

`https://httpd.apache.org/`

- ▶ première version publique en avril 1995
- ▶ logiciel libre
- ▶ maintenu par la fondation Apache
- ▶ environ 20% des serveurs web en février 2024 sont des serveurs apache (source : `https://news.netcraft.com`)
- ▶ serveur web multi-sites (ou hôtes virtuels)
- ▶ architecture modulaire (un module = une extension du serveur web)
 - ▶ exemples de modules :
 - ▶ modules d'exécution de scripts php, python ou cgi (voir page 5)
 - ▶ module d'authentification
 - ▶ module de support des sessions
 - ▶ Les modules peuvent être chargés au lancement d'apache.

Fichiers et répertoires principaux (tous dans `/etc/apache2/`) :

- ▶ `apache2.conf` — fichier de configuration principal
 - ▶ `mods-available` — répertoire des fichiers de configuration des modules disponibles
 - ▶ `mods-enabled` — répertoire des fichiers de configuration des modules activés (liens symboliques vers les fichiers de `mods-available`)
 - ▶ `sites-available` — répertoire de fichiers de configuration des hôtes virtuels disponibles
 - ▶ `sites-enabled` — répertoire des fichiers de configuration des hôtes virtuels activés (liens symboliques vers les fichiers de `sites-available`)
- (valable sur `debian ≥ 9`)

Tous ces fichiers contiennent des **directives** qui modifient le comportement du serveur.

- ▶ DocumentRoot — répertoire contenant les ressources du serveur

```
DocumentRoot /var/www/html
```

- ▶ DirectoryIndex — page d'accueil d'un répertoire

```
DirectoryIndex index.html
```

- ▶ Include — inclusion d'un autre fichier de configuration

```
Include un_autre_fichier.conf
```

- ▶ User — utilisateur système qui lance le service (généralement, www-data)

```
User www-data
```

- ▶ ErrorDocument — document renvoyé par le serveur en cas d'erreur

```
ErrorDocument 404 /errors/404.html
```

- ▶ ErrorLog — fichier de journal des erreurs apache

```
ErrorLog /var/log/apache2.log
```

- un hôte virtuel $\approx$ un site web hébergé par le serveur
- fichiers de configuration dans `/etc/apache2/sites-available`
- Pour déclarer un hôte virtuel :

```
1 <VirtualHost ip:port>
2     ServerName nom.du.site
3     ...
4 </VirtualHost>
```

1. IP et port de réception des requêtes (* = n'importe laquelle/lequel)
 2. définit le nom du site
 3. on place ici les directives spécifiques à cet hôte.
- Exemple de deux sites hébergés accessibles par le port 80 :

```
<VirtualHost *:80>
    ServerName www.truc.com
    DocumentRoot /var/www/html/truc
</VirtualHost>
<VirtualHost *:80>
    ServerName www.machin.com
    DocumentRoot /var/www/html/machin
</VirtualHost>
```

- ▶ CGI = Common Gateway Interface
- ▶ méthode d'interface entre un serveur Web et des programmes générateurs de contenus
- ▶ par extension, un script exécuté par le serveur qui renvoie du contenu HTTP
- ▶ Un script CGI peut être écrit dans n'importe quel langage (python, bash, C, java, ...).

Un script CGI **doit** écrire sur la sortie standard :

1. une partie de l'en-tête du message HTTP renvoyé au client
 - ▶ Le type de contenu renvoyé (champ Content-Type) doit obligatoirement apparaître.
2. une ligne vide
3. le corps de la réponse HTTP (éventuellement vide)

La sortie d'un CGI correspond donc à un bout du message HTTP renvoyé par le serveur au client (partie de l'en-tête + corps de la réponse).

```
#!/bin/bash
```

```
echo "Content-Type:␣text/html"
```

```
echo ""
```

```
echo "<html>"
```

```
echo "<head><title>Informations</title></head>"
```

```
echo "<body>...</body>"
```

```
echo "</html>"
```