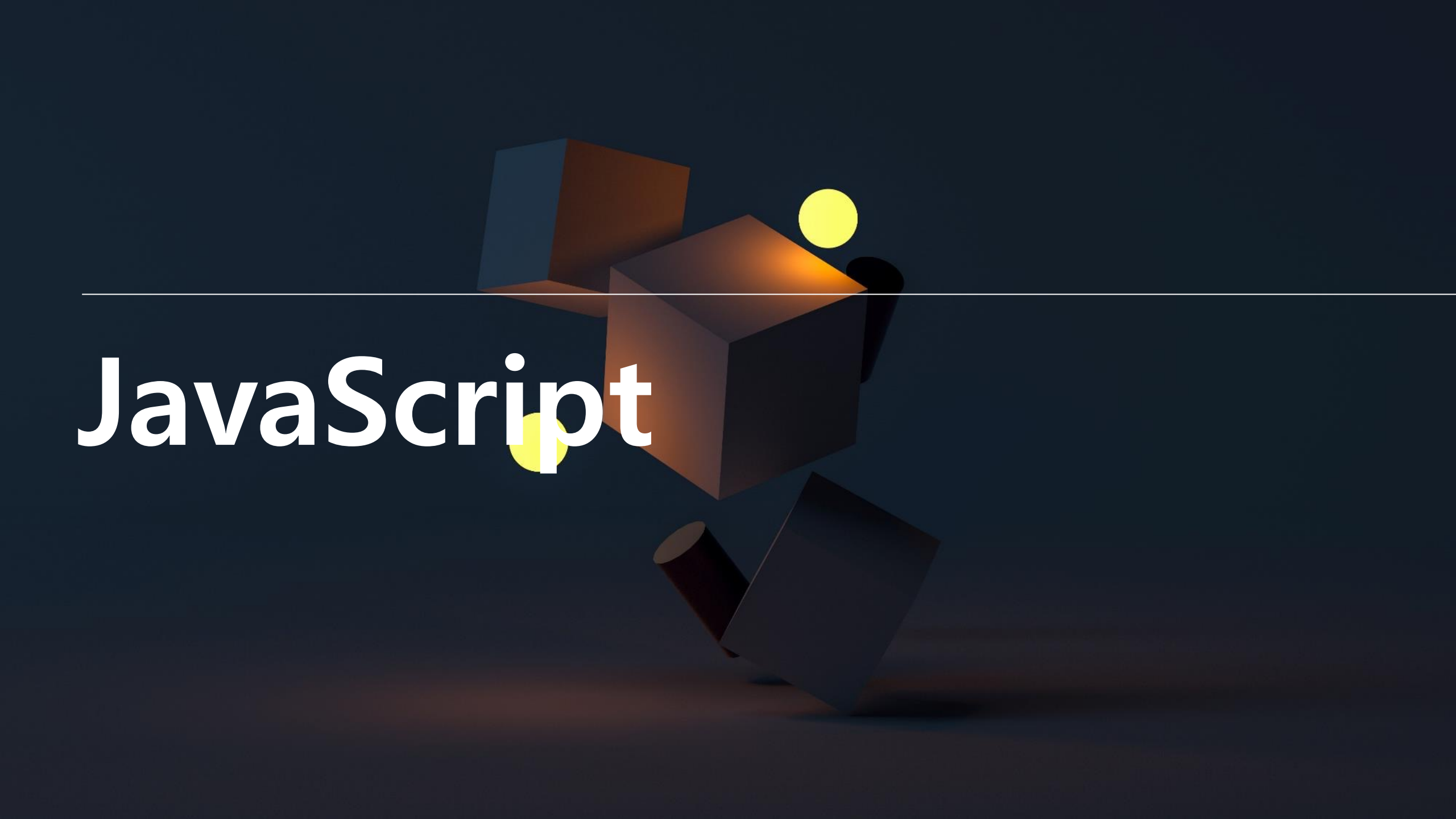
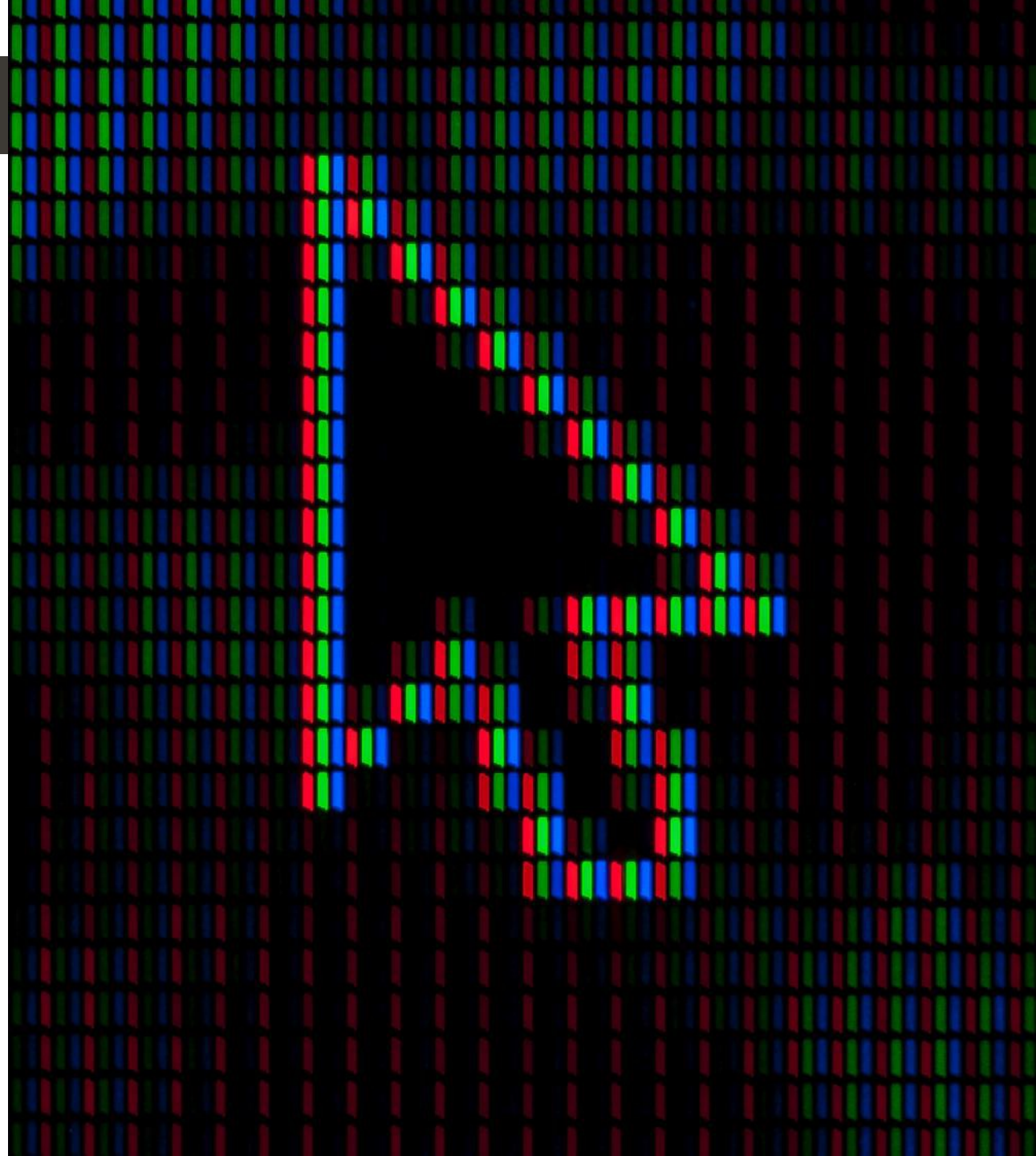
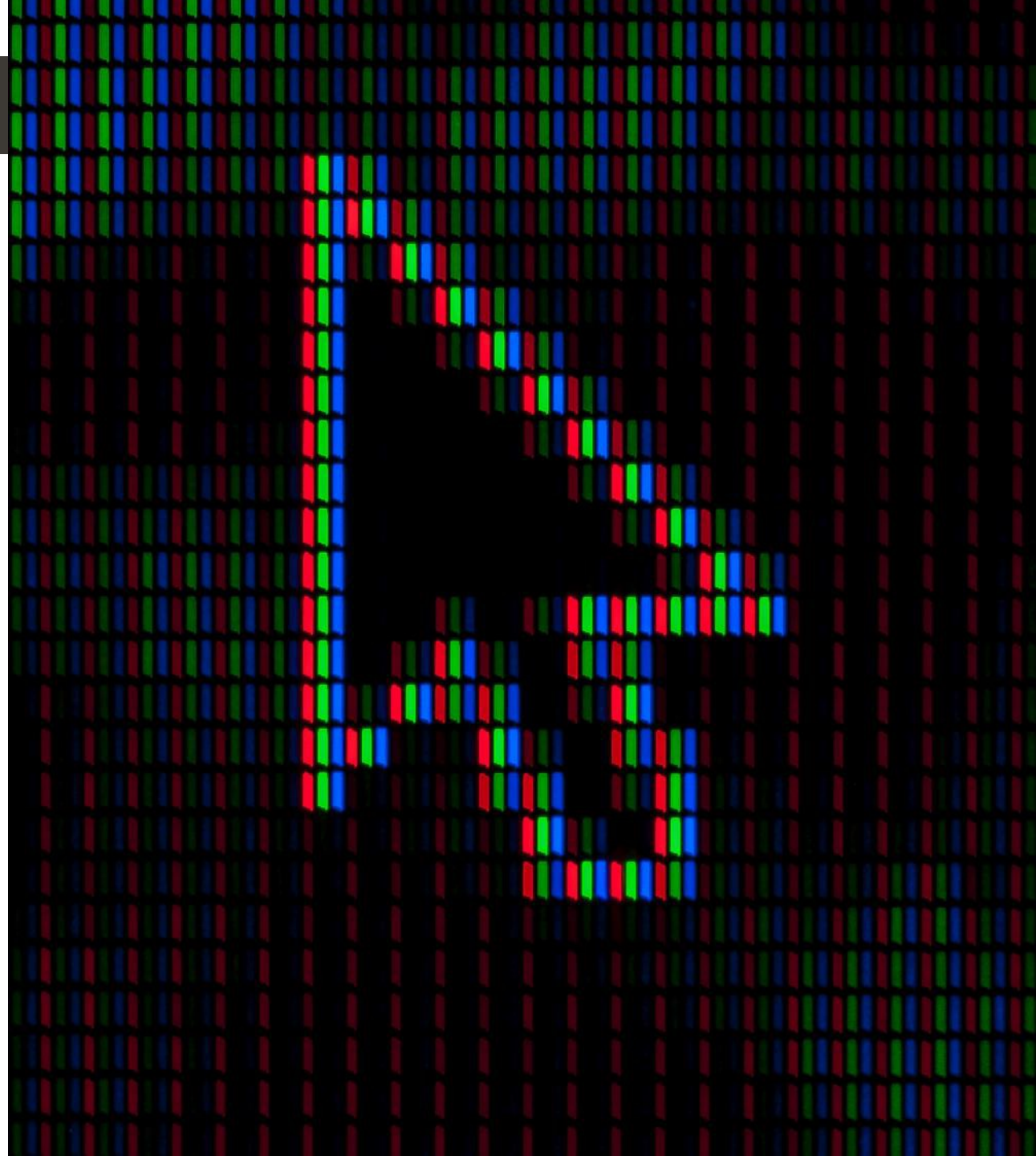
JavaScript



