



Serveur HTTP ET PAGE WEB EN PYTHON

1.Introduction

Un serveur Web HTTP n'est rien d'autre qu'un processus en cours sur votre machine et fait exactement deux choses:

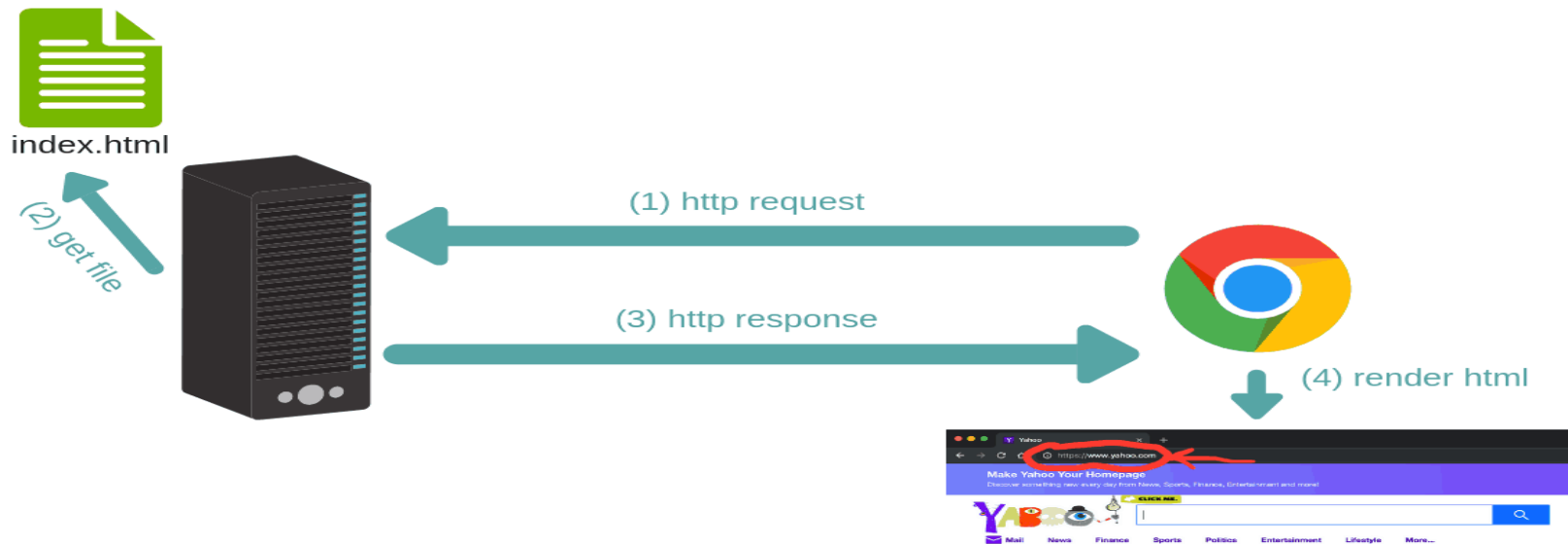
- 1- *Ecoute* les requêtes http entrantes sur une adresse de socket TCP spécifique (adresse IP et un numéro de port dont je parlerai plus tard)
- 2- *Gère* cette demande et renvoie une réponse à l'utilisateur.

ces demandes sont des documents au format HTML, que l'on peut consulter *via* un réseau (intranet ou internet) à l'aide d'un logiciel appelé browser web ou *navigateur* (ex : [Firefox](#), [Google Chrome](#), [Konqueror](#), [Internet Explorer](#), ...).

Les pages HTML sont installées dans les répertoires publics d'un autre ordinateur où fonctionne en permanence un logiciel appelé *serveur Web* ([Apache](#), [IIS](#), [Zope](#), ...). Lorsqu'une connexion a été établie entre cet ordinateur et le vôtre, votre logiciel navigateur peut dialoguer avec le logiciel serveur (par l'intermédiaire de toute une série de dispositifs matériels et logiciels dont nous ne traiterons pas ici : lignes téléphoniques, routeurs, caches, protocoles de communication ...).

