공유데이

정승욱

필수 과제

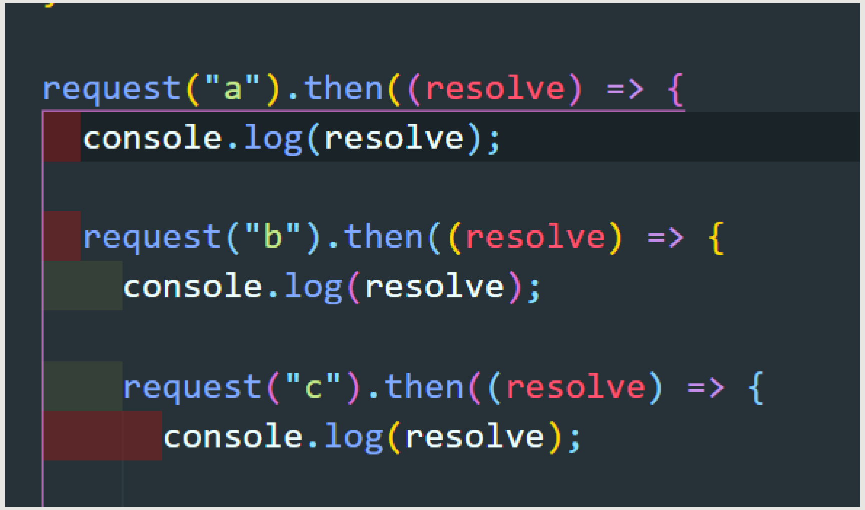
1 - 1학기와 2학기의 차이

2 - Database 설계 및 SQL 작성

➡ 5 - 비동기 프로그래밍 이해

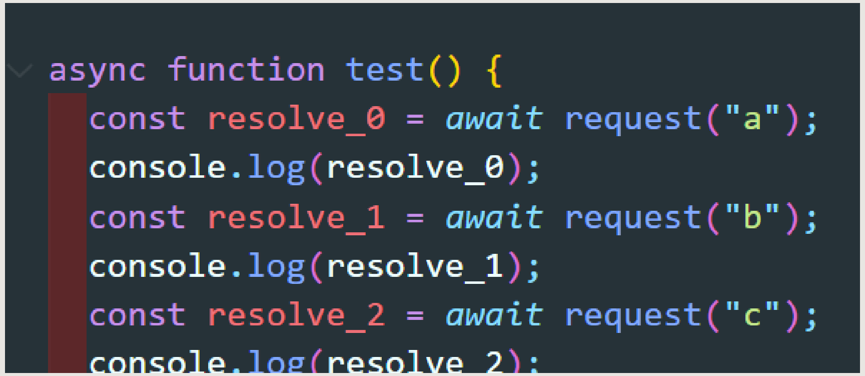
➡ 6 - node.js와 socket.io를 활용한 채팅 server/client 구현

비동기 프로그래밍 이해



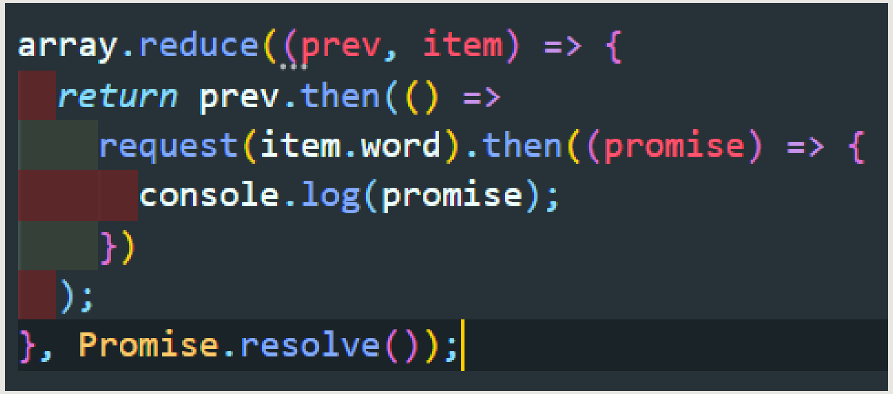
```
request("a").then((resolve) => {
  console.log(resolve);

  request("b").then((resolve) => {
    console.log(resolve);

    request("c").then((resolve) => {
      console.log(resolve);
```

