

K-Digital Training 웹 풀스택 과정

# Socket.io

# Socket.io

- 클라이언트와 서버 간의 짧은 지연 시간, 양방향 및 이벤트 기반 통신을 실시간으로 가능하게 하는 라이브러리
- WebSocket 프로토콜 위에서 구축되었으며 통신 과정을 단순화하고 개선하기 위한 추가 기능을 제공
- 특징
  - 1) 이벤트 기반
  - 2) 자동 재연결



<https://socket.io/>

# 기본 이벤트

- connection
  - 클라이언트가 서버에 연결되었을 때 발생.
  - 클라이언트와의 상호작용을 초기화하거나 초기 데이터를 전달할 수 있음
- disconnect: 클라이언트가 연결을 해제했을 때 발생
- disconnecting: 클라이언트가 연결을 해제하려는 경우에 발생
- error: 연결 중에 오류가 발생했을 때 발생

# 사용자 지정 이벤트

`socket.emit(eventName[, ...args][, ack])`

```
socket.emit("hello", "world", (response) => {
  console.log(response); // "got it"
});
```

이벤트를 보내는 쪽

```
socket.on("hello", (arg, callback) => {
  console.log(arg); // "world"
  callback("got it");
});
```

이벤트를 받는 쪽

# 룸 관련 함수

```
//클라이언트를 특정 방으로 추가
socket.join("roomA");
//클라이언트를 특정 방에서 제거
socket.leave("roomA");
//특정 방에 속한 모든 클라이언트에게 특정 이벤트와 데이터를 발송
io.to("roomA").emit("message", "Hello, Room A!");
//특정 방에 속한 클라이언트들에게만 특정 이벤트와 데이터를 발송.
//이때 이벤트는 송신자 클라이언트를 제외한 다른 클라이언트들에게 전달
socket.to("roomA").emit("message", "Hello, Room A!");
socket.broadcast.to("roomA").emit("message", "A new user has joined the room.");
//송신자를 제외한 서버에 연결된 모든 클라이언트에게 이벤트와 데이터를 발송
socket.broadcast.emit("message", "A new user has joined the room.");
//송신자를 포함하여 모든 클라이언트에게 이벤트와 데이터를 발송
io.socket.emit("message", "Hello, Room A!");
//특정 방에 속한 클라이언트들의 정보를 확인
const roomInfo = io.sockets.adapter.rooms.get(room);
console.log(roomInfo); // 방에 속한 클라이언트 id정보가 출력됩니다.
```

# Socket.io 실습

# Socket.io 불러오기 (server)

```
npm install socket.io
```

```
const http = require("http");
const express = require("express");
const SocketIO = require("socket.io");
const app = express();
const PORT = 8000;
const server = http.createServer(app);
const io = SocketIO(server);

// ...

server.listen(PORT, () => {
  console.log(`http://localhost:${PORT}`);
});
```



# Socket.io 불러오기 (client)

```
localhost:8000/socket.io/socket.io.js
```

서버가 연결되면 브라우저를 통해 소켓io 파일에 접근이 가능

```
<script src="/socket.io/socket.io.js"></script>
```

```
const socket = io();
```

io() 함수는 클라이언트 소켓을 생성하고, 서버와의 연결을 설정

# 실습1 Socket 연습

- 각 버튼을 누를 때마다 서버로 메시지 전송
- 서버
  - 클라이언트로부터 받은 메시지를 console 에 찍고, 그에 대한 응답을 보내기
- 클라이언트
  - 버튼을 누를 때 서버로 메시지 보내기
  - 서버로부터 받은 응답 메시지를 화면에 보여주기

```
PS E:\Development\github\SeSAC\SeSAC4_web\
Listening on *:3000
Server Socket Connected
PS E:\Development\github\SeSAC\SeSAC4_web\
PS E:\Development\github\SeSAC\SeSAC4_web\
Listening on *:3000
Server Socket Connected
█
```

Hello Wolrd!