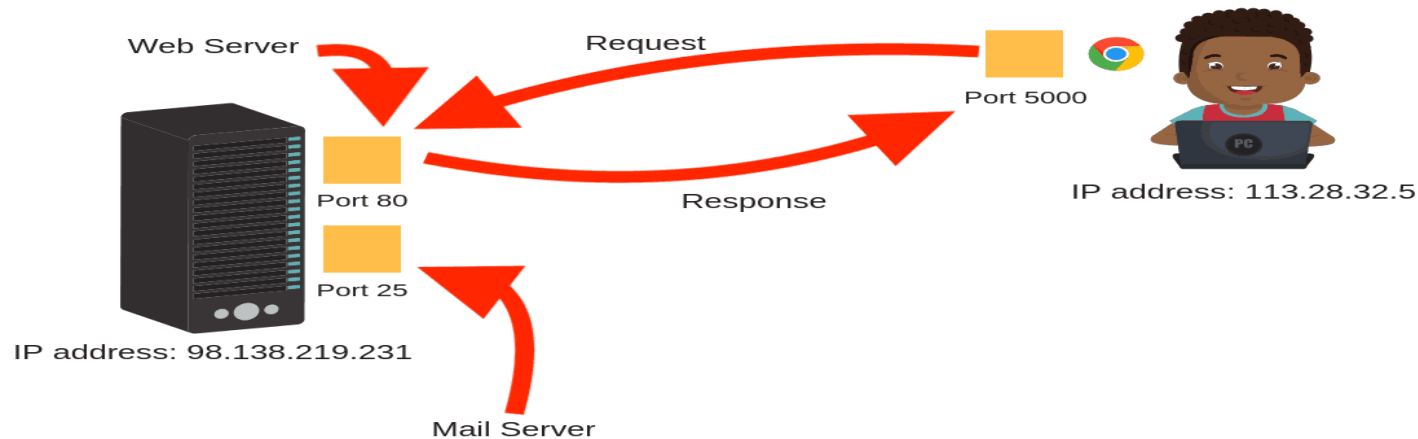
2. Protocole HTTP

Le *protocole* [HTTP](#) qui gère la transmission des pages web autorise l'échange de données dans les deux sens. Mais dans la grande majorité des cas, le transfert d'informations n'a pratiquement lieu que dans un seul, à savoir du serveur vers le navigateur : des textes, des images, des fichiers divers lui sont expédiés en grand nombre (ce sont les pages consultées) ; en revanche, le navigateur n'envoie guère au serveur que de toutes petites quantités d'information : essentiellement les adresses URL des pages que l'internaute désire consulter.



Chaque interaction avec la page d'accueil de Yahoo après celle-ci (par exemple, lorsque vous cliquez sur un lien) lance une nouvelle demande et une réponse exactement comme la première.

Pour rappel, la machine qui reçoit la demande http a un processus logiciel appelé un serveur Web en cours d'exécution. Ce serveur Web est responsable de l' **interception** de ces demandes et de leur traitement approprié .



Tout message http (qu'il s'agisse d'une demande ou d'une réponse) doit savoir comment atteindre sa destination.

Pour atteindre sa destination, chaque message http porte une adresse appelée *adresse TCP de destination* .

Et chaque *adresse TCP* est composée d'une **adresse IP** et d'un **numéro de port** .

L'adresse IP seule permettra au message HTTP d'arriver sur la bonne machine, mais vous avez toujours besoin du numéro de port pour que la demande HTTP parvienne exactement au serveur Web.

En d'autres termes, le serveur Web est une application réseau normale qui écoute sur un port spécifique.

Et la demande http DOIT être adressée à ce port.

Créer un simple fichier HTML en python

Nous voulons créer un serveur http simple qui sert une page Web HTML statique.
index.html.

```
<html>
  <head>
    <title>Python is awesome!</title>
  </head>
  <body>
    <h1>Afternerd</h1>
    <p>Congratulations! The HTTP Server is working!</p>
  </body>
</html>
```

Pour créer un serveur Web dans **Python 3** , vous devez importer deux modules: `http.server` et `socketserver`.

Dans **Python 2** , il y avait un module nommé [SimpleHTTPServer](#) . Ce module a été fusionné dans `http.server` dans **Python 3**.

```
import http.server
import socketserver

PORT = 8080
Handler = http.server.SimpleHTTPRequestHandler

with socketserver.TCPServer(("", PORT), Handler) as httpd:
    print("serving at port", PORT)
    httpd.serve_forever()
```

1. un serveur Web est un processus qui écoute les demandes entrantes sur une adresse TCP spécifique.(une adresse TCP est identifiée par une *adresse IP* et un *numéro de port*)

2. un serveur Web doit également savoir comment traiter les demandes entrantes.