Promise_hard_code



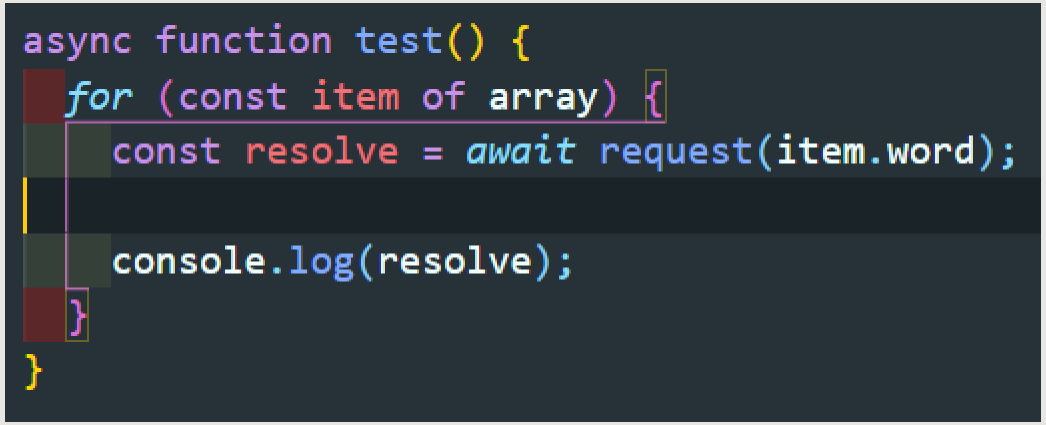
```
async function test() {
  const resolve_0 = await request("a");
  console.log(resolve_0);
  const resolve_1 = await request("b");
  console.log(resolve_1);
  const resolve_2 = await request("c");
  console.log(resolve_2);
```

Await_hard_code



```
array.reduce((prev, item) => {
  return prev.then(() =>
    request(item.word).then((promise) => {
      console.log(promise);
    })
  );
}, Promise.resolve());
```

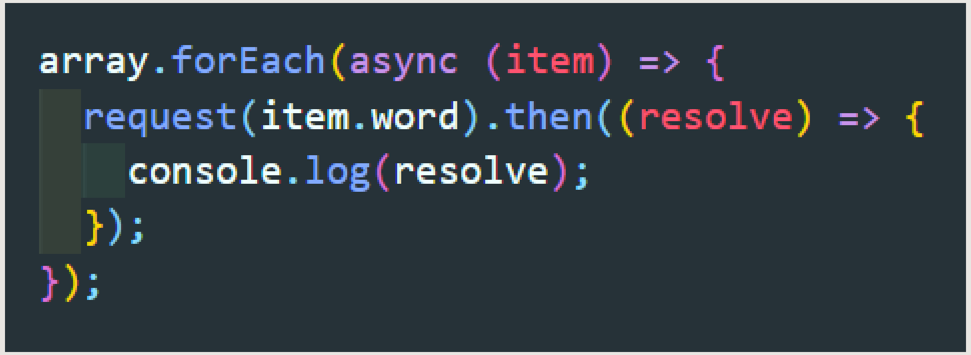
Promise_soft_code



```
async function test() {
  for (const item of array) {
    const resolve = await request(item.word);

    console.log(resolve);
  }
}
```

