サーバーサイドスクリプトⅠ

０．環境構築

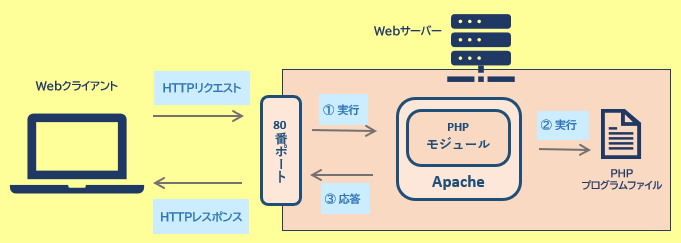
０－１． 前提条件

1. 授業では学校支給のWindowsを対象に行います。（授業ではMacには対応していません。）
2. XAMPPをインストールしている場合は、アンインストールしておいてください。

０－２． WebサーバーとPHP

PHPはWebアプリケーション開発を行うための技術です。

Webの仕組みは下図のようになっています。＊赤色部分の環境を構築していきます。



|  |  |
| --- | --- |
| Webクライアント | Webサーバー |
| Webサーバーの持つ情報や機能にアクセスして利用できるもののことで、狭義ではショッピングサイトなどを閲覧する際に使用しているブラウザを指します。 | クライアントからリクエストを受け取り、どのファイルをレスポンスとして送り返せばいいかを判断します。  通常の企業のサイトはWebサーバーを、オンプレミス（自社で保有管理）やクラウドで運用しています。  ※セキュリティが厳格なシステム（例えば金融系など）ほど、オンプレミスを使用していることが多いです。  ただし、近年ではクラウドに移行する企業やオンプレミスとクラウドを混合して使う企業も増えています。 |

|  |  |
| --- | --- |
| 警告 単色塗りつぶし | 前期のWebサイト制作（HTML）授業では、PHPはnt24という学校のサーバーを使用しましたが、後期のサーバーサイドスクリプト言語では、皆さんのパソコン（ローカル）上にWebサーバー（Apache HTTP Server）を構築しました。  実際の業務では、このようにローカル環境を構築し、開発とテストを行い、その後、テスト用のWebサーバーで検証するという流れになることが多いです。 |

０－３．　Apache

Apacheは、オープンソースのWebサーバーソフトウェアです。 汎用性が高く、WindowsやLinuxなどの様々なOSで動作させることができます。

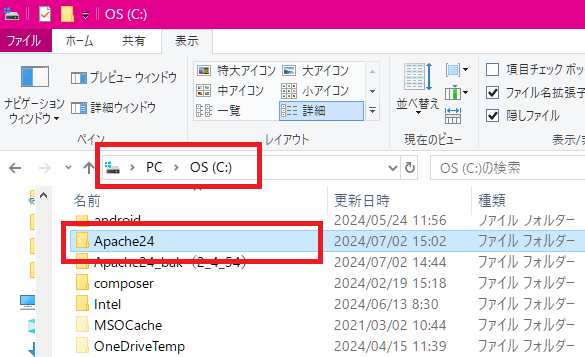
０－３－１．　Apacheのダウンロード

公式から紹介されているWindows版の配布サイトよりApacheのダウンロードをおこないます。

ダウンロードサイトへ移動し、「 Apache 2.4.60-240701 Win64 」のZIPファイルをダウンロードします。＊マイナーバージョンは違っていてもOKです。（例：Apache2.4.62など）

０－３－２．　Apacheの展開と配置

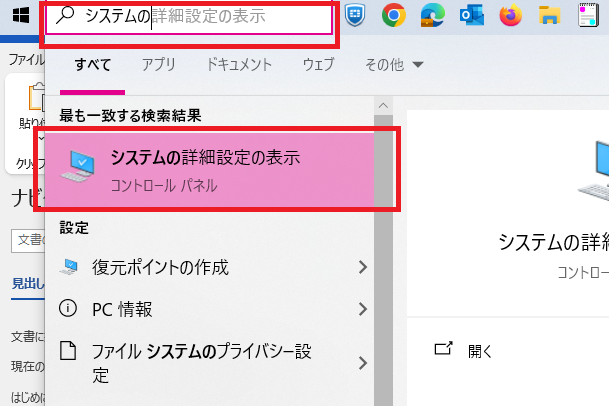
1. ダウンロードしたZIPファイル「 httpd-2.4.60-240701-win64-VS17.zip 」を解凍してください。
2. 解凍したフォルダ「 httpd-2.4.60-240701-win64-VS17 」内の「 Apache24　」フォルダをCドライブの直下に配置してください。