Ces demandes entrantes sont traitées par des gestionnaires spéciaux. Vous pouvez considérer un serveur Web comme un répartiteur. Une demande arrive, le serveur http inspecte la demande et l'envoie à un gestionnaire désigné.

3. `http.server.SimpleHTTPRequestHandler` est: un simple gestionnaire de requêtes HTTP qui sert les fichiers du **répertoire actuel** et de l' **un de ses sous-répertoires** .

4. La classe `socketserver.TCPServer` .

Une instance de `TCPServer` décrit un serveur qui utilise le protocole TCP pour envoyer et recevoir des messages (http est un protocole de couche d'application au-dessus de TCP).

Pour instancier un serveur TCP, nous avons besoin de deux choses:

1- *L'adresse TCP (adresse IP et un numéro de port)*

2- *Le gestionnaire*

```
socketserver.TCPServer(("", PORT), Handler)
```



```
Handler = http.server.SimpleHTTPRequestHandler
```

l'adresse TCP est transmise sous forme de tuple de (adresse ip, numéro de port)

Le fait de passer une chaîne vide en tant qu'adresse IP signifie que le serveur écoutera sur n'importe quelle interface réseau (toutes les adresses IP disponibles).

Et comme **PORT** stocke la valeur 8080, le serveur écoute les demandes entrantes sur ce port.

Pour le gestionnaire, nous passons au gestionnaire simple dont nous avons parlé plus tôt.

serve_forever est une méthode sur l'instance de `TCPServer` qui démarre le serveur et commence à écouter et à répondre aux demandes entrantes.

sauegardons ce fichier sous le nom **server.py** dans *le même répertoire* que **index.html** car, par défaut, **SimpleHTTPRequestHandler** recherchera un fichier nommé **index.html** dans le répertoire en cours.