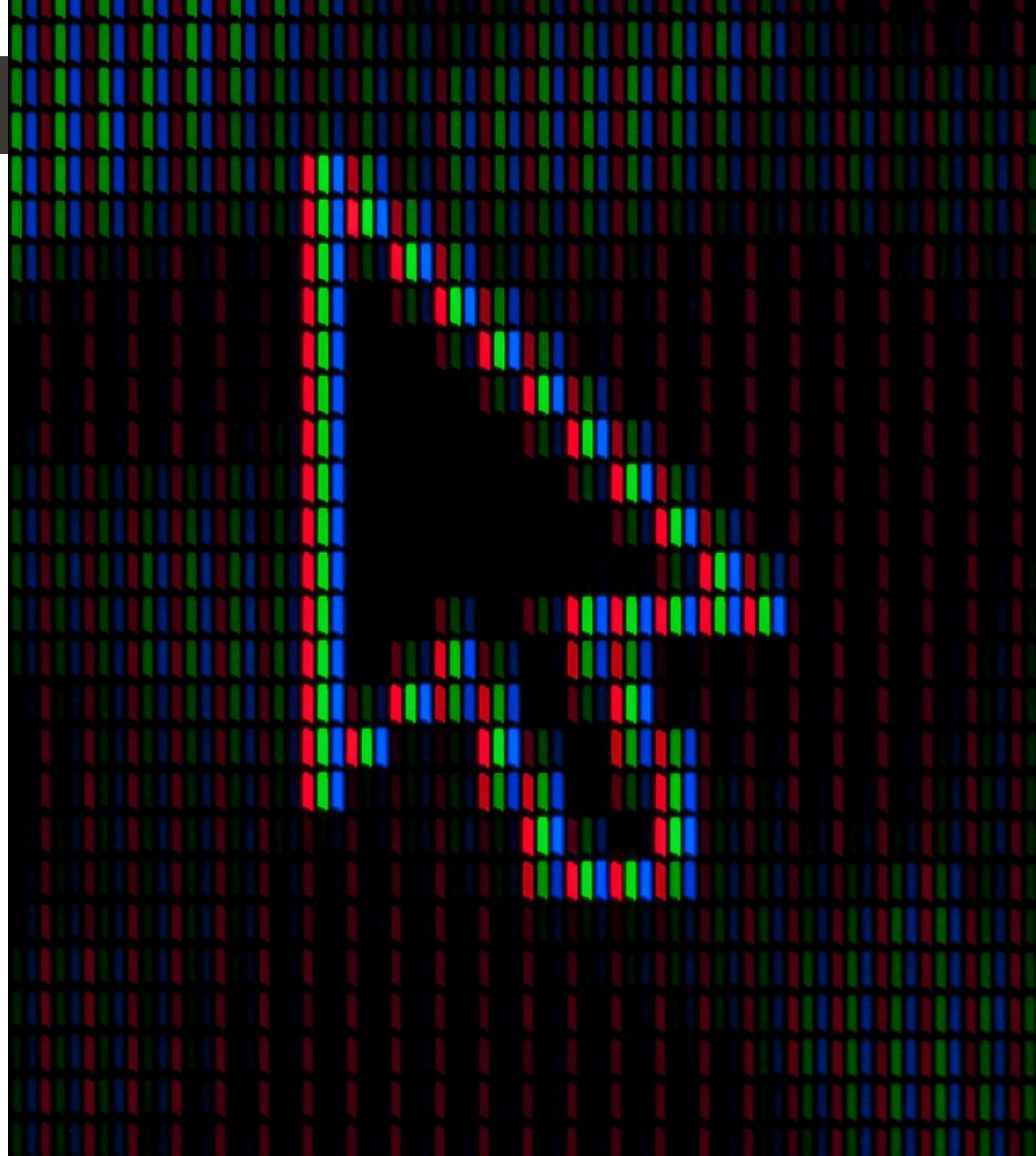
목차 a table of contents



목차 a table of contents



목차 a table of contents



JavaScript 활용

Part 1

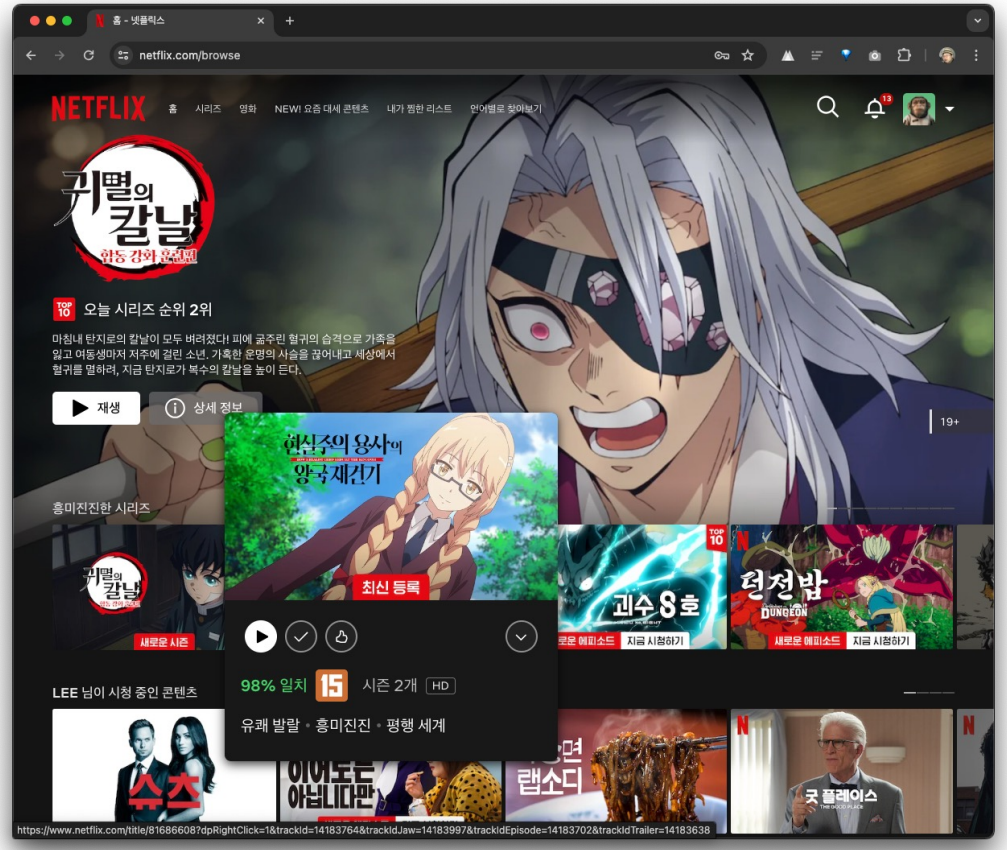
자바스크립트

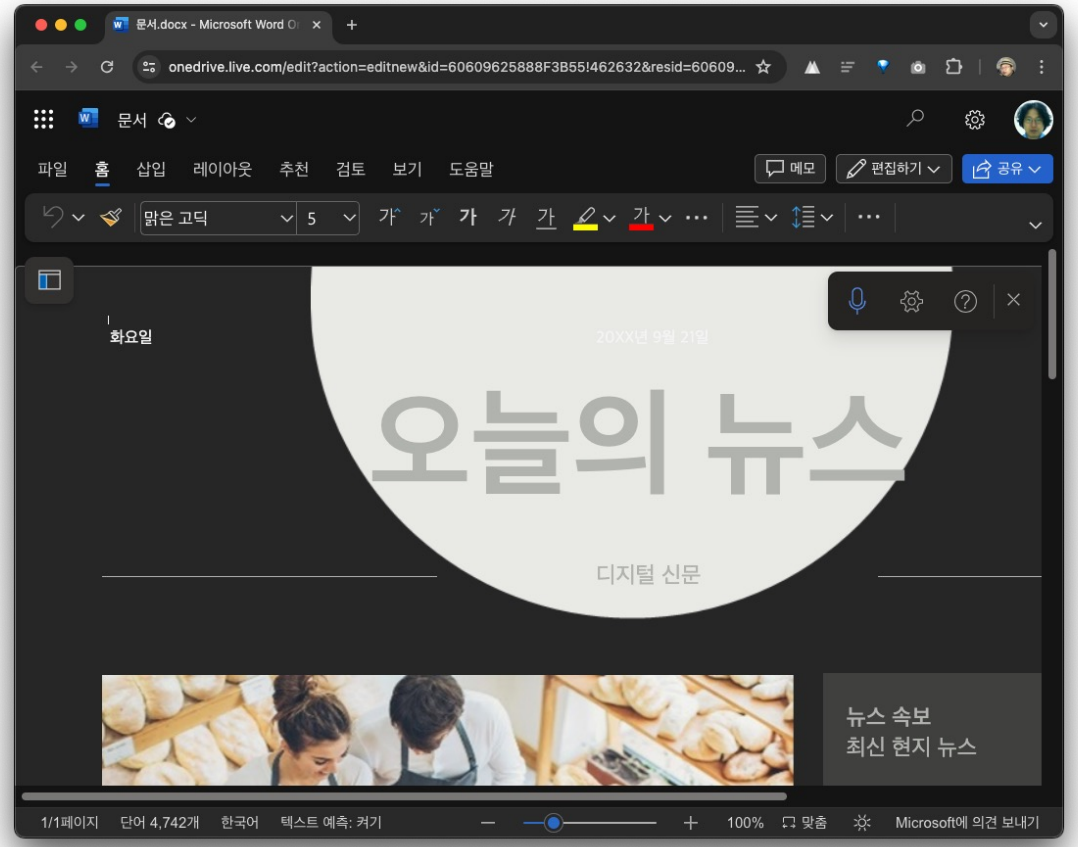