０－３－３．　Apacheのパスを通す

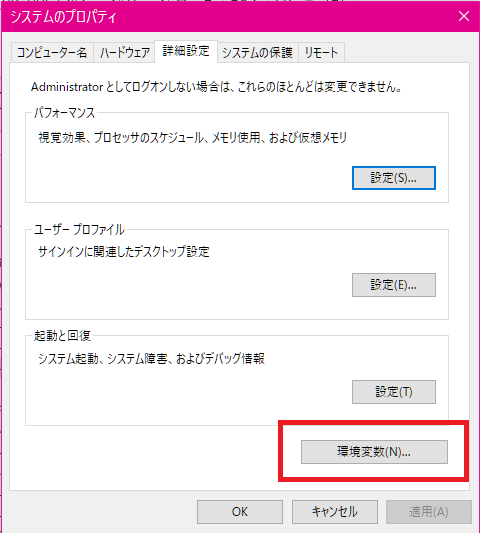
1. 「システムの詳細設定」を開く

検索で「システムの詳細」と打ち込む

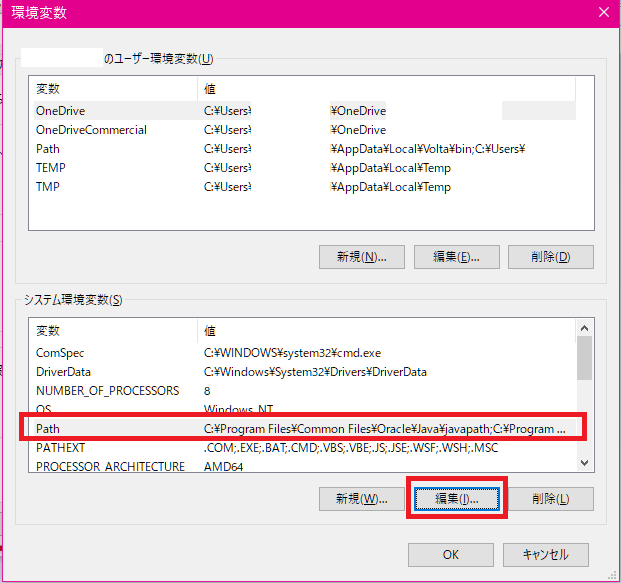


コントロールパネルの「システムの詳細設定の表示」が結果に表示されるので、選択する

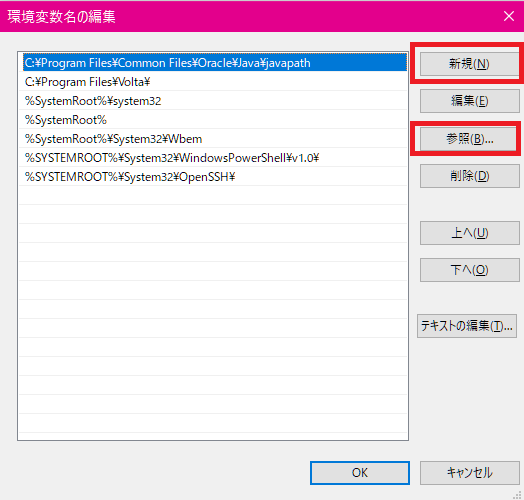
1. 「システムの詳細設定」から「環境変数」を押下



1. 「環境変数」　―　「システム環境変数（S）」のリスト内にある「Path」を編集



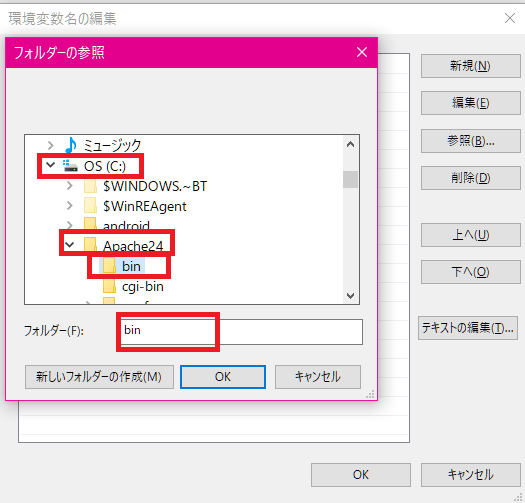
1. 環境変数の変数画面で、「新規」－「参照」の順に押下



**②**

**①**

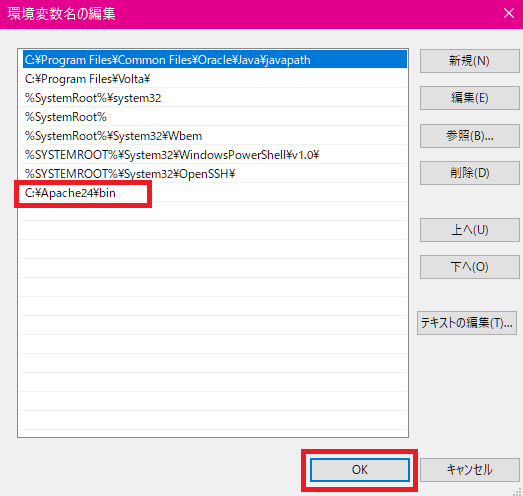
1. ダイアログで「C:\Apache24\bin」を選択し、「OK」押下



選択したフォルダ名となっていることを確認

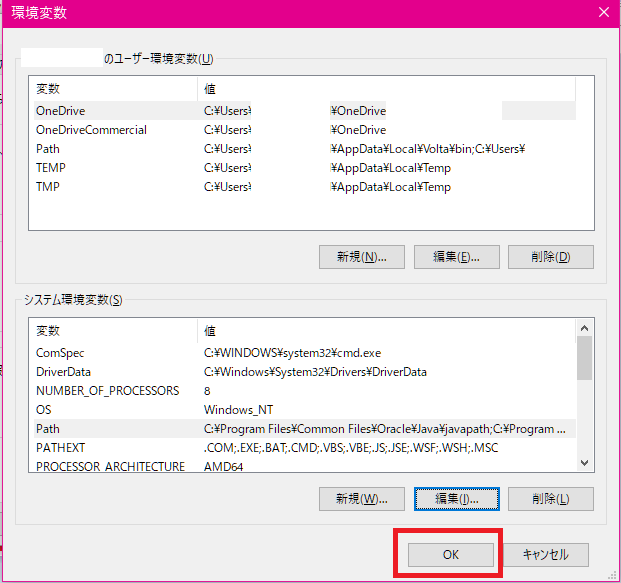
1. 環境変数名に追加されていることを確認後、「OK」押下

＊「OK」を押し忘れると適用されないので注意。



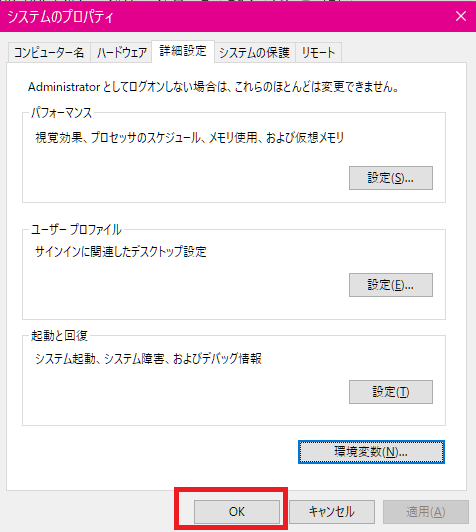
1. 環境変数ダイアログでも「OK」押下

＊「×」で画面を閉じると、設定が適応されないことがあるため注意。



1. システムの詳細設定（システムのプロパティ）画面も「OK」押下

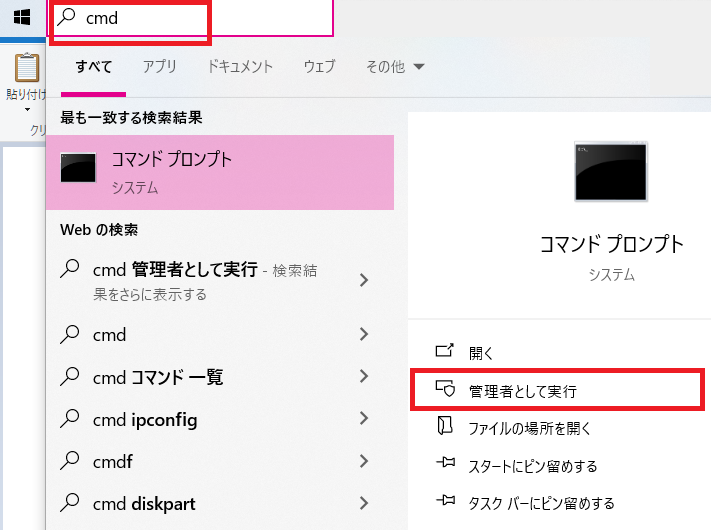