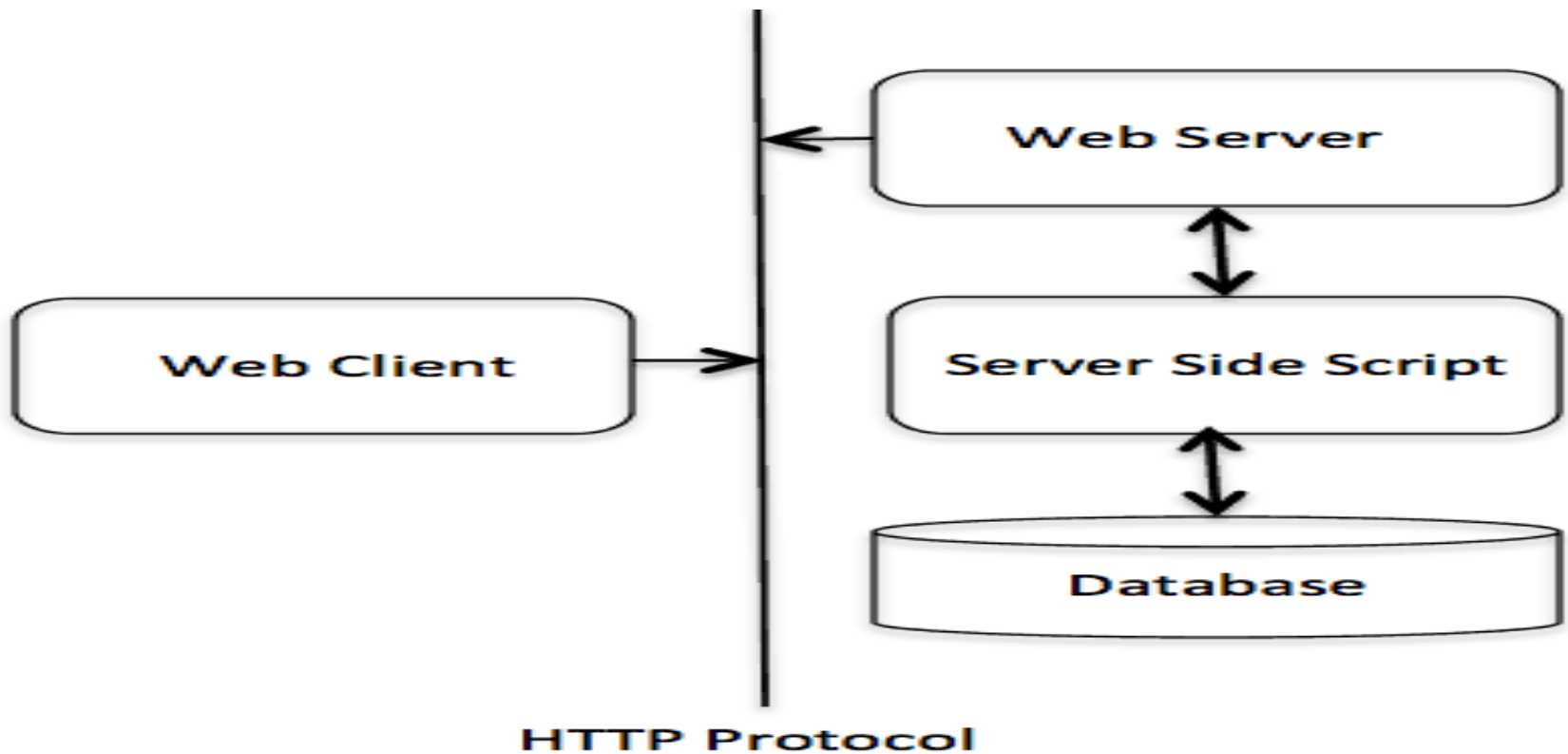
Programmation CGI

L'interface commune de passerelle, ou CGI, est un ensemble de normes définissant la manière dont les informations sont échangées entre le serveur Web et un script personnalisé.

CGI, est une norme permettant aux programmes de passerelle externes de s'interfacer avec des serveurs d'informations tels que les serveurs HTTP.

1. Le navigateur contacte le serveur Web HTTP et demande l'adresse URL, c'est-à-dire le nom du fichier.
2. Web Server analyse l'URL et recherche le nom du fichier. S'il trouve ce fichier, il le renvoie au navigateur, sinon il envoie un message d'erreur indiquant que vous avez demandé un fichier incorrect.
3. Le navigateur Web reçoit la réponse du serveur Web et affiche le fichier reçu ou le message d'erreur.

Cependant, il est possible de configurer le serveur HTTP afin que, lorsqu'un fichier d'un répertoire donné soit demandé, ce fichier ne soit pas renvoyé; au lieu de cela, il est exécuté en tant que programme et tout ce que les sorties du programme sont renvoyées pour que votre navigateur les affiche. Cette fonction est appelée Common Gateway Interface ou CGI et les programmes sont appelés scripts CGI. Ces programmes CGI peuvent être un script Python, un script PERL, un script shell, un programme C ou C ++, etc.



Interaction entre des pages web en python

Création de serveur web http

Python 2

server.py

```
#!/usr/bin/python

import BaseHTTPServer
import CGIHTTPServer

PORT = 8888
server_address = ("", PORT)

server = BaseHTTPServer.HTTPServer
handler = CGIHTTPServer.CGIHTTPRequestHandler
handler.cgi_directories = ["/"]
print "Serveur actif sur le port :", PORT

httpd = server(server_address, handler)
httpd.serve_forever()
```

Python 3

server.py

```
import http.server

PORT = 8888
server_address = ("", PORT)

server = http.server.HTTPServer
handler = http.server.CGIHTTPRequestHandler
handler.cgi_directories = ["/"]
print("Serveur actif sur le port :", PORT)

httpd = server(server_address, handler)
httpd.serve_forever()
```

Interaction entre des pages web en python

Création d'une page http en python

index.py

```
# coding: utf-8

import cgi

form = cgi.FieldStorage()
print("Content-type: text/html; charset=utf-8\n")

print(form.getvalue("name"))

html = """<!DOCTYPE html>
<head>
    <title>Mon programme</title>
</head>
<body>
    <form action="/index.py" method="post">
        <input type="text" name="name" value="Votre nom" />
        <input type="submit" name="send" value="Envoyer information au serveur">
    </form>
</body>
</html>
"""

print(html)
```