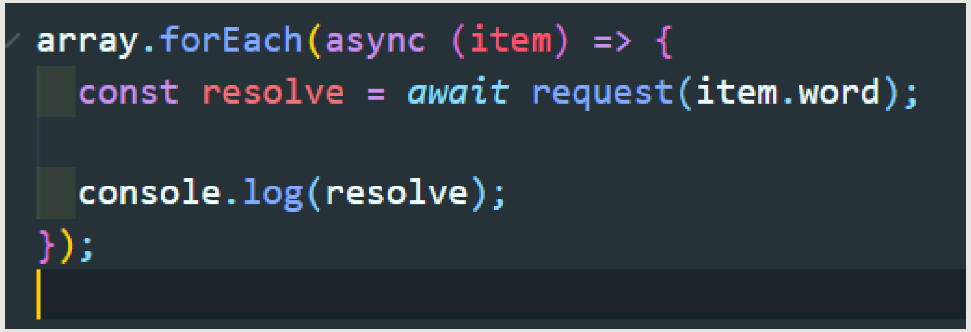
Await_soft_code

비동기 프로그래밍 이해



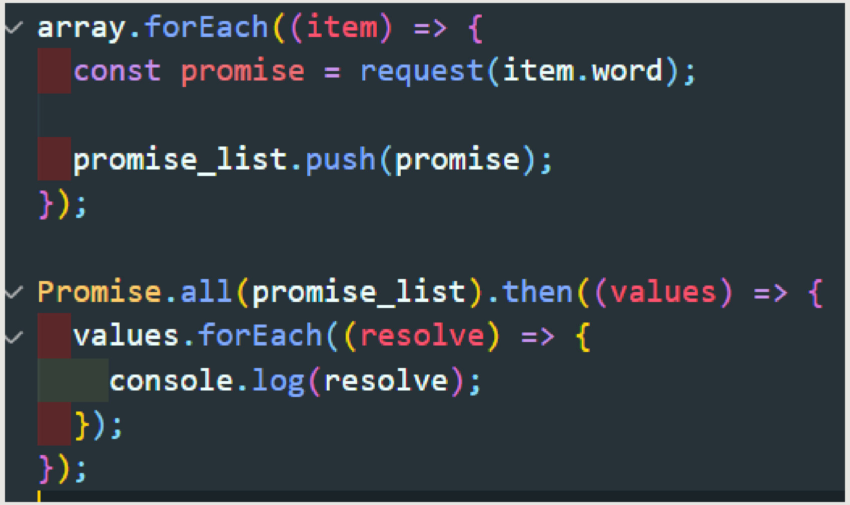
```
array.forEach(async (item) => {
  request(item.word).then((resolve) => {
    console.log(resolve);
  });
});
```

Promise_all_non_sequence



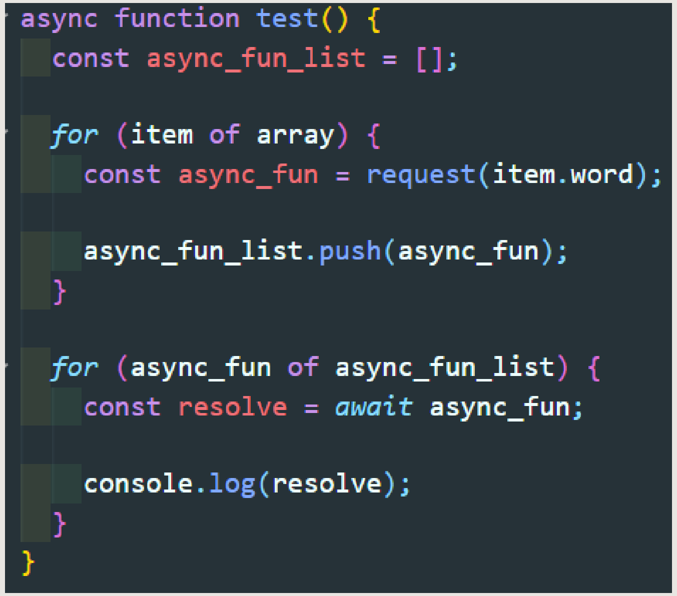
```
array.forEach(async (item) => {
  const resolve = await request(item.word);

  console.log(resolve);
});
```

Await_all_non_sequence



```
array.forEach((item) => {
  const promise = request(item.word);

  promise_list.push(promise);
});

Promise.all(promise_list).then((values) => {
  values.forEach((resolve) => {
    console.log(resolve);
  });
});
```

