

Creation d un mini serveur web avec python

Comment démarrer http.server depuis la ligne de commande

Pour démarrer un serveur HTTP en utilisant Python, la bibliothèque standard offre un module très pratique : **http.server**. Ce module permet de lancer un serveur web minimaliste en une seule ligne de commande, ce qui est idéal pour des tests rapides ou le partage temporaire de fichiers sur un réseau local. Voici comment procéder :

Vérification de l'installation de Python

Avant de commencer, assurez-vous que Python est installé sur votre machine. Vous pouvez vérifier cela en ouvrant votre terminal et en exécutant la commande suivante :

```
python --version
```

Si Python est installé, cette commande affichera la version de Python. Si ce n'est pas le cas, vous devez installer Python depuis le site officiel python.org.

Lancer le serveur HTTP

Une fois que vous avez confirmé que Python est installé, naviguez dans le terminal à **l'emplacement du répertoire que vous souhaitez partager** sur le réseau. Utilisez la commande `cd` pour changer de répertoire, comme suit :

```
cd /chemin/du/répertoire
```

Une fois dans le bon répertoire, exécutez la commande suivante pour démarrer le serveur HTTP :

```
python -m http.server
```

Cette commande lance un serveur HTTP sur le port par défaut 8000. Vous verrez alors un message indiquant que le serveur est en cours d'exécution :

```
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
```

Accéder au serveur

Pour accéder au serveur depuis un navigateur web, ouvrez votre navigateur préféré et entrez l'adresse suivante dans la barre d'adresse :

```
http://localhost:8000
```

Cette URL vous permet d'accéder aux fichiers et sous-répertoires du répertoire que vous avez choisi de partager.

Changer le port

Si vous souhaitez utiliser un autre port que 8000, vous pouvez spécifier le port directement dans la commande. Par exemple, pour utiliser le port 8080, tapez :

```
python -m http.server 8080
```

Sécurité et limites



Notez que `http.server` est conçu pour une utilisation en développement ou en réseau local. Il n'est pas sécurisé pour un usage en production, car il ne gère pas les connexions sécurisées (HTTPS) et n'offre pas de protection contre les attaques malveillantes.

En suivant ces instructions, vous pouvez rapidement mettre en place un serveur HTTP simple avec Python pour des tests ou le partage de fichiers.

From:

<https://chanterie37.fr/fablab37110/> - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:

<https://chanterie37.fr/fablab37110/doku.php?id=start:reseaux:python&rev=1766245850>

Last update: **2025/12/20 16:50**

