Сторона сервера (в твоем случае это второй компьютер, ты же не забыла мы экспериментируем в пределах двух компьютеров)

Далее код различается для стороны **сервера** и **клиента**. Рассмотрим сначала сторону сервера.

**s.bind((‘127.0.0.1’, 8888))**

**result = s.recv(1024)**

**print(‘Message:’, result.decode(‘utf-8’))**

**s.close()**

Эти 4 строки кода вообще непонятные. Даже интуитивно понимать сложно. Кроме известного **print().** Ну пройдемся и пытаемся понимать.

**s.bind((‘127.0.0.1’, 8888))** означает, что мы резервируем на сервере (т.е. на нашей же машине) адрес **127.0.0.1** и порт **8888**. На нем мы будем слушать и принимать пакеты информации. Здесь стоят двойные скобки, т.к. методу bind() передается **кортеж** данных – в нашем случае состоящий из строки с адресом и номера порта.

Резервировать можно только свободные порты. Например, если на порте 80 уже работает веб-сервер, то он будет нам мешать.

Далее метод **recv()** объекта **s** прослушивает указанный нами порт (**8888**) и получает данные по одному килобайту (поэтому мы задаем размер буфера **1024** байта). Если на него присылают датаграмму, то метод считывает указанное количество байтов и они попадают в переменную result.

Далее идет всем знакомая функция **print()**, в которой мы выводим сообщение Message: и декодированный текст. Поскольку данные в **result** – это текст в кодировке **UTF-8**, мы должны интерпретировать его, вызвав метод **decode(‘utf-8’).**

Ну и, наконец, вызов метода **close()** необходим, чтобы остановить прослушивание **8888**-го порта и освободить его.

Таким образом, сторона сервера имеет следующий вид:

**import socket**

**s = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)**

**s.bind((‘127.0.0.1’, 8888))**

**result = s.recv(1024)**

**print(‘Message:’, result.decode(‘utf-8’))**

**s.close()**

Сегодня попробую запустить данный код. Как будет на практике даже представлять страшно.

Мысли aiDoni

21/11/2024