Part 1

웹 클라이언트 애플리케이션 개발

Part 1

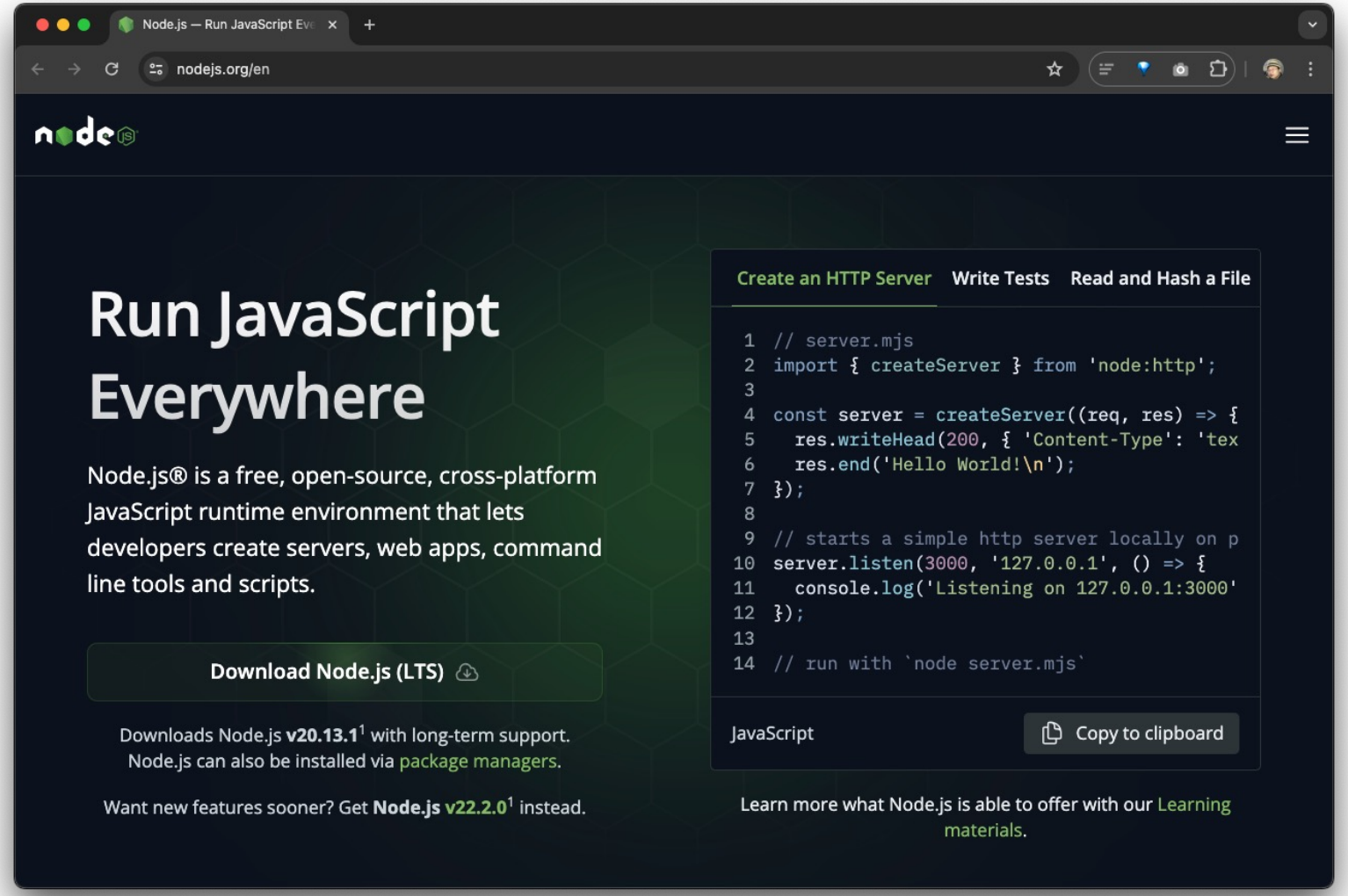
웹 클라이언트 애플리케이션 개발





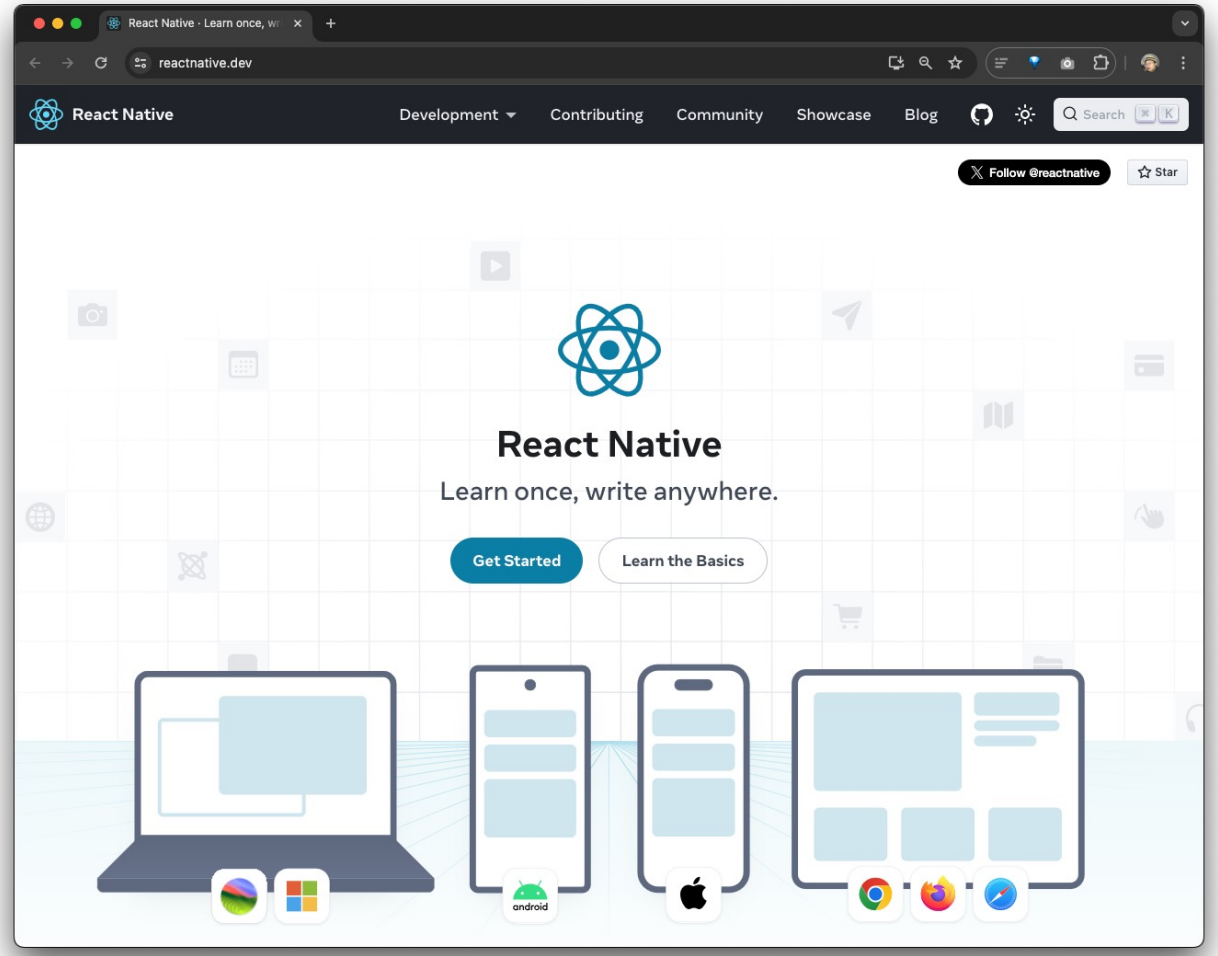
Part 1

웹 서버 애플리케이션 개발



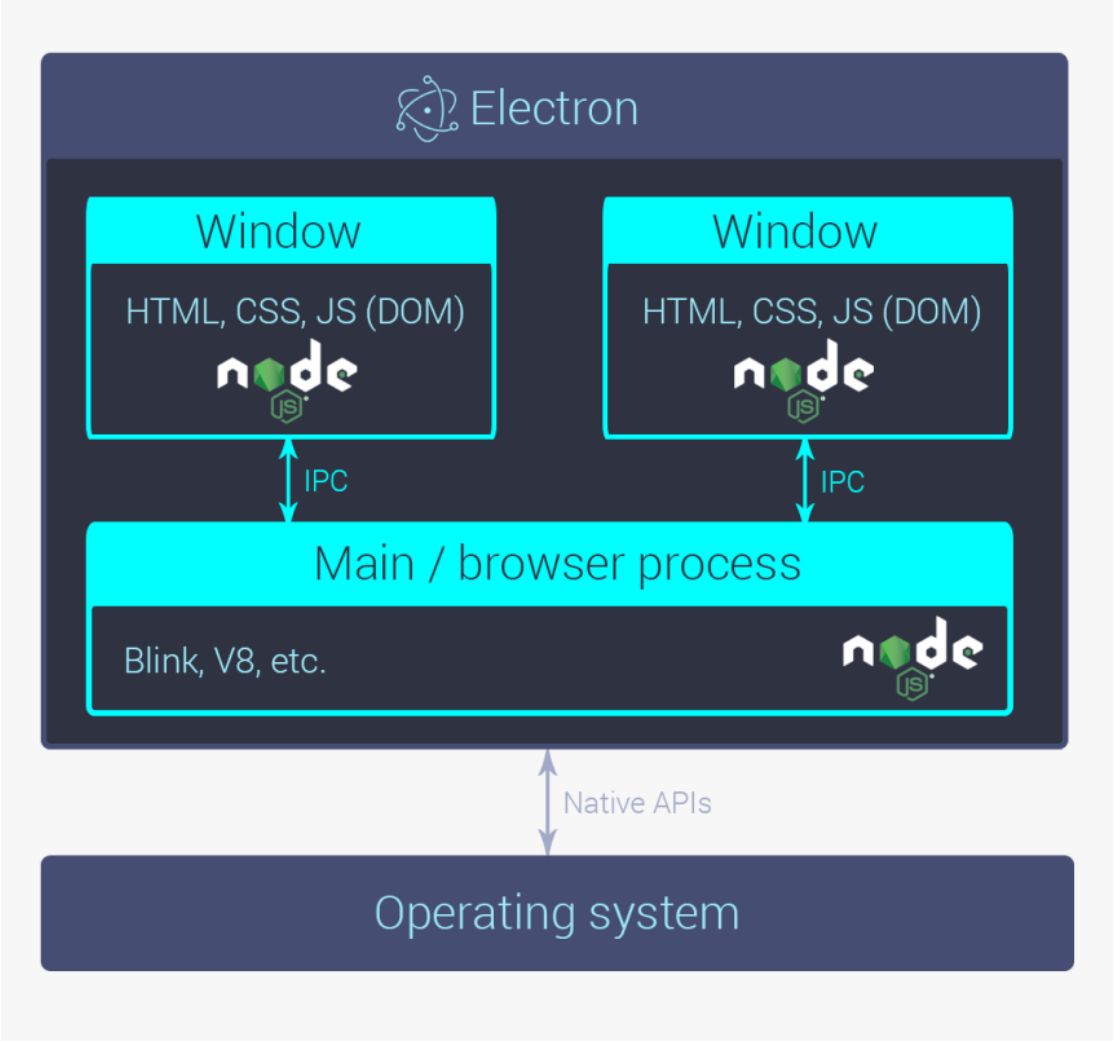
