

# PQP CREPP Atelier n°5 - Introduction Serveur Web

## 1. Interface graphique

Celle-ci doit être accessible depuis n'importe quel type de terminal

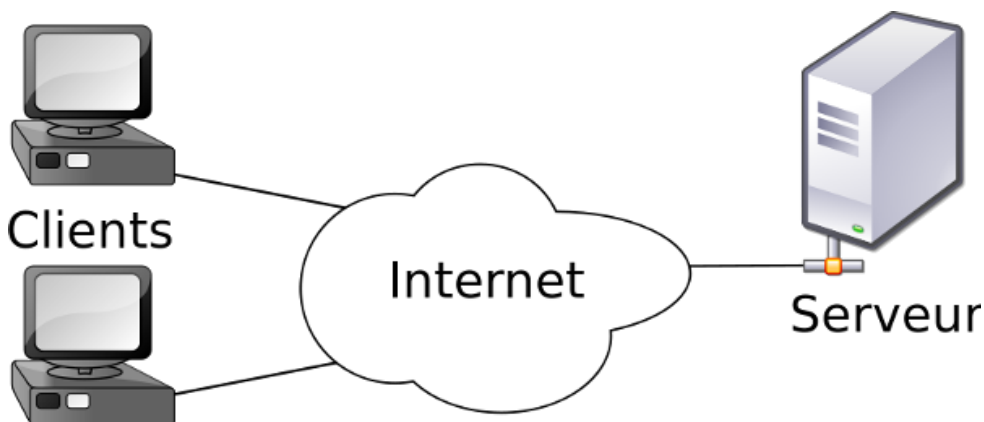
- PC
- Mac
- Linux
- Tablette
- Téléphone

Disposer d'un ensemble de pages

- Configuration
- Affichage des dernières mesures
- Affichage des courbes sur une période donnée

## 2. Client-serveur

L'environnement client-serveur désigne un mode de communication à travers un réseau entre plusieurs programmes : l'un, qualifié de client, envoie des requêtes ; l'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent.



## 3. Internet / HTTP / URL

Suite des protocoles Internet :

La suite TCP/IP est l'ensemble des protocoles utilisés pour le transfert des données sur Internet. En 7 couches.

N° Couche	Exemple de protocole
7 Application	HTTP, FTP, Telnet
6 Présentation	ASCII, Unicode
5 Session	RPC, Netbios, ASP
4 Transport	TCP, UDP

N° Couche	Exemple de protocole
3 Réseau	IPv4 ou IPv6
2 Liaison	Ethernet
1 Physique	Réseaux filaires, Fibre optique

### 3.1 HTTP

L'Hypertext Transfer Protocol (HTTP, littéralement « protocole de transfert hypertexte ») est un protocole de communication client-serveur développé pour le World Wide Web.

On s'en sert tous les jours. Exemple avec le navigateurs Web :

`http://www.crepp.org`

Une requête HTTP présente le format suivant :

```
Ligne de commande (Commande, URL, Version de protocole)
En-tête de requête
[Ligne vide]
Corps de requête
```

Les réponses du Serveur HTTP présentent le format suivant :

```
Ligne de statut (Version, Code-réponse, Texte-réponse)
En-tête de réponse
[Ligne vide]
Corps de réponse
```

Requête :

```
GET /page.html HTTP/1.0
Host: example.com
Referer: http://example.com/
User-Agent: CERN-LineMode/2.15 libwww/2.17b3
```

Réponse :

```
HTTP/1.0 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Server: Apache/0.8.4
Content-Type: text/html
Content-Length: 59
Expires: Sat, 01 Jan 2000 00:59:59 GMT
Last-modified: Fri, 09 Aug 1996 14:21:40 GMT
<TITLE>Exemple</TITLE>
<P>Ceci est une page d'exemple.</P>
```

### 3.2 URL (Uniform Resource Locator)

`http://www.crepp.org:80`

- `http` - protocole de communication, en l'occurrence ici HTTP pour accéder à un serveur web.
- `:` - caractère de séparation obligatoire si le protocole est précisé.
- `//` - chaîne de caractères pour les protocoles dont la requête comprend un chemin d'accès, permettant de préciser et localiser le service avant ce chemin.
- `www.crepp.org` - nom de domaine du service
- `:` - caractère indiquant qu'un numéro de port est précisé en suffixe
- `80` - numéro de port TCP/IP du serveur HTTP

## 4. Le langage HTML

L'HyperText Markup Language, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web.

Exemple du principe des balises ouvrantes et fermantes :

```
<TITLE>Exemple de HTML</TITLE>
```

Exemple d'une page HTML

```
<!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
  <head>
    <title>
      Exemple de HTML
    </title>
  </head>
  <body>
    Ceci est une phrase avec un <a href="cible.html">hyperlien</a>.
    <p>
      Ceci est un paragraphe où il n'y a pas d'hyperlien.
    </p>
  </body>
</html>
```

## 5. Serveur Web avec Micropython sur l'ESP8266

Code Micropython :

```
import usocket as socket

def web_page():
    html = """
    <html><head><title>CREPP PQP</title></head>
    <body>Vive le CREPP</body></html>"""
    return html

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('', 80))
s.listen(5)

while True:
    conn, addr = s.accept()
    print('Got a connection from %s' % str(addr))
    request = conn.recv(1024)
    request = str(request)
    print('Content = %s' % request)
    response = web_page()
    conn.send(response)
    conn.close()
```

URL à saisir dans le navigateur Web pour afficher la page :

http://192.168.4.1:80/

## Document



Crepp 2019 - yvoz.lg@gmail.com

Document source composé en Markdown + Pandoc