＊「×」で画面を閉じると、設定が適応されないことがあるため注意。



０－３－４．　Apacheのパスを確認

1. コマンドプロンプトを管理者で起動

①「ｃｍd」入力

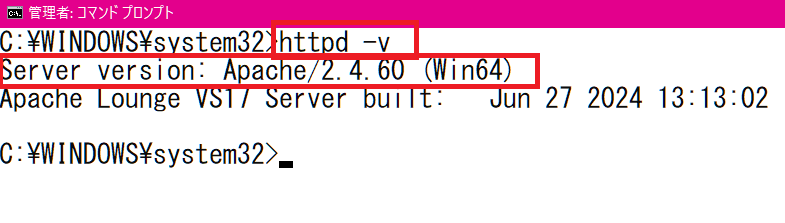


②「管理者として実行」を選択

1. コマンドで「httpd -v」と入力しEnterキー押下

＊バージョンが表示されることを確認。

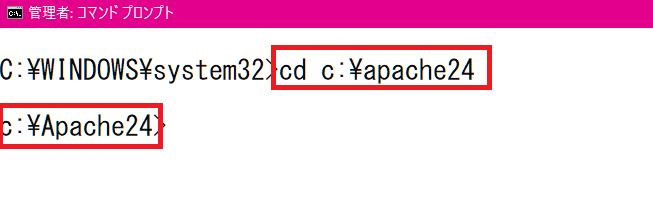
＊バージョンが表示されない場合は、「０－３－２」から設定を再確認すること。



コマンドプロンプトは次項で使用するので、開いておく。

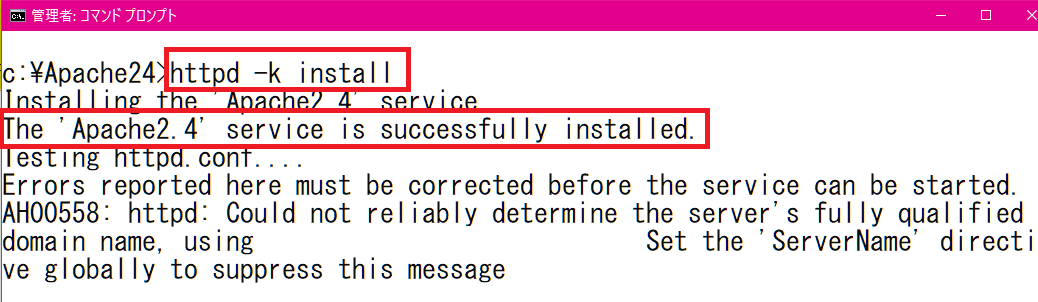
０－３－５．　Apacheをサービスへ登録

1. コマンドで「cd c:\apache24\bin」と入力しEnterキー押下（フォルダ移動）



1. コマンドで「httpd -k install」と入力しEnterキー押下

＊下図のように「The ‘Apache2.4’ service is successfully installed」と表示されていればOK。

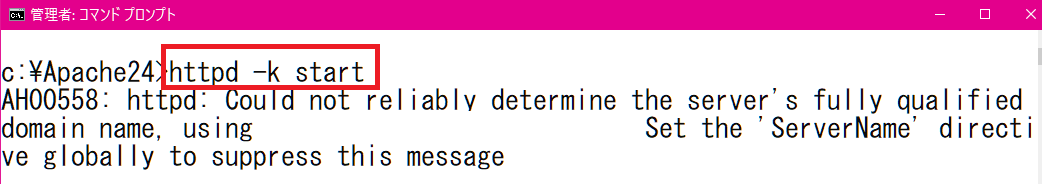


コマンドプロンプトは次項で使用するので、開いておく。

０－３－６．　Apacheを起動

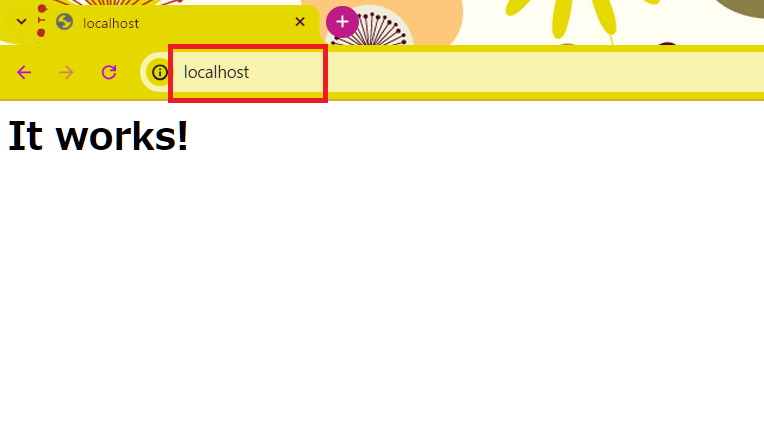
1. コマンドで「httpd -k start」と入力しEnterキー押下

＊エラーがでなければOK



1. ブラウザで「<http://localhost>」と入力。

