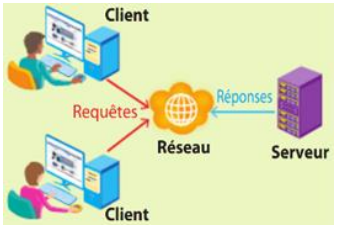
Interaction client serveur

1. Client et serveur

Sur un réseau, les machines échangent des données, suivant un protocole, à l'aide de **requêtes** formulées par des programmes. Les machines ou programmes émettant ces requêtes sont appelés des **clients** et ceux qui y répondent, des **serveurs**.
Le protocole fonctionne par l'intermédiaires de **requêtes** émises par le client et de **réponses** fournies par le serveur.

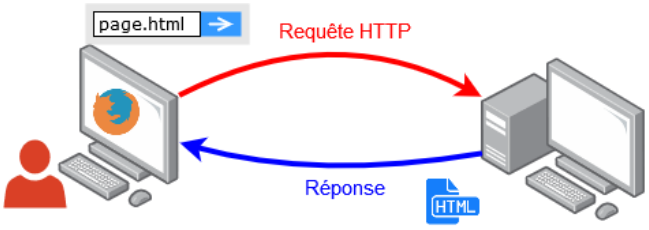


2. Serveur Web ou serveur HTTP

Souvent les serveurs sont spécialisés dans certaines tâches, par exemple, les serveurs qui envoient aux clients des pages au format HTML sont appelés serveur Web.

Cela permet par exemple à un client (navigateur Web) de récupérer des documents hypermédias (HTML) auprès du serveur Web.

- Le **client** : un logiciel de type navigateur Web, installé sur la machine de l'utilisateur.
- Le **serveur** : un logiciel de type serveur Web, qui reste en permanence à l'écoute des demandes qui lui parviennent.



3. Web statique ou dynamique

Dans de nombreux sites Web, le concepteur écrit son code HTML et ce code est simplement envoyé par le serveur Web au client. La page est **statique**, elle est purement consultative.
Dans un page **dynamique**, l'affichage à l'écran de la page dépend des demandes de l'utilisateur (données issues de serveur Web) ou est issue de la programmation au sein de la page (CSS ou programme Javascript).

4. Côté serveur, côte client

Une page dynamique peut être exécuté :

- Côté client : le programme est exécuté sur la machine du client. C'est le cas du CSS et du Javascript.
- Côté serveur : le programme est exécuté sur le serveur. C'est le cas du langage PHP et les formulaires qui sollicitent le serveur. Cela implique qu'une requête est faite au serveur et que l'on attend sa réponse. Il est aussi possible d'utiliser le langage Python.

4.1. Page dynamique côté client

✍ Enregistrer le script ci-dessous dans un fichier **heure.js**.

```
function heure() {
    var start = Date.now();
     function actualiser() {
          var millis = Date.now() - start;
          var date = new Date();
          var str = date.getHours();
          str += ':'+(date.getMinutes()<10?'0':'')+date.getMinutes();
          str += ':'+(date.getSeconds()<10?'0':'')+date.getSeconds();
          document.getElementById('id_heure').innerHTML =  str;
    }
    actualiser();
    setInterval(actualiser,1000);
}
```

✍ Enregistrer le script ci-dessous dans un fichier **heure.html**.

```
<!DOCTYPE html>
<html>
<head>
     <script src="heure.js"></script>
</head>
<body>
     <p>Voici une horloge réalisée en javascript.</p>
     <p id="id_heure">
          <script>
               heure();
          </script>
     </p>
</body>
```

Lors de l'exécution de la page **heure.html**, l'heure s'affiche sans avoir besoin de recharger la page. La page s'exécute côté client.

4.2. Page dynamique côté serveur

✍ Enregistrer le script ci-dessous dans un fichier **heure.py**.

```
import datetime

debut_page = """
    <!DOCTYPE html>
    <html lang ="fr">
        <head>
            <meta charset="utf-8"/>
        </head>

        <body>
            <p> Voici une horloge réalisée avec un serveur Python</p>
            <p> """
maintenant = f"{datetime.datetime.now().strftime('%H:%M:%S')}"
fin_page = """
            </p>
        </body>
    </html> """
page = debut_page + maintenant + fin_page
print(page)
```

✍ Enregistrer le script ci-dessous dans un fichier **serveur.py**.

```
import http.server

PORT = 80 #ou 8888 suivant la configuration du PC
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()
```

✍ Lancer le serveur en exécutant le fichier **serveur.py** et dans un navigateur taper l'URL : http://localhost:80/heure.py (les deux fichiers doivent se situer dans le même répertoire).

L'heure s'affiche mais il faut recharger la page pour actualiser l'heure. L'exécution se fait côte serveur et le client doit faire une nouvelle requête pour réactualiser l'heure.