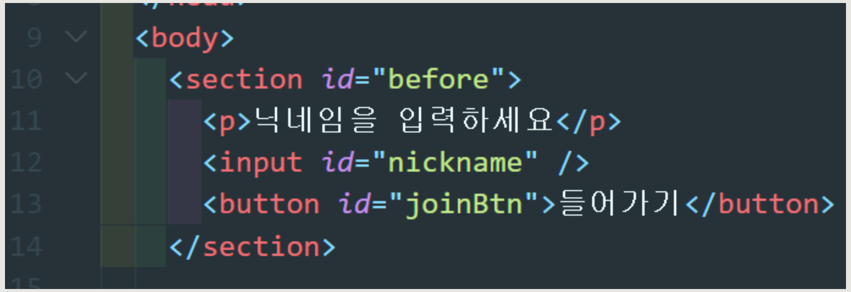
Promise_all_sequence



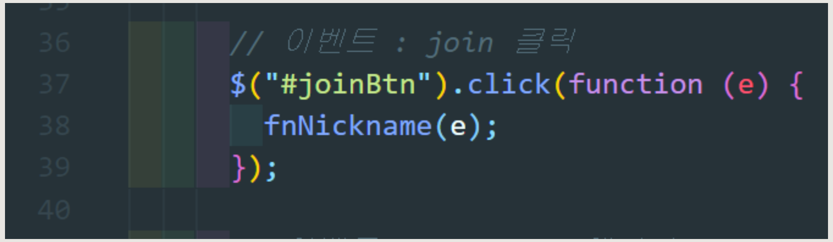
```
async function test() {
  const async_fun_list = [];

  for (item of array) {
    const async_fun = request(item.word);

    async_fun_list.push(async_fun);
  }

  for (async_fun of async_fun_list) {
    const resolve = await async_fun;

    console.log(resolve);
  }
}
```

Await_all_sequence

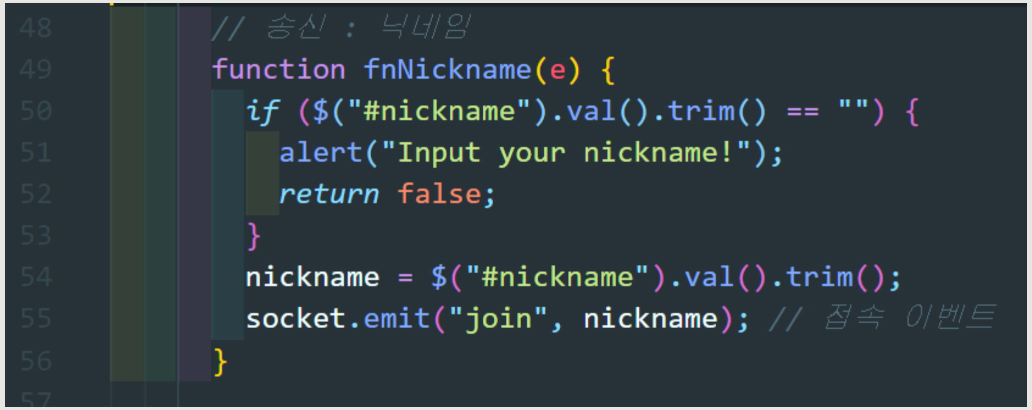
node.js와 socket.io를 활용한 채팅 server/client 구현



```
<body>
  <section id="before">
    <p>닉네임을 입력하세요</p>
    <input id="nickname" />
    <button id="joinBtn">들어가기</button>
  </section>
```

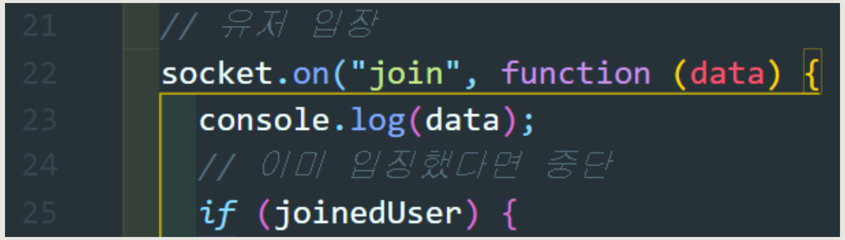


```
// 이벤트 : join 클릭
$("#joinBtn").click(function (e) {
  fnNickname(e);
});
```