＊「It Works!」と表示されればOK。（Apacheが正常に動作している。）



０－４．　PHP

授業ではPHP８．３を使用します。

０－４－１．　PHPのダウンロード

PHPのサイトへアクセス（<https://windows.php.net/download#php-8.2>）

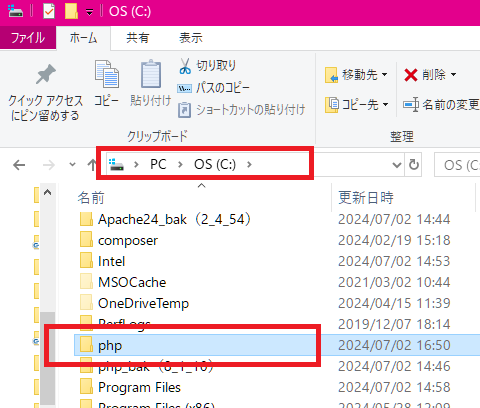
「VS16 x64 Thread Safe」の「ZIP」をダウンロード

＊「Non Thread Safe」と間違えないように、充分注意すること。

０－４－２．　PHPの展開と配置

1. ダウンロードしたZIPファイル「 php-8.2.22-Win32-vs16-x64.zip 」を解凍してください。
2. 解凍したフォルダ「 php-8.2.22-Win32-vs16-x64 」をCドライブの直下に配置してください。

＊フォルダ名は「php」へ変更する。

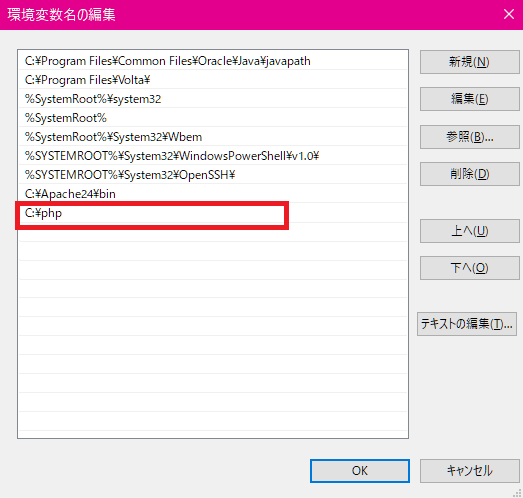


※ PHP8.2.のマイナーバージョンは上がっていきます。マイナーバージョンは最新のもので大丈夫です。

０－４－３．　PHPのパスを通す

Apacheの時と同じ手順で、環境変数を設定します。

1. 「システムの詳細設定」－「環境変数」－「システム環境変数」－「Path」を編集
2. 環境変数の変数画面で、「新規」－「参照」の順に押下し、「C:\php」を追加



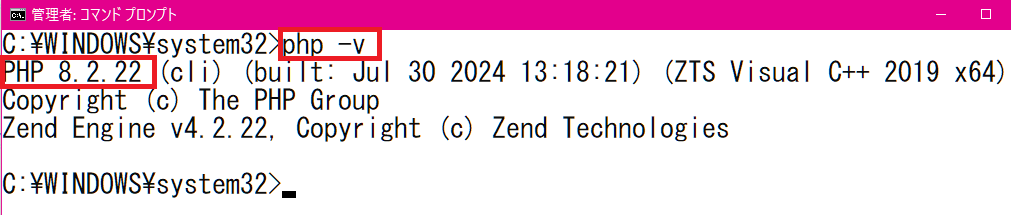
1. 追加が完了したら、順に画面のOKボタンを押下して、設定が適用されるようにすること。

