

## Phase A : La Connexion (socket)

C'est comme passer un coup de téléphone.

- **Code :** `s.connect((HOST, PORT))`
- **Explication :** Le script "compose le numéro" (l'adresse IP de challenge01.root-me.org) et demande le poste 52023. Une fois la connexion établie, le canal est ouvert pour discuter.

## Phase B : L'Écoute (recv)

Une fois connecté, le serveur parle en premier.

- **Code :** `data = s.recv(1024)`
- **Explication :** On tend l'oreille. Le 1024 signifie qu'on prépare un "seau" capable de contenir 1024 octets de données. Le serveur envoie sa phrase (ex: "My string is 'SGVsbG8='...").

## Phase C : L'Extraction (Regex)

Le serveur est bavard. Il envoie du texte inutile ("My string is..."). Nous devons isoler la partie utile.

- **Code :** `re.search(r"(.*)", data)`
- **Explication :** On utilise une expression régulière (Regex). On dit au programme : *"Cherche-moi tout ce qui se trouve entre deux guillemets simples"*. C'est ce qui nous permet de récupérer proprement SGVsbG8=.

## Phase D : Le Décodage

C'est le cœur du challenge. La chaîne est illisible car encodée.

- **Code :** `base64.b64decode(encoded_string)`
- **Explication :** Le format utilisé est le **Base64** (reconnaissable aux caractères alphanumériques et au signe = à la fin). Le script traduit SGVsbG8= en Hello.

## Phase E : L'Envoi (send)

Il faut répondre au serveur avant qu'il ne raccroche.

- **Code :** `s.send(... + "\n")`
- **Explication :** On renvoie la réponse décodée.
  - **Important :** Le `\n` (saut de ligne) est crucial. C'est l'équivalent d'appuyer sur la touche "Entrée" du clavier. Sans cela, le serveur attend indéfiniment que vous finissiez votre phrase.