトによっては、ファイルサイズを小さくするため、改行、スペースを削除する場合もある
- 1行で一つの動きを書く
- 「;」で行の終わりを示す

```
<?php
    print('PHPで文字を表示します');
?>
```

変数

- 変数名には「\$」をつける

```
$X = 5;
```

- この中のXが変数。Xには5が入る
 - 変数は自由に値を格納できる箱みたいなもの
-
- 変数名は自由に設定できる
 - 但し、予め別の役割が決められているものは使用不可（予約語）
 - 変数名の最初は英字で始まること。数字は文頭には使えない
 - 「_（アンダーバー）」から始まる変数は特殊な用途に利用される

変数（2）

- 変数は宣言せずに使用可能
 - 初期化できる（現在は初期化をすることが推奨）

```
$variables = "";
```

- 型は一切なし
 - \$txt = "" ;とすれば文字列型
 - \$txt = 0;とすれば数値

文字列連結

- 変数同士をくっつける
- 「.(ドット)」を使う

```
$txt_first = "PHP";  
$txt_second = "Program";  
$moji = $txt_first . $txt_second;
```

- \$mojiにはPHPProgramが入る
- 「.」を使う際には「" ."」とする

制御

- スクリプトの流れを制御するもの
 - if文を使用
 - elseifのみ空白が無いので注意

```
if(x > 3){  
    print("Xは3よりデカイ");  
}  
elseif(x > 2){  
    print("Xは2よりデカイ");  
}  
else{  
    print("Xは小さい");  
}
```

コメント

- スクリプトの実行時には無視される
- プログラムの動きなどを記述しておき、後から見直しやすいようにする

```
//1行だけコメント
```

```
/*この挟まれた間がコメント*/
```

関数

- 多数の関数が存在
- 引数を与えることで任意の動作が可能
- PHPの場合、関数が大量にあるため、上手く組み合わせる

```
<?php  
print("今の時刻を10回表示します");  
for($i = 0; $i < 10; $i++){  
    print(time());  
}  
?>
```

PHPの記述

- 単一のファイルに記述して保存
- 拡張子は**php**とする
 - アクセスされるときもhttp://example.jp/example.phpとしてアクセスされる
- サーバ上で動作するため、サーバにFTPなどでアップして実際にアクセスして動作を確認する
 - ローカルで開発環境を作成して確認しておく
- 文字コードは**UTF-8**で作成する
 - Shift-JISでは作らないこと

サーバ上のPHP環境情報

PHP Version 5.1.6 	
System	Linux serverlx01 2.6.18-92.1.10.el5 #1 SMP Tue Aug 5 07:41:53 EDT 2008 i686
Build Date	Jul 16 2008 19:54:37
Configure Command	"/configure" "--build=i686-redhat-linux-gnu" "--host=i686-redhat-linux-gnu" "--target=i686-redhat-linux-gnu" "--program-prefix=" "--prefix=/usr" "--exec-prefix=/usr" "libdir=lib" "--with-config-file-path=/etc" "--with-config-file-scan-dir=/etc/php.d" "--disable-debug" "--with-pic" "--disable-rpath" "--without-pear" "--with-bz2" "--with-layout=GNU" "--enable-exif" "--enable-ftp" "--enable-magic-quotes" "--enable-sockets" "--enable-syssem" "--enable-sysvshm" "--enable-sysvmsg" "--enable-sysvsem" "--with-libxml-dir=/usr" "--with-xml" "--with-system-tdzdata" "--with-apxs2=/usr/sbin/apxs" "--without-mysql" "--without-gd" "--without-odbc" "--dis
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
additional .ini files parsed	/etc/php.d/dbase.ini, /etc/php.d/dom.ini, /etc/php.d/eaccelerator.ini, /etc/php.d/gd.ini, /etc/php.d/ldap.ini, /etc/php.d/mbstring.ini, /etc/php.d/mcrypt.ini, /etc/
PHP API	20041225
PHP Extension	20050922
Zend Extension	220051025
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, http, ftp, compress, bzip2, compress, zlib, https, ftps
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, bzip2.*, zlib.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.1.0, Copyright (c) 1998-2006 Zend Technologies
with eAccelerator v0.9.5.2, Copyright (c) 2004-2006 eAccelerator, by eAccelerator

Powered By 

- サーバで動いているPHPの情報入手

```
<?php  
    phpinfo();  
?>
```

- 適当な名前でも保存して実行する