

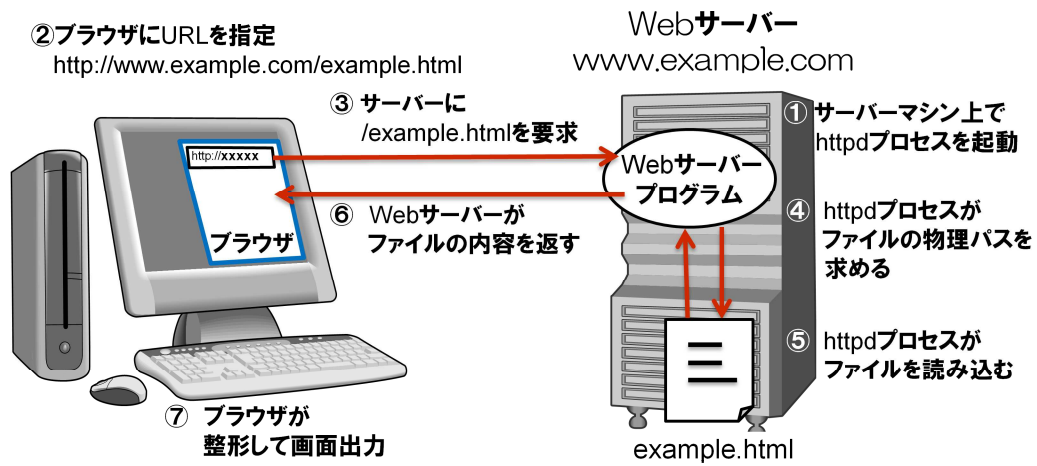
# Apacheスタートアップセミナー

NECラーニング株式会社  
テクノロジー研修事業部 堀  
<http://www.neclearning.jp/>

## Apacheとは

- Apache HTTP Server と呼ばれ、世界中で最も使われているWebサーバーソフトウェア
- Apacheライセンスのもとでソースが公開されている代表的なOSSのひとつ
- マルチプラットフォーム対応
  - UNIX/LinuxのみならずWindows上でも動作する

## クライアント(ブラウザ)とApache HTTP Serverの通信



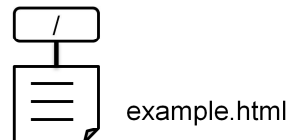
## 物理パスへの変換(DocumentRootディレクティブ)

### 1. ブラウザのアドレスバーに入力されたURL

`http://www.example.com/example.html`

### 2. Webサーバーに要求するファイルパス(仮想パス)

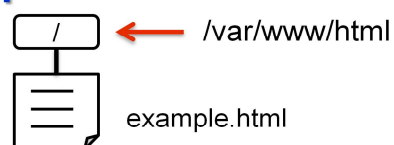
`/example.html`



### 3. Webサーバーが返送するファイルパス(物理パス)

`/var/www/html/example.html`

DocumentRoot  
ディレクティブで設定！



## ■確認しよう

1. ApacheのDocumentRootの設定を確認します。  
Apacheの設定ファイルは、`/etc/httpd/conf/httpd.conf`です。

```
# grep `DocumentRoot` /etc/httpd/conf/httpd.conf
DocumentRoot    "/var/www/html"
```

2. DocumentRoot に設定されている `/var/www/html` にHTMLファイルを配置します。

```
# cp example.html /var/www/html
# ls /var/www/html
example.html
```

3. Apacheを起動します。

```
# /etc/rc.d/init.d/httpd start
```

4. ブラウザのアドレスに、次のURLを入力して、アクセスできることを確認します。

`http://localhost/example.html`

## ディレクトリアクセスへの対応（1）

■ http://localhost/ というURLでアクセスしてきた場合  
何が起こる？

http://localhost/ → 仮想パスは「/」



物理パスは「**/var/www/html/**」  
ディレクトリを要求している！

実際にアクセスすると・・・。

### ■確認しよう

1. ブラウザのアドレスに、次のURLを入力して、アクセスします。

http://localhost/

testページが表示されることを確認します。

※ この設定は /etc/httpd/conf.d/welcome.conf 内で行われています。

## ディレクトリアクセスへの対応（2）

■ `http://localhost/` というURLでアクセスしてきた場合に `example.html` を返送するには？

`http://localhost/` → 仮想パスは「`/`」

物理パスは「`/var/www/html/`」

ディレクトリアクセスの場合、  
DirectoryIndexに設定されたファイルを探し、  
あればそのファイルを返送する。

### ■確認しよう

1. ApacheのDirectoryIndexの設定を確認します。

```
# grep `DirectoryIndex` /etc/httpd/conf/httpd.conf
DirectoryIndex index.html index.html.var
```

2. ApacheのDirectoryIndexの設定に「`example.html`」を追加し、ディレクトリアクセスが生じた際に、`example.html` も参照するように設定します。

```
# gedit /etc/httpd/conf/httpd.conf

DirectoryIndex index.html index.html.var example.html
```