Part 1

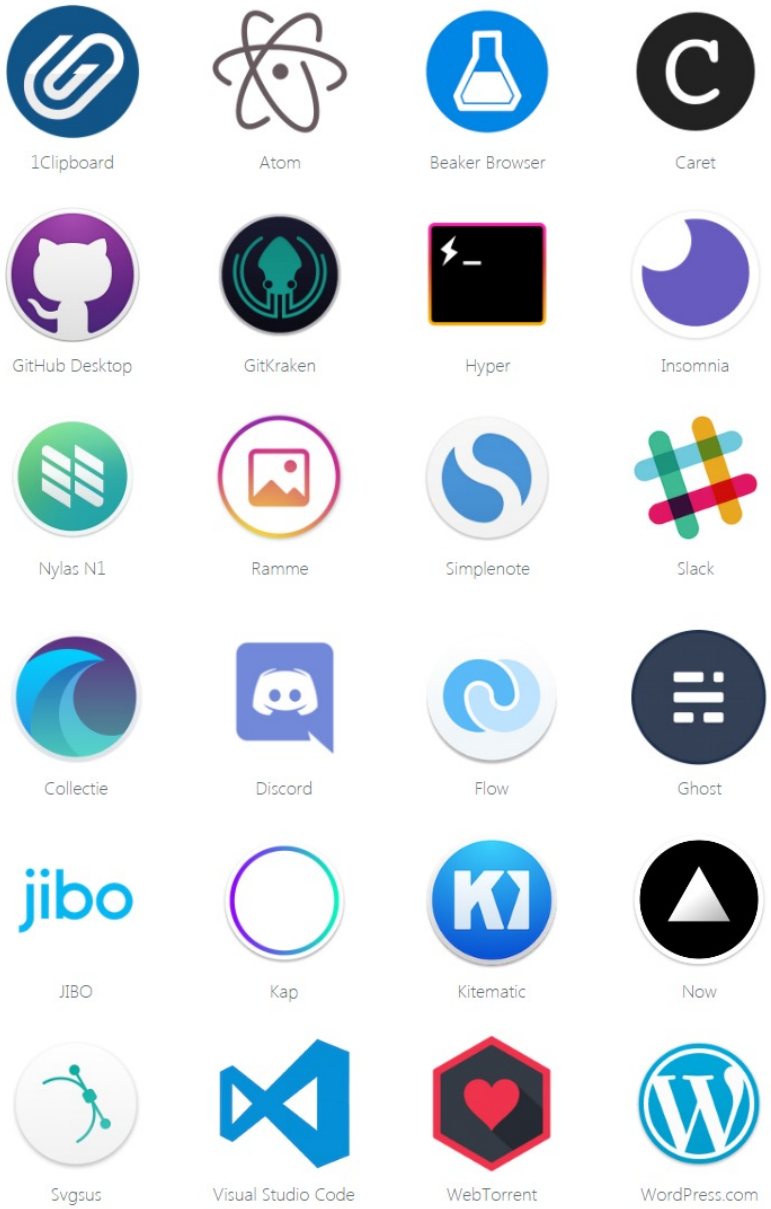
모바일 애플리케이션 개발



Part 1

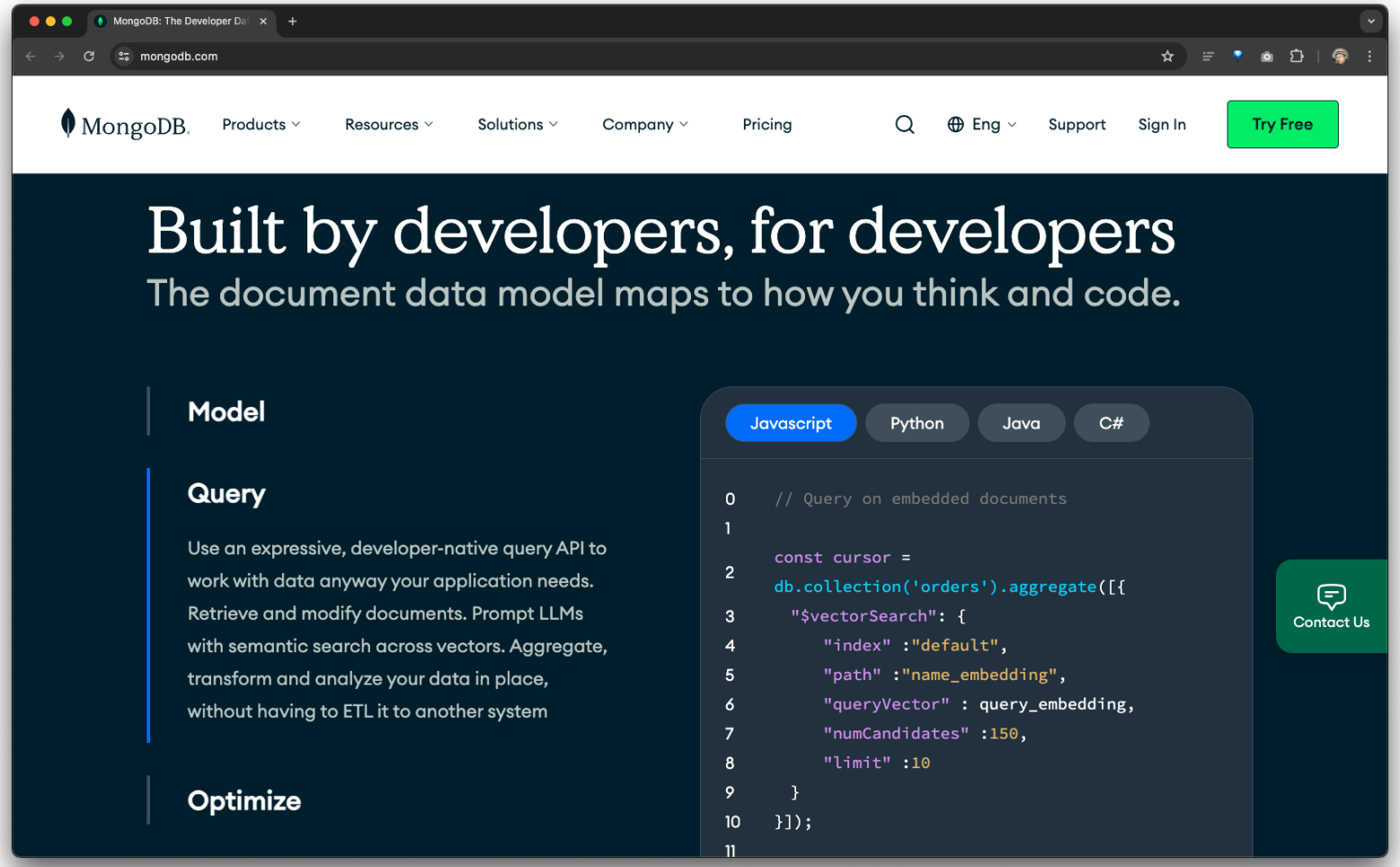
데스크톱 애플리케이션 개발





Part 1

데이터베이스 관리



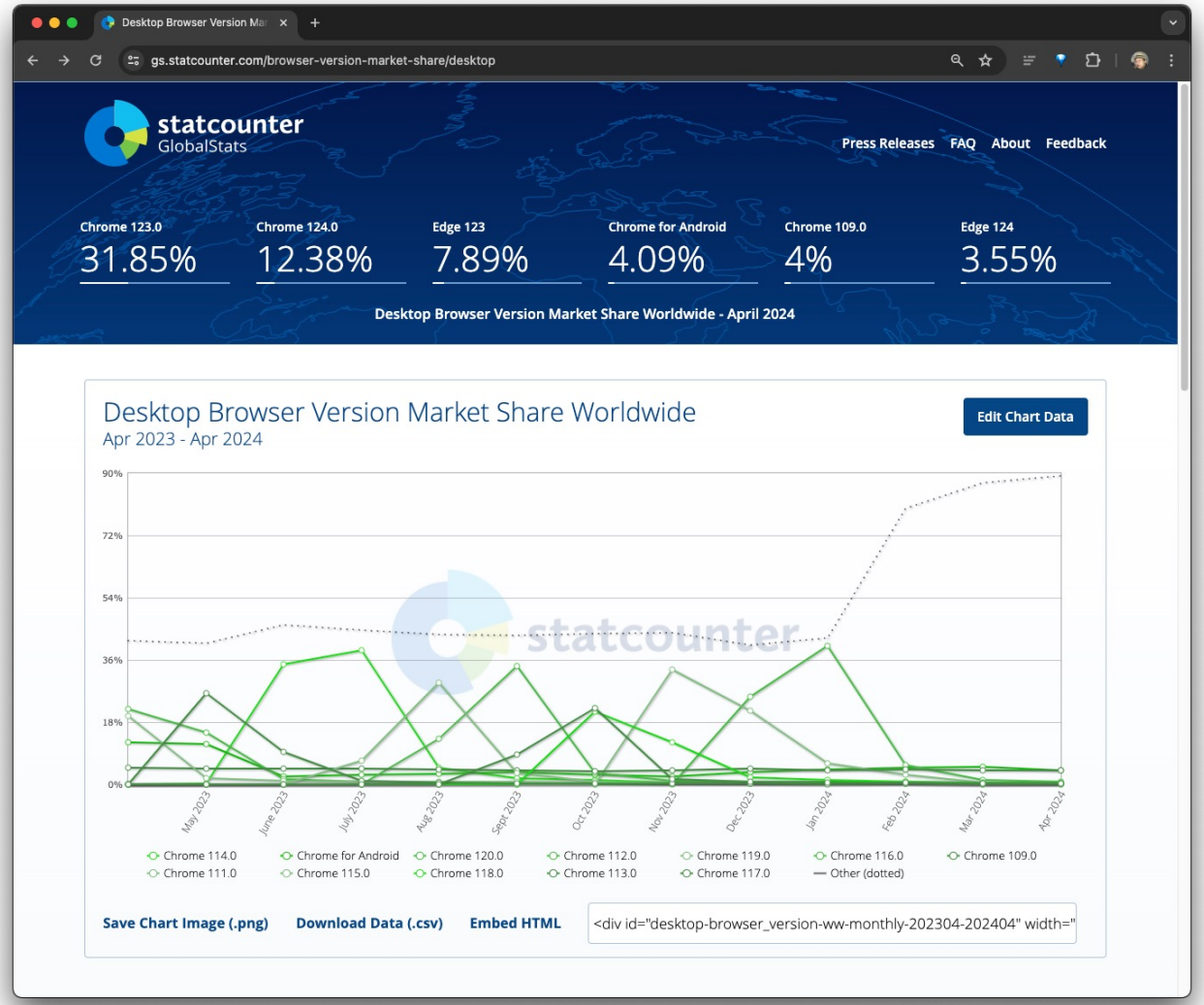
Part 1

자바스크립트의 종류

ECMAScript 버전	표준 발표 시기
ECMAScript 1	1997년 6월
ECMAScript 2	1998년 6월
ECMAScript 3	1999년 12월
ECMAScript 4	2008년 10월
ECMAScript 5	2009년 12월
ECMAScript 2015	2015년 6월
…	…
ECMAScript 2023	2023년 6월

