Сокет — это программный интерфейс для обеспечения информационного обмена между процессами.

Существуют клиентские и серверные сокеты. Серверный сокет прослушивает определенный порт, а клиентский подключается к серверу. После того, как было установлено соединение начинается обмен данными.

Рассмотрим это на простом примере. Представим себе большой коридор с множеством дверей, за которыми стоят люди. Есть и двери, за которыми никого нет. Те самые двери — это порты. Там, где стоит человек — это открытый порт, за которым стоит какое-то приложение, которое его прослушивает. То есть, если, вы подойдете к двери с номером 9090 и постучите в нее, то вам откроют и спросят, чем могут помочь. Так же и с сокетами. Создается приложение, которое прослушивает свой порт. Когда клиент устанавливает соединение с сервером на этом порту именно данное приложение будет ответственно за работу этим клиентом.

После успешной установки соединения сервер и клиент начинают обмениваться информацией. Например, сервер посылает приветствие и предложение ввести какую-либо команду. Клиент в свою очередь вводит команду, сервер ее анализирует, выполняет необходимые операции и отдает клиенту результат.

Давайте создадим 2 файла. Один для сервера, другой для клиента.

Сервер

В Python для работы с сокетами используется модуль socket:

import socket

создадим новый сокет:

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

Данная часть является общей и для клиентских и для серверных сокетов. Дальше мы будем писать код для сервера.

Теперь необходимо определится с адресом и портом для нашего сервера. Оставим строку адреса пустой, чтобы сервер был доступен для всех. А порт возьмем любой от нуля до 65535. В большинстве операционных систем

прослушивание портов с номерами 0 - 1023 требует особых прав доступа. Возьмем номер порта 1090:

```
server_address = ('', 1090)
s.bind(server_address)
```

С помощью метода listen мы запустим для данного сокета режим прослушивания. Метод принимает один аргумент — максимальное количество подключений в очереди.

```
s.listen(1)
```

Принять подключение мы можем с помощью метода accept, который возвращает кортеж с двумя элементами. Новый сокет и адрес клиента.

```
conn, addr = s.accept()
```

После установления соединения мы можем начать прием данных от клиента. Так как мы не знаем точно какой количество данных пошлет клиент, будем получать их небольшими порциями по 1024 байта и постепенно добавлять на уже полученным данным.

```
data=[ ]
while True:
    packet = conn.recv(1024)
    if not packet: break
    data.append(packet)
print(data)
```

Прием данных осуществляется в бесконечном цикле с помощью метода recv. Если данных больше нет, то выходим из цикла.

После получения данных закрываем соединение

```
conn.close()
```

Сервер готов и способен принять от клиента данные. Перейдем к написанию клиента.

Клиент

Для работы клиента необходимо создать сокет, подключиться к серверу, послать ему данные и закрыть соединение.

```
import socket

s = socket.socket()
s.connect(('localhost', 1090))
s.send('hello, world!')
s.close()
```