※ DirectoryIndexに指定したファイルは、左側から順に検索されます。

3. Apacheの設定を変更したため、Apacheを再起動します。

```
# /etc/rc.d/init.d/httpd restart
```

4. 再度ブラウザのアドレスに以下のURLを入力し、`example.html`の内容が表示されることを確認します。

`http://localhost/`

## ユーザー認証(1)

### ■ 特定のユーザーのみアクセス可能なサイトに設定する .htaccessを配置する

http://localhost/ にアクセスしてきた場合に、認証ダイアログを表示し、登録されているユーザーのみアクセスを許可する。



ユーザー認証を行うように設定した .htaccess ファイルを、  
/var/www/html に配置する。

/var/www/html/.htaccess

AuthType	Basic
AuthName	"Secret Directory"
AuthUserFile	"/etc/httpd/members
require	valid-user

← パスワードファイル名

### ■ 確認しよう

1. .htaccessファイルを、/var/www/htmlにコピーします。

```
# cp .htaccess /var/www/html
# cat /var/www/html/.htaccess
```

2. ブラウザで以下のURLにアクセスします。

http://localhost/

※ユーザー認証用のダイアログボックスは表示されません。

## ユーザー認証(2)

### ■ .htaccessによる設定を許可する

アクセス制限したいディレクトリに .htaccessファイルを配置することにより設定の追加(上書き)が可能。  
ただし、httpd.conf 内で、許可した場合のみ適用される。



AllowOverrideディレクティブを設定する。

```
/etc/httpd/conf/httpd.conf  
<Directory /var/www/html>  
...  
AllowOverride AuthConfig  
...  
</Directory>
```

### ■確認しよう

1. /etc/httpd/conf/httpd.conf のAllowOverrideの設定を変更します。

```
# gedit /etc/httpd/conf/httpd.conf
```

270行目付近の以下の行を書き換えます。

```
# AllowOverride controls what directives may be placed in .htaccess files.  
# It can be "All", "None", or any combination of the keywords:  
# Options FileInfo AuthConfig Limit  
#  
AllowOverride None  
↓  
AllowOverride AuthConfig
```

2. Apacheを再起動します。

```
# /etc/rc.d/init.d/httpd restart
```

3. 一度ブラウザを終了し、再度起動したブラウザで以下のURLにアクセスします。

http://localhost/

※今度は、ユーザー認証用のダイアログボックスが表示されます。



## ユーザー認証(3)

### ■ パスワードファイルを作成する

.htaccessに設定したパスワードファイルをhtpasswdコマンドで作成する。

/var/www/html/.htaccess

AuthType	Basic
AuthName	"Secret Directory"
AuthUserFile	"/etc/httpd/members
require	valid-user

← パスワードファイル名



# htpasswd -c /etc/httpd/members mickey
New password: <u>system5</u>
Re-type new password: <u>system5</u>
Adding password for user mickey

### ■ 確認しよう

1. パスワードファイルを作成します。  
ファイル名は/etc/httpd/membersとし、mickeyユーザーをパスワード 'system5' で登録します。

```
# htpasswd -c /etc/httpd/members mickey
New password: system5
Re-type new password: system5
Adding password for user mickey
```

2. ブラウザで以下のURLにアクセスします。

http://localhost/

ユーザー認証用のダイアログボックスが表示されたら、ユーザー名にmickey、パスワードにsystem5を指定します。認証が成功すると、Webページが表示されます。

## 研修のご紹介(1)

### ■ UNIX/Linux Webサーバー構築 -Apache・SSL- (2日間)

#### ● コース概要

Linuxをベースに、インターネット／イントラネットを利用するうえで必要となるWebサーバーを構築するための知識を、マシン実習を通して修得します。

#### ● 到達目標

- Apacheを利用したWebサーバーの基本的な設定ができる。
- サイト内の任意のコンテンツに適切なアクセス制限を設定することができる。
- SSL接続を行うサイトの構築ができる。

#### ● 開催予定日

3/5 ~ 6 (東京)

※4月以降は、「Linux Webサーバー構築 -Apache・CMS・SSL-」になります。

## 研修のご紹介(2)

### Linuxインターネットサーバー構築基礎 -Web・DNS・メール- (2日間)

#### ●コース概要

Linuxを使用してインターネットサーバー(DNS、Web、メール)を構築するための基礎知識を、マシン実習を通して修得します。

#### ●到達目標

- Web、DNS、メールサーバーの概要を説明できる。
- DNSサーバーの基本的な設定ができる。
- Webサーバーの基本的な設定ができる。
- メールサーバーの基本的な設定ができる。

#### ●開催予定日

6/4 ~ 5 (大阪)	6/21 ~ 22 (東京)
8/9 ~ 10 (大阪)	8/28 ~ 29 (東京)