Part 1

자바스크립트의 종류



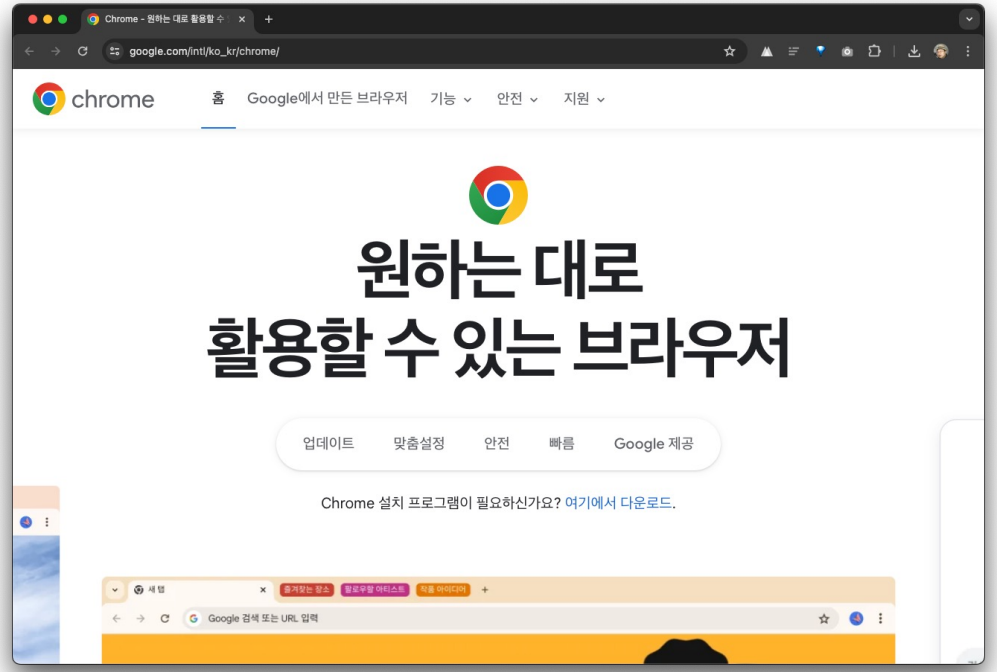
Part 1

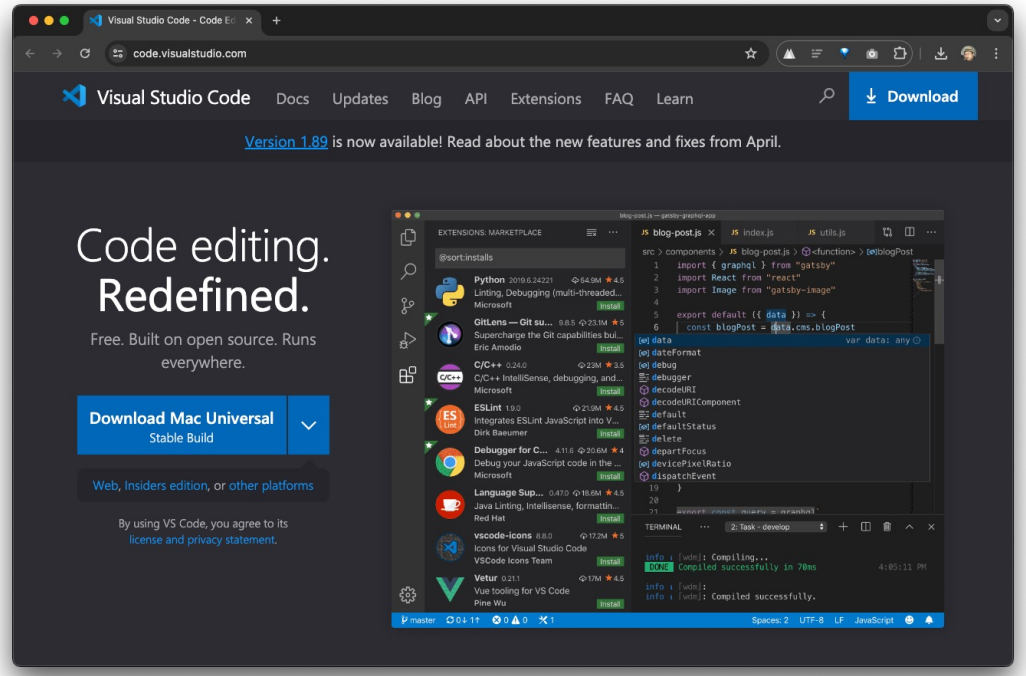
모바일 애플리케이션의 종류

개발환경 설정