０－４－４．　PHPのパスを確認

1. コマンドプロンプトを管理者で起動
2. コマンドで「php -v」と入力しEnterキー押下

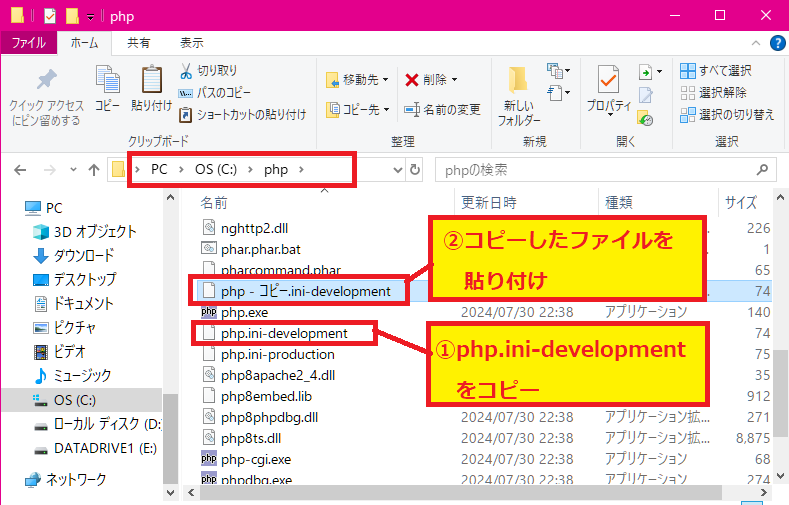
＊バージョンが表示されることを確認。

＊バージョンが表示されない場合は、「０－４－２」から設定を再確認すること。

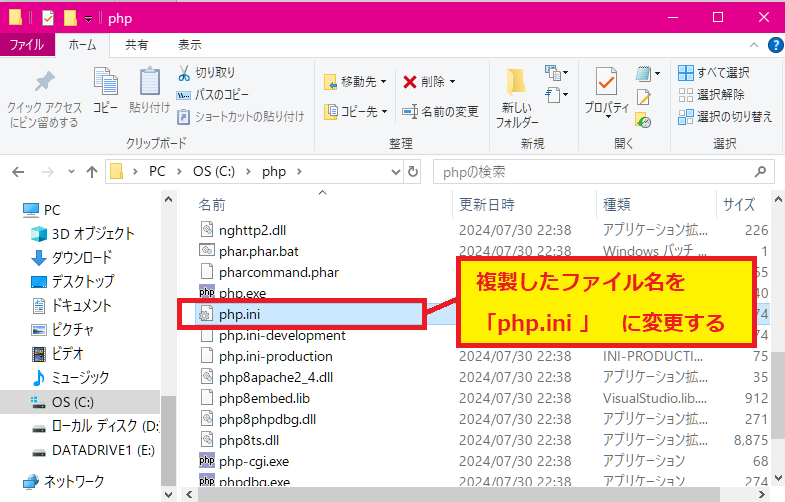


０－４－５．　php.ini の作成

1. 「C:\php」をフォルダ内の「 php.ini-development 」　ファイルをコピーして複製する。



1. 複製したファイルの名前を「 php.ini 」に変更する。



０－４－６．　php.ini の編集

「 php.ini 」ファイルは、PHPの設定ファイルになります。この「 php.ini 」で利用する拡張機能の有効化や各種設定などをおこないます。

エディタでファイルを開き、以下の編集を行います。（Sakuraエディタ推奨）

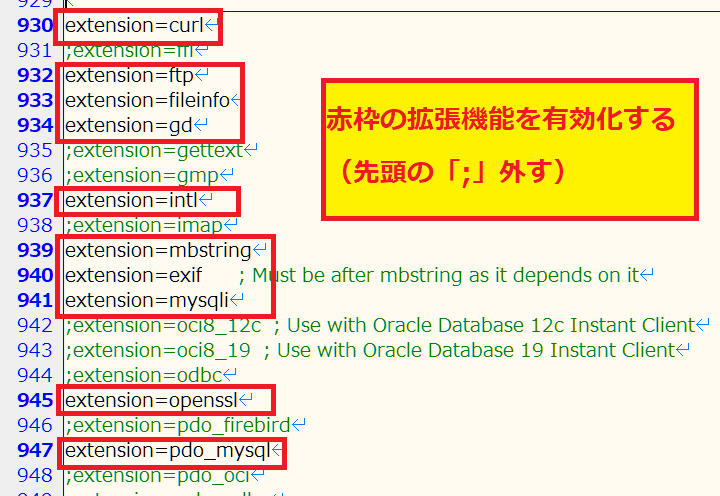
1. 765行目付近の拡張機能ディレクトリのパスを設定

＊php.iniでは先頭に「;」があると1行コメント扱いになります。

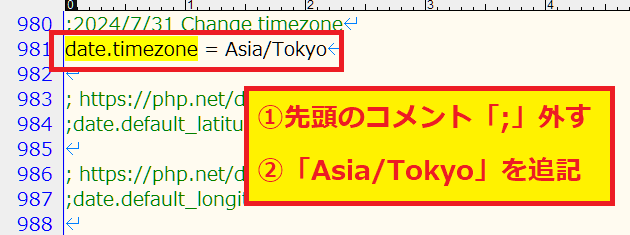
テキスト

自動的に生成された説明

1. 930行目付近からの拡張機能を有効化（コメント「;」を外す）  
   ※ 今回は授業ベースで必要となる拡張機能のみ有効化します。



1. 980行目付近のタイムゾーンを設定



以上、編集が完了したら、上書きしてファイルを閉じます。

０－５．　ドキュメントルートの設定

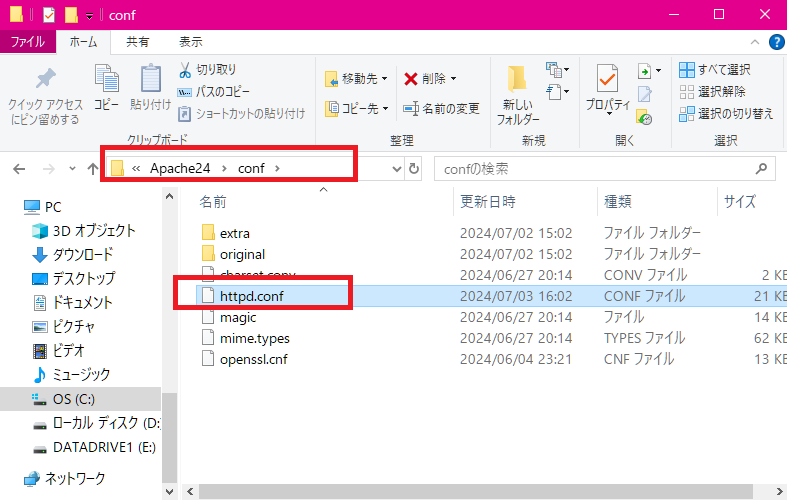
０－５－１．　ドキュメントフォルダの作成