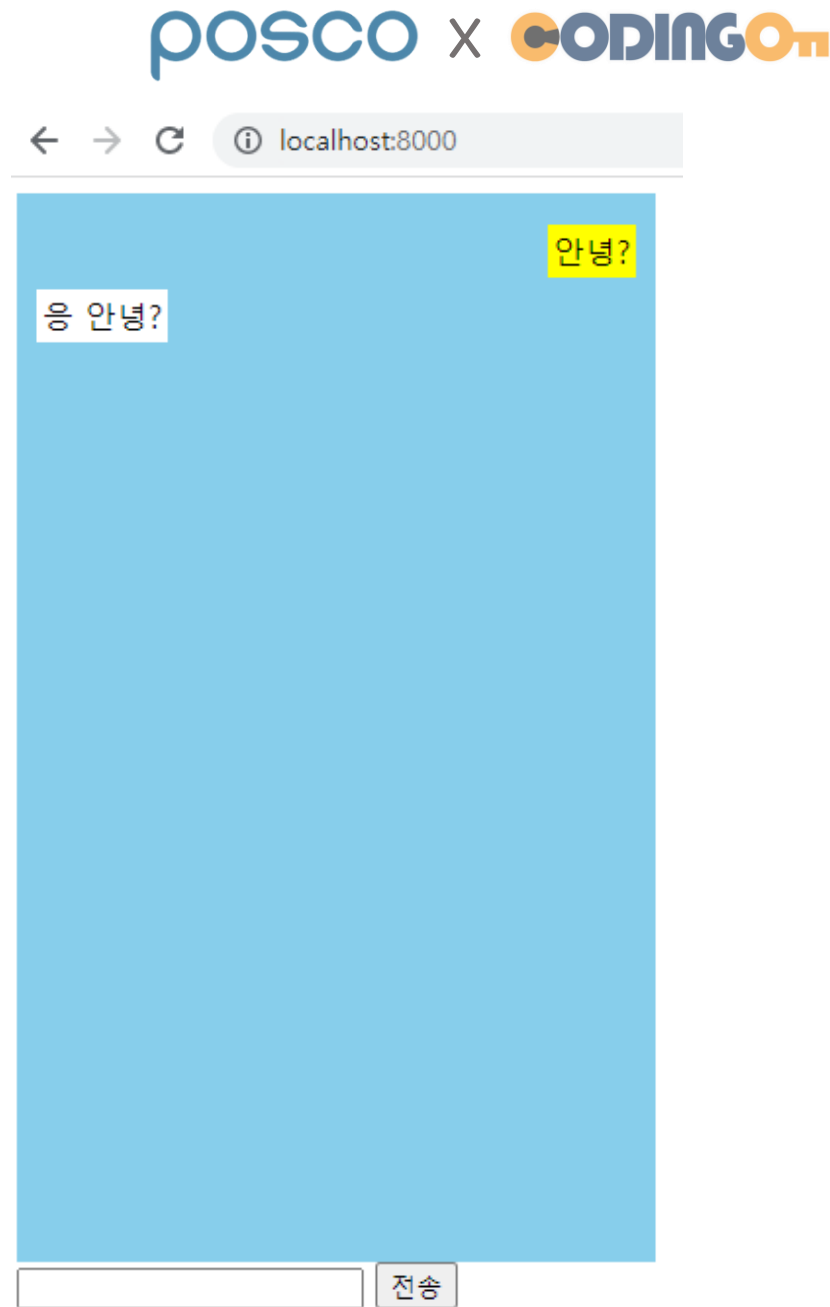
hello study bye

```
// 이벤트 명을 지정하고 메세지를 보냄
socket.emit('전송할 이벤트 명', msg);

// 해당 이벤트를 받고 콜백함수를 실행
socket.on('받을 이벤트 명', (msg) => { ... });
```