Part 1

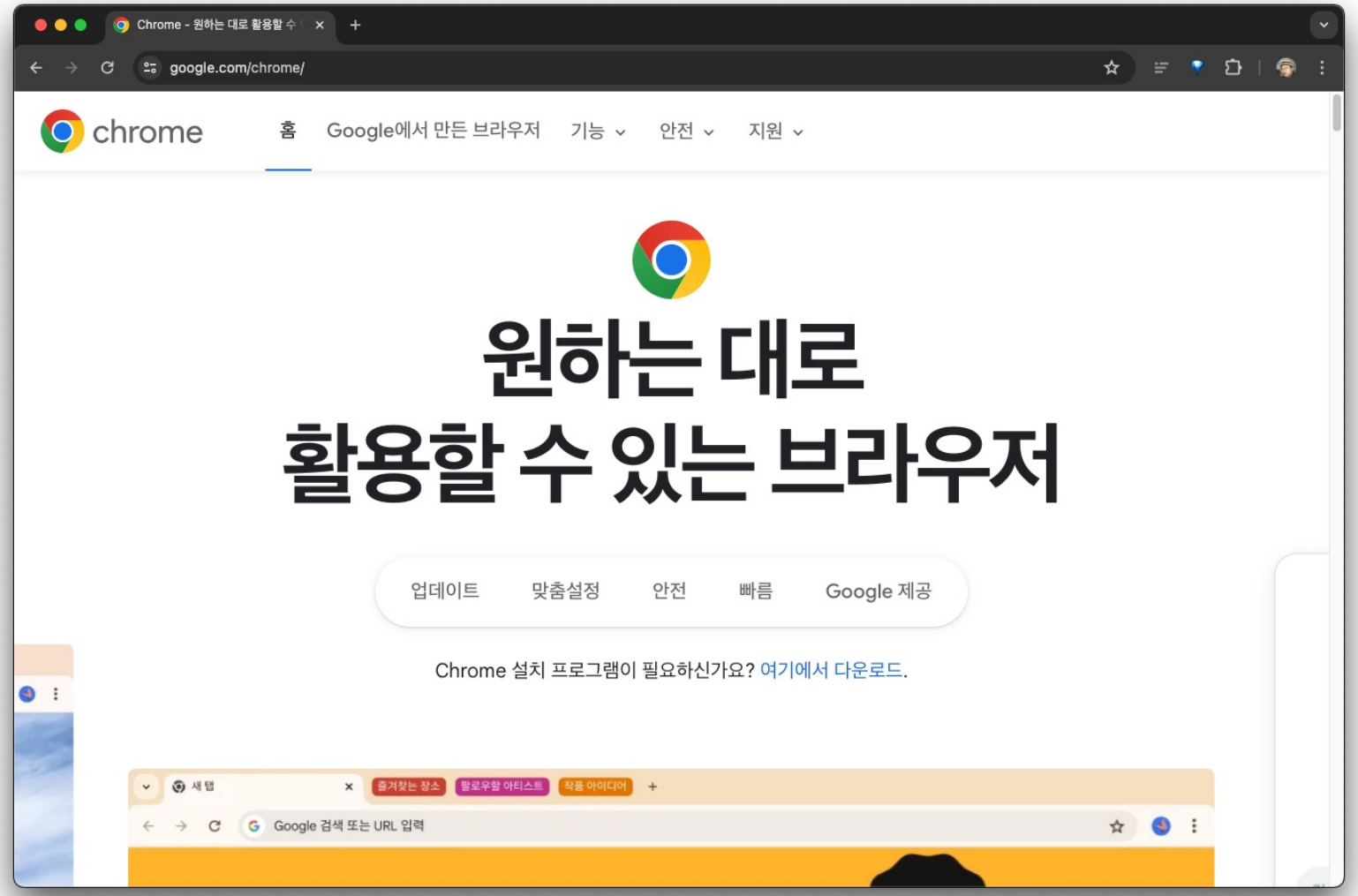
개발환경 설치와 코드 실행





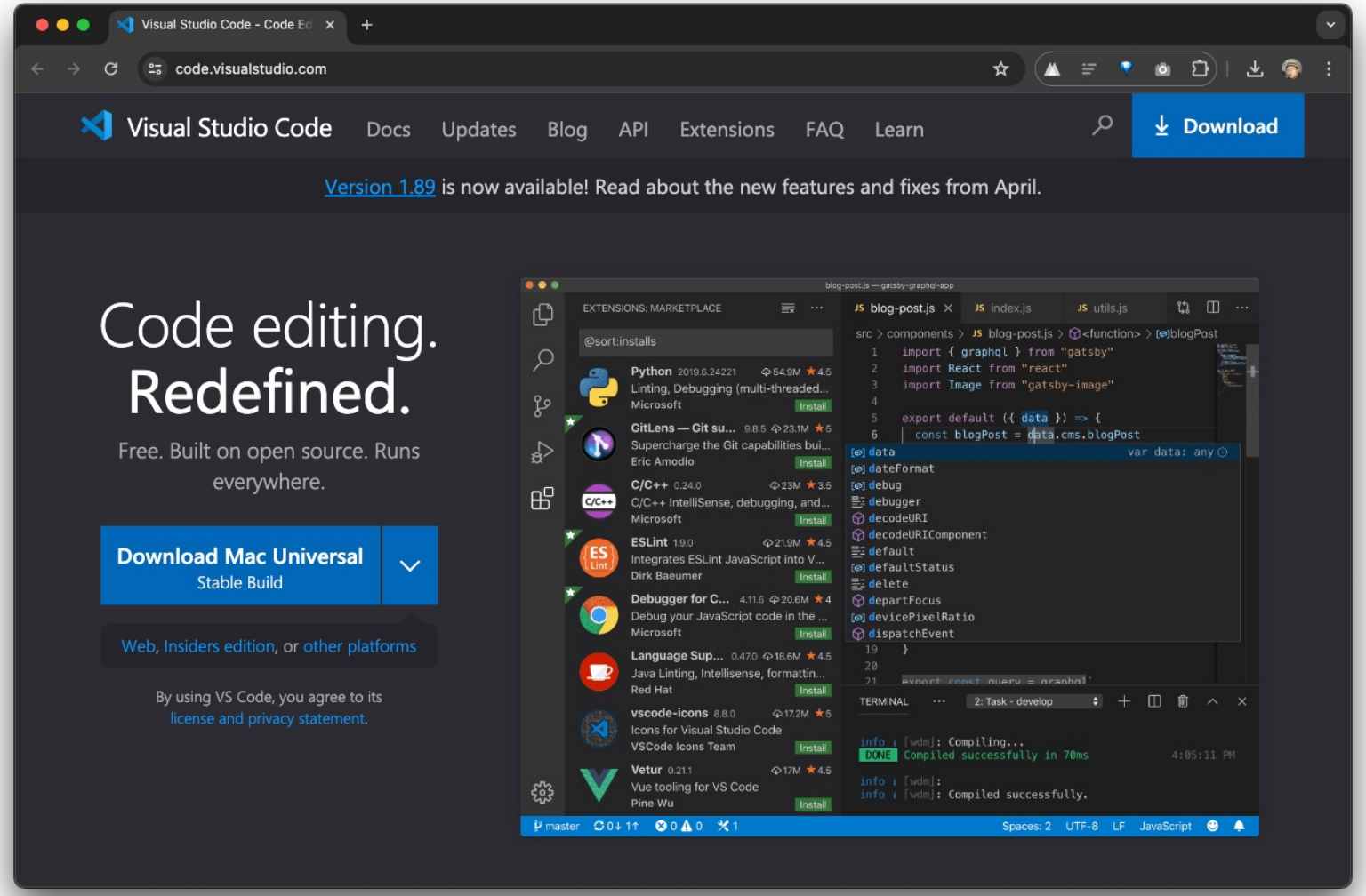
Part 1

구글 크롬 설치