Cドライブ直下に新規フォルダを作成して、名前を「 Sites 」に変更してください。

|  |  |
| --- | --- |
| 警告 単色塗りつぶし | ドキュメントフォルダの名前は、任意に決めることができます。「 www 」というフォルダ名も広く使われている名前となります。 |

０－５－２．　httpd.confの編集

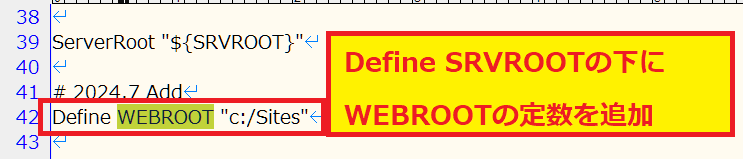
「 C:\Apache\conf 」フォルダ内にある「 httpd.conf 」ファイルを開き、編集を行います。



1. ルートディレクトリのパス

40行目付近 Define SRVROOTの下にWEBROOTの定数を追加

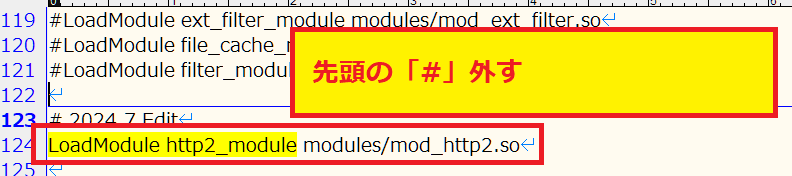
「Define WEBROOT "c:/Sites"」



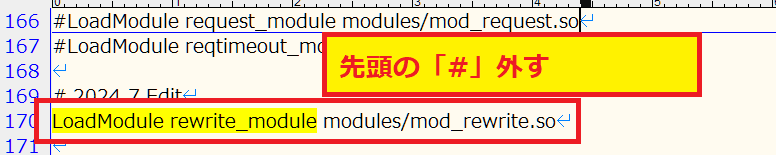
1. ロードモジュールの有効化（2箇所）

＊httpd.confでは先頭に「#」があると1行コメント扱いになります。

125行目付近「 LoadModule http2\_module 」のコメントを外す。



170行目付近「 LoadModule rewrite\_module 」のコメントを外す



1. PHPモジュールの追加と拡張子の登録

195行目付近（LoadModule群の下）に下記を追記

# 2024.7 Add-------------------------------------------------

# PHP8.2

LoadModule php\_module "C:/php/php8apache2\_4.dll"

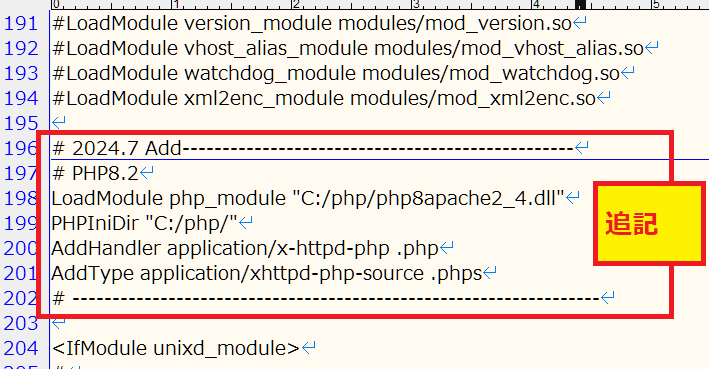
PHPIniDir "C:/php/"