# 실습2 채팅창 UI 만들기

- 내가 보낸 채팅은 오른쪽에
- 다른 사람이 보낸 채팅은 왼쪽에 표시될 수 있도록



# 실습3 채팅창 입장 안내 문구

- Socket에 담겨 있는 id 정보를 이용하여 000님이 입장했습니다. 라는 메시지 보여주기

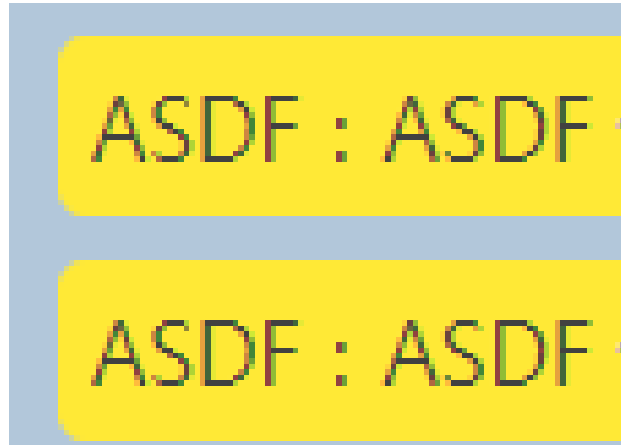
```
io.on( "connection", function ( socket ){
    // 최초 입장했을 때
    console.log( "Server Socket Connected", socket.id );
    io.emit( "notice", `${socket.id}님이 입장하셨습니다.` );
});
```

zEzrHgFXe-XsytExAAAR님이 입장하셨습니다.

전송

# 실습4 채팅창 메시지 전송

- 메시지에 누가 작성한 메시지인지 작성자 이름 받고 이름 보여주기
- 아래와 같은 형식으로 보여줘도 OK
  - 작성자 : MESSAGE

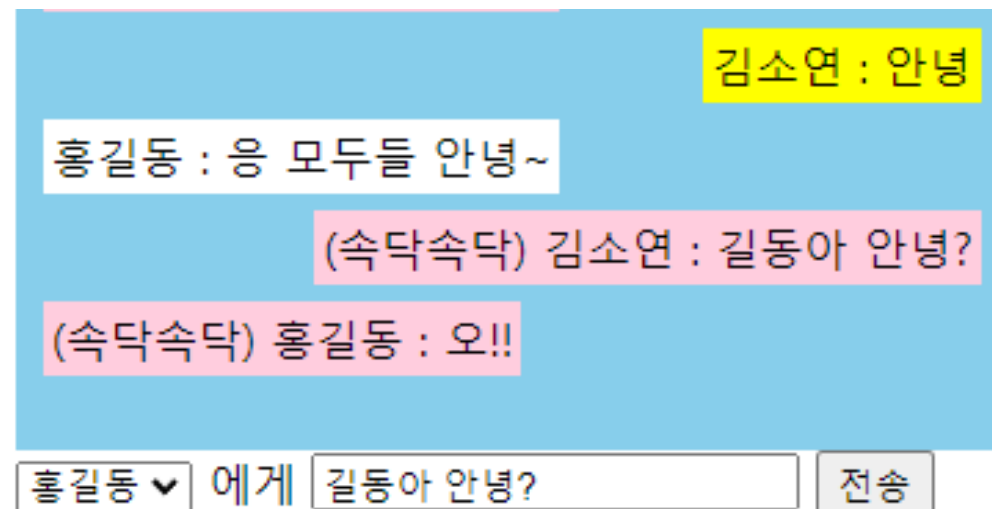


# 실습5 DM기능 추가하기

- 특정 사람에게만 메시지를 보낼 수 있도록 DM 기능 추가해보기
- DM 메시지는 일반 메시지와 다르게 ui 상으로 변화 주기



A UI mockup showing a light blue rectangular area. Below it, there is a dropdown menu with '전체' (All) selected, followed by the text '에게' (to) and an empty input field. To the right of the input field is a button labeled '전송' (Send). Below the dropdown menu, a list of names is shown: '전체', '김소연', and '홍길동'. The '전체' option is highlighted with a dark background.



A UI mockup of a chat conversation on a light blue background. The messages are as follows:  
- Kim So-yeon (yellow bubble): 안녕하세요 (Hello)  
- Hong Gil-dong (white bubble): 응 모두들 안녕~ (Yes, everyone is hello~)  
- Kim So-yeon (pink bubble): (속닥속닥) 길동아 안녕? (Whisper) Gil-dong, hello?  
- Hong Gil-dong (pink bubble): (속닥속닥) 오!! (Whisper) Oh!!  
At the bottom, there is a dropdown menu with '홍길동' (Hong Gil-dong) selected, followed by the text '에게' (to) and an input field containing '길동아 안녕?' (Gil-dong, hello?). To the right of the input field is a button labeled '전송' (Send).