

파이썬 소켓 데브로그

파이썬을 통해 소켓 프로그래밍을 공부해보고 있다. 그리고 이를 이용해서 간단한 숫자 게임을 제작해보았다. 게임의 룰은 60 이하의 숫자를 선택해 상대방보다 높은 수를 고르되, 자신과 상대방의 수의 차이가 60과 자신의 수의 차이보다 크면 지는 방식이다. [Highest 60]라 이름 짓고 소켓 통신을 이용해 구현해보았다.

우선 소켓은 서버 소켓과 클라이언트 소켓으로 나뉘어 생성된다. 서버는 하나이되 클라이언트는 여러개일 수 있다. 우선 소켓 모듈을 임포트해주고 각자 스크립트에 소켓 인스턴스를 생성해준다. 소켓은 생성될 때 어떤 네트워크를 통해 통신할 것인지를 결정해야하는데, AF_INET은 IPv4를 쓸지 IPv6을 쓸지 결정하고 두번째 파라미터에서는 어떤 프로토콜을 통해 통신할 것인지를 결정한다.

SOCK_STREAM -> TCP, SOCK_DGRAM -> UDP, SOCK_RAW -> 직접 패킷으로 통신하는 방식이다.

서버소켓은 우선 바인딩을 한다. 바인딩이란 말 그대로 묶어주는 것이다. 어떤 IP주소와 포트로 통신할 것인지를 결정한다.

포트(Port)는, 네트워크를 통해 데이터를 주고받는 프로세스를 식별하기 위해 호스트 내부적으로 프로세스가 할당받는 고유한 값이다. 바로 식별자 역할을 하는 것이다. (물론 포트는 쌍방으로 존재해야한다.) 그리고 리슨을 통해 서버 소켓은 클라이언트 소켓으로부터 오는 통신을 수신을 준비를 한다.

클라이언트 소켓도 대동소이하다. 다만 바인딩과 리슨은 필요 없으며 단지 통신할 IP와 포트에 커넥트만 해주면 된다. 커넥트 요청을 서버 IP와 포트에 보낸 후 서버 소켓은 리슨을 통해 클라이언트 소켓을 받고 엑셉트로 연결을 성립시킨다. 이렇게 서버와 클라이언트에 각각 '항구' 역할을 할 소켓이 생성되었다. 이 소켓들은 정박해있고 이제 옮겨다니며 데이터를 전달할 '배' 역할의 소켓이 필요하다. 이것이 바로 서버 스크립트에 이제 생성될 '커넥션' 소켓이다.

서버소켓을 엑셉트 메서드를 통해 새로운 커넥션 소켓을 생성한다. 자신의 주소를 할당하여 클라이언트 소켓을 향해 출발할 준비를 마친 것이다. 센드와 리시브 함수를 통해 이들은 통신한다. 다만 서버에서 무조건 먼저 출발하고 클라이언트에서 수신한 뒤 클라이언트에서 새로이 보낸 후에야만 다음 메시지를 보낼 수 있다. 그리고 아직까지는 1:1 통신만 가능한 상황이다. 추가 공부를 통해 채팅 프로그램과 같이 대다 통신과 순서 상관 없는 기능까지 구현해봐야할 것이다.

파이썬 스레드 데브로그

연속된 채팅을 위해 '스레딩' 기능을 공부했다. 파이썬에서 스레딩 모듈을 임포트하여 사용할 수 있는데 스레드의 개념에 대해 정리해 보자면, 컴퓨터의 프로세스 내에서 병렬로 업무를 동시에 진행할 수 있도록 하는 것이다. 스레드 함수는 타겟 함수의 이름과 해당 함수가 받는 인자를 argument라는 변수명으로 입력된다. 그렇게 스레드 객체를 생성한 후, 스타드 메서드를 통해 스레딩을 구현한다. 그리고 파이썬

전체 코드에 무한 반복문을 넣어준다. 그 이유는 스레드는 프로세스의 객체이므로 프로세스가 끝나면 스레드 역시 같이 사라진다. 그렇기 때문에 스레드를 유지하기 위해서 프로세스(해당 파이썬 코드)에 무한 반복을 넣어서 계속해서 작동하도록 하는 것이다.

스레드의 단점은 동시에 같은 메모리 영역에 접근할 경우 발생할 수 있는 에러이다. 이런 현상을 방지하기 위해 스레드에서는 락이라는 메서드를 제공하여 동기화 작업을 한다. 또한 파이썬은 인터프리터 언어라 미리 함수로 모듈화를 한 다음 스레드에 넣어주는 것이 좋다.