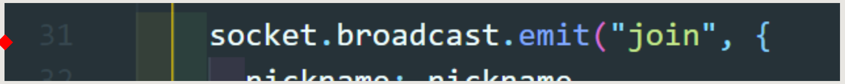


```
// 송신 : 닉네임
function fnNickname(e) {
  if ($("#nickname").val().trim() == "") {
    alert("Input your nickname!");
    return false;
  }
  nickname = $("#nickname").val().trim();
  socket.emit("join", nickname); // 접속 이벤트
}
```

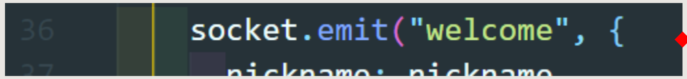
Emit!!



```
// 유저 입장
socket.on("join", function (data) {
  console.log(data);
  // 이미 입장했다면 중단
  if (joinedUser) {
```

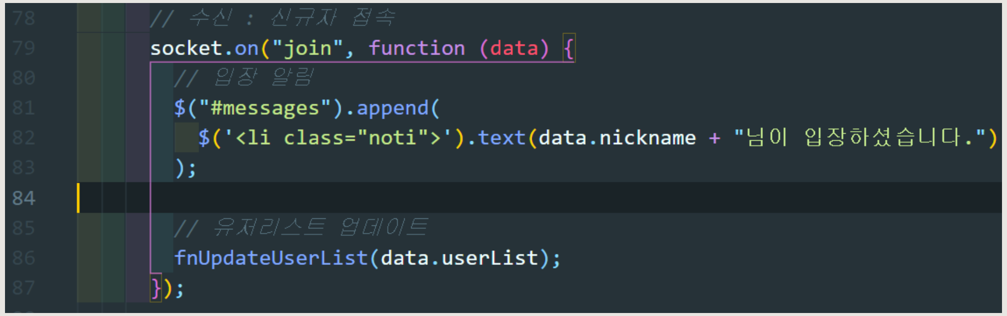


```
socket.broadcast.emit("join", {
  nickname: nickname
```

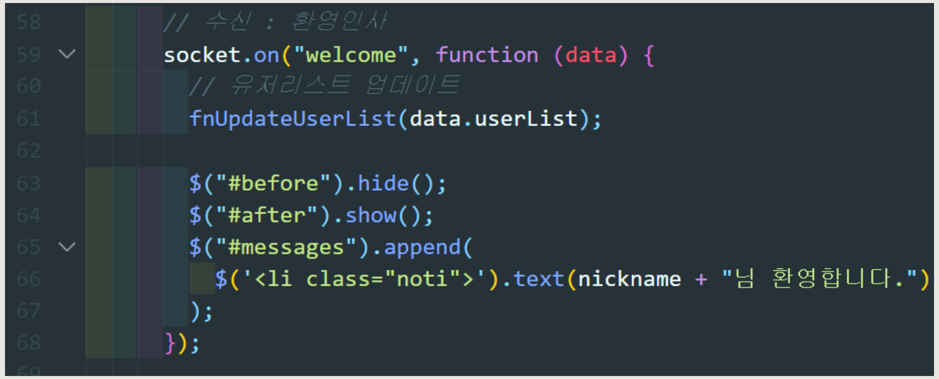


```
socket.emit("welcome", {
  nickname: nickname
```

Emit!! *2



```
// 수신 : 신규자 접속
socket.on("join", function (data) {
  // 입장 알림
  $("#messages").append(
    $('<li class="noti">').text(data.nickname + "님이 입장하셨습니다.")
  );

  // 유저리스트 업데이트
  fnUpdateUserList(data.userList);
});
```



```
// 수신 : 환영인사
socket.on("welcome", function (data) {
  // 유저리스트 업데이트
  fnUpdateUserList(data.userList);

  $("#before").hide();
  $("#after").show();
  $("#messages").append(
    $('<li class="noti">').text(nickname + "님 환영합니다.")
  );
});
```

감사합니다