AddHandler application/x-httpd-php .php

AddType application/xhttpd-php-source .phps

# ------------------------------------------------------------------

■追記例



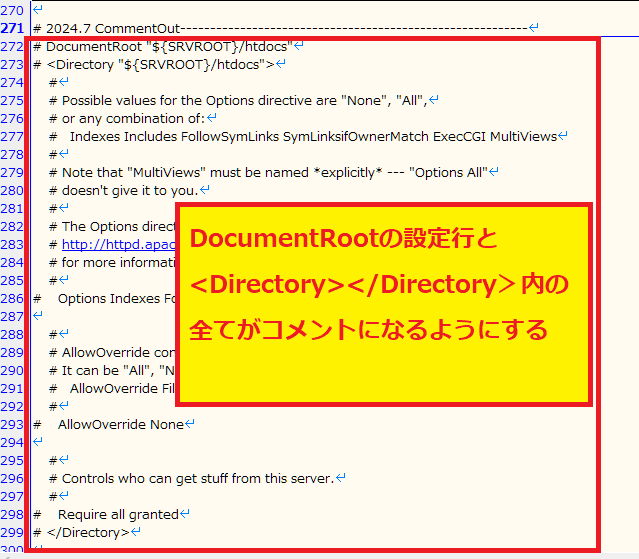
1. ServerNameの登録

245行目付近「 ServerName 」のコメントを外す



1. ルートディレクトリのパスと設定の修正

2７0行目付近 初期設定のコメント化



コメントアウトしたディレクトリ関連の下にCドライブ直下に作成した「 Sites 」フォルダへの設定を追加（300行目付近に下記を追記）

# 2024.7 Add

DocumentRoot "${WEBROOT}"

<Directory "${WEBROOT}">

Options Indexes FollowSymLinks MultiViews ExecCGI

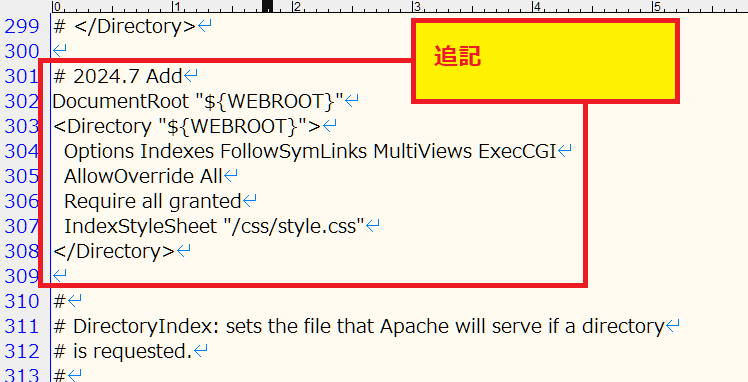
AllowOverride All

Require all granted

IndexStyleSheet "/css/style.css"

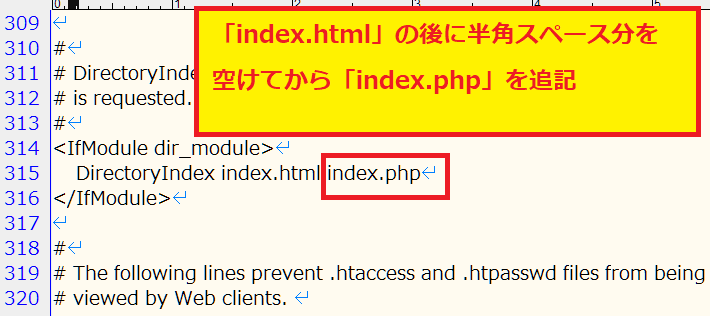
</Directory>

■追記例



index.phpも省略対象に追加

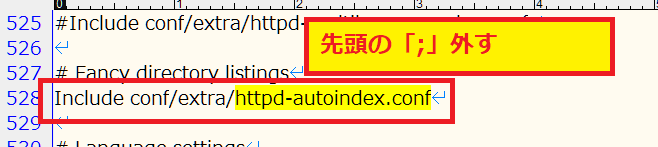
315行目付近 「 <IfModule dir\_module> 」内のDirectoryIndexにindex.phpを追記



1. IndexStyleSheetの設定を読み込む（コメント「#」を外す）

525行目付近 「 httpd-autoindex.conf 」をIncludeするコメントを外す

＊名前が似た設定があるので混同注意。



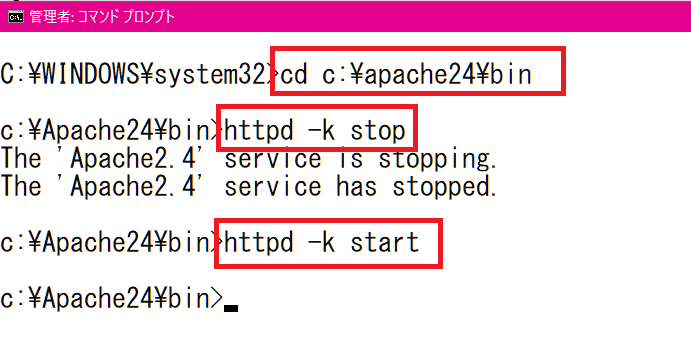
以上、編集が完了したら、上書き保存してください。

０－５－３．　ルートディレクトリの変更確認

「 httpd.conf 」の編集が完了したら、Apacheを再起動する。

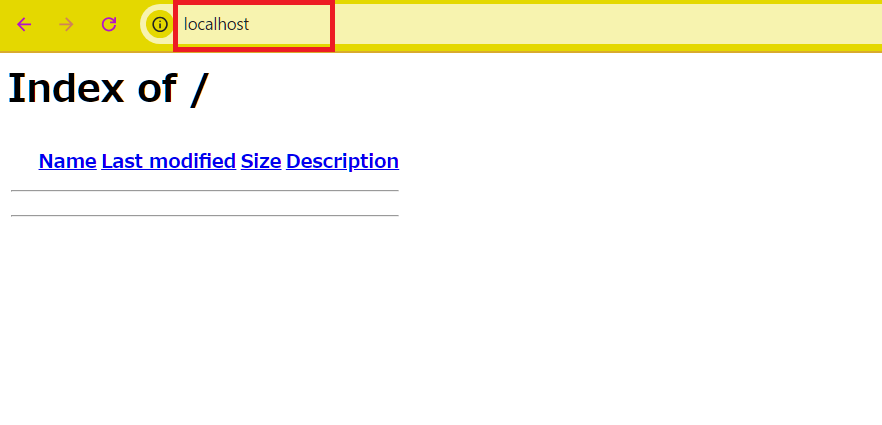
コマンドプロンプトを管理者で起動し、apacheのフォルダへ移動、apacheの停止と起動コマンドを実行する。

※ Apacheの再起動が出来なかった場合は、httpd.confの設定を再確認してください。



Apacheの再起動できたら、ブラウザの「 [localhost](http://localhost) 」のページを更新して、「 It works! 」からSitesフォルダへ変更されていれば、ルートディレクトリの変更は完了になります。

Sitesフォルダ内が表示されればOK。（現在フォルダは空なので下記のように表示される）



０－６．　PHPの設定を確認

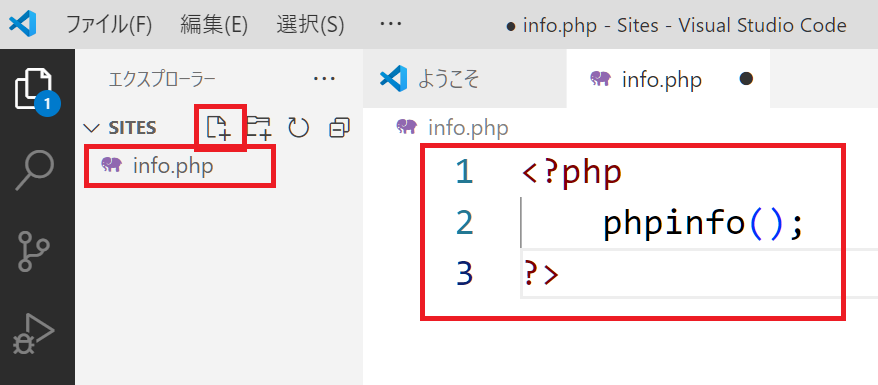
1. VSCodeでSitesフォルダを開く



1. 新規ファイルを作成する

「info.php」ファイルを作成し、phpinfo関数を記述する

＊記述したら、上書き保存する。上書き保存すると、下図3行目の「?>」は省略されるが、phpのコードのみの場合は問題ない。HTMLも記述する場合は省略できない。



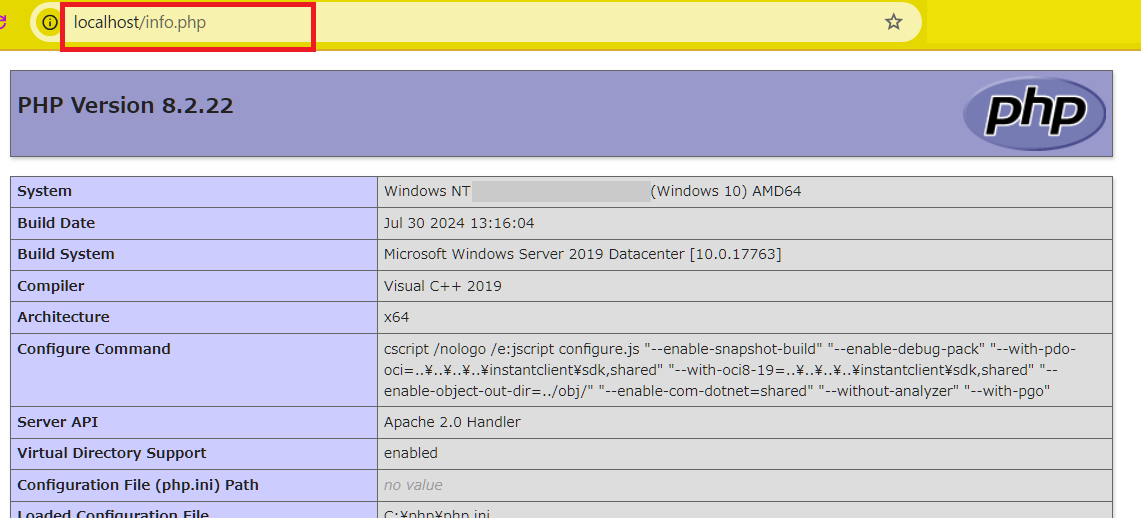
1. ブラウザで「<http://localhost>」にアクセスする

「info.php」ファイルが表示されるので、クリックする。



1. 「<http://localhost/info.php>」にアクセスでき、下図のように表示されていれば、環境設定したWebサーバーでPHPが処理されたことになります。

＊phpinfo()関数の実行結果。



以上で、PHPを動作させるローカル環境構築は終了となります。