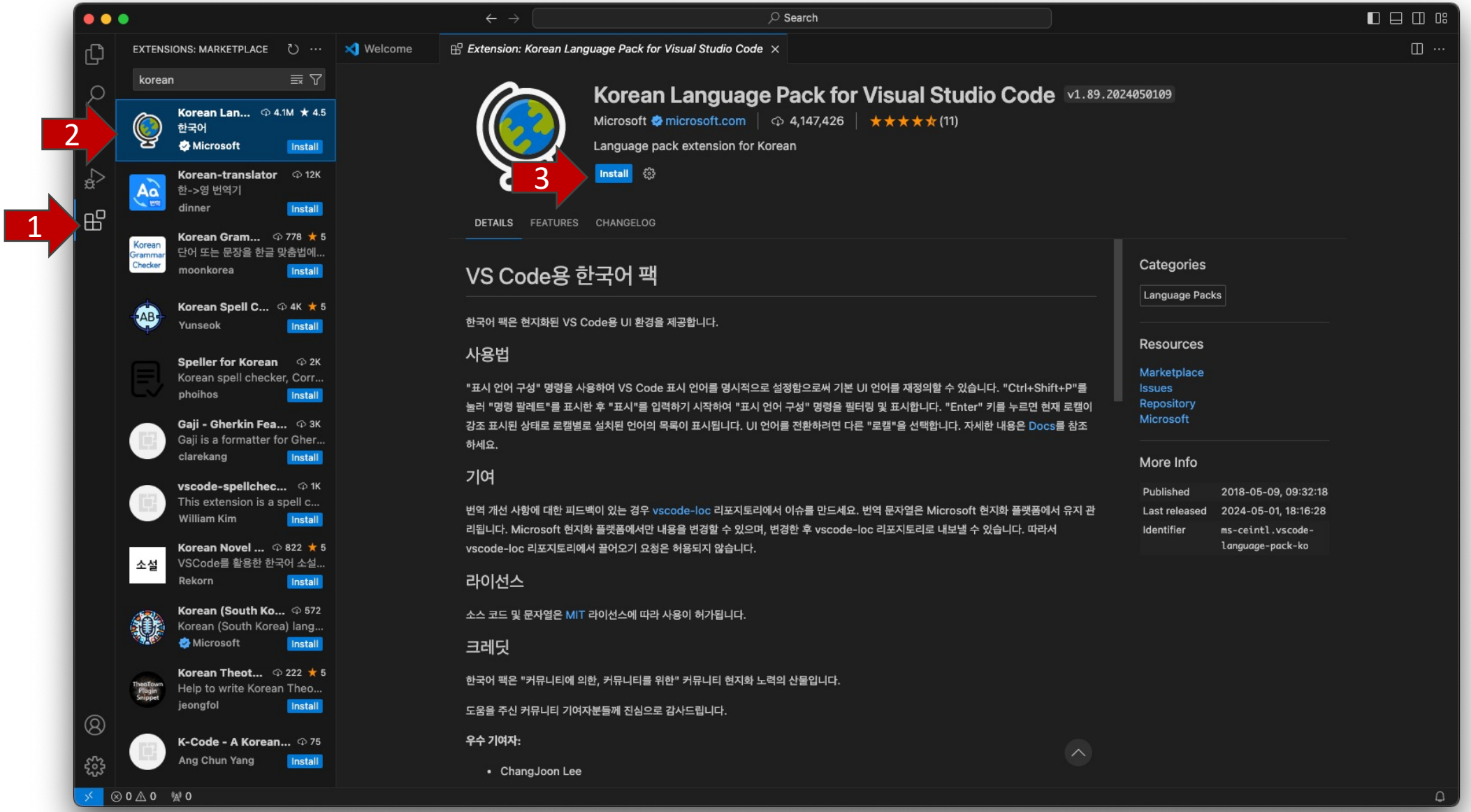


Part 1

Visual Studio Code 설치

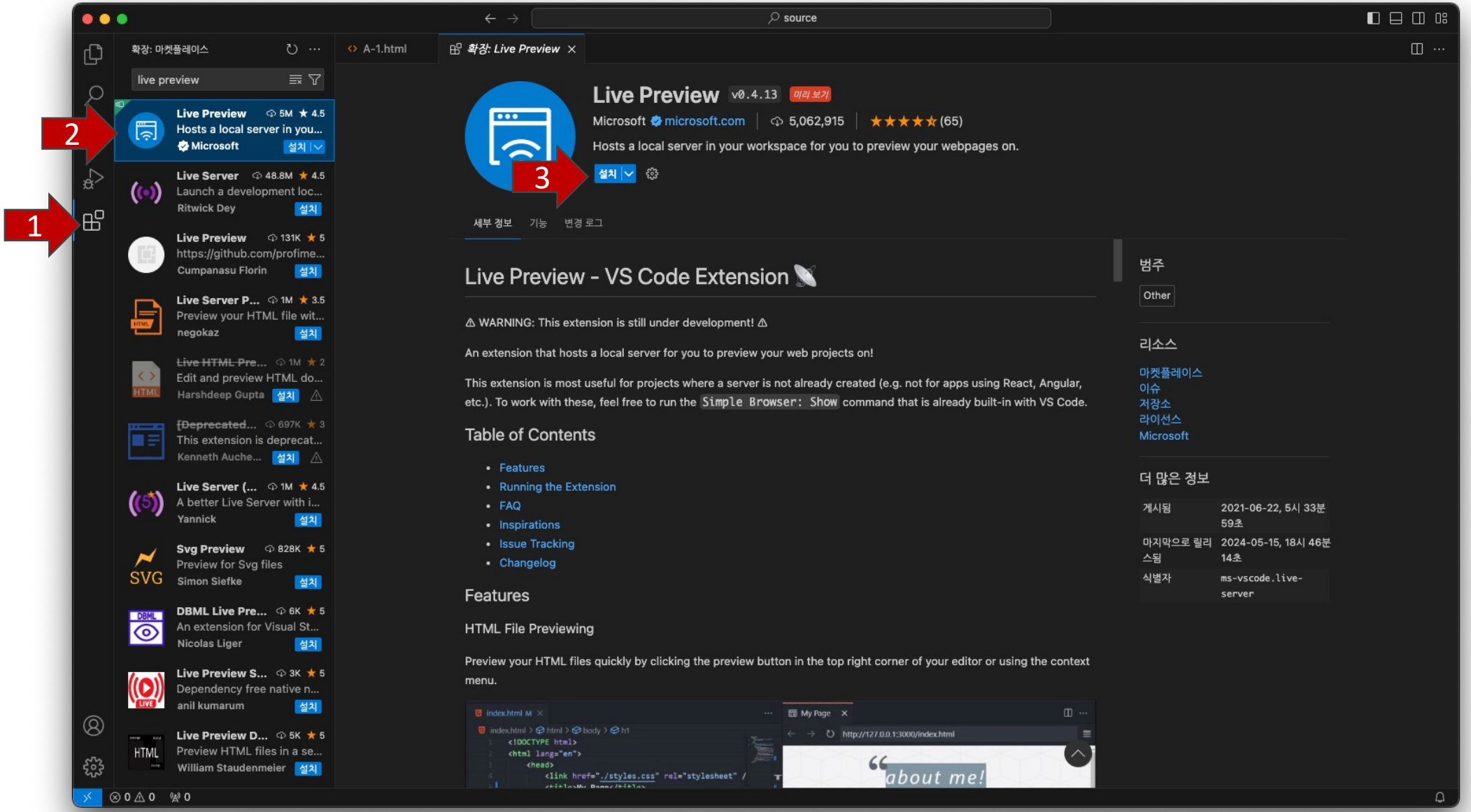


Part 1 Visual Studio Code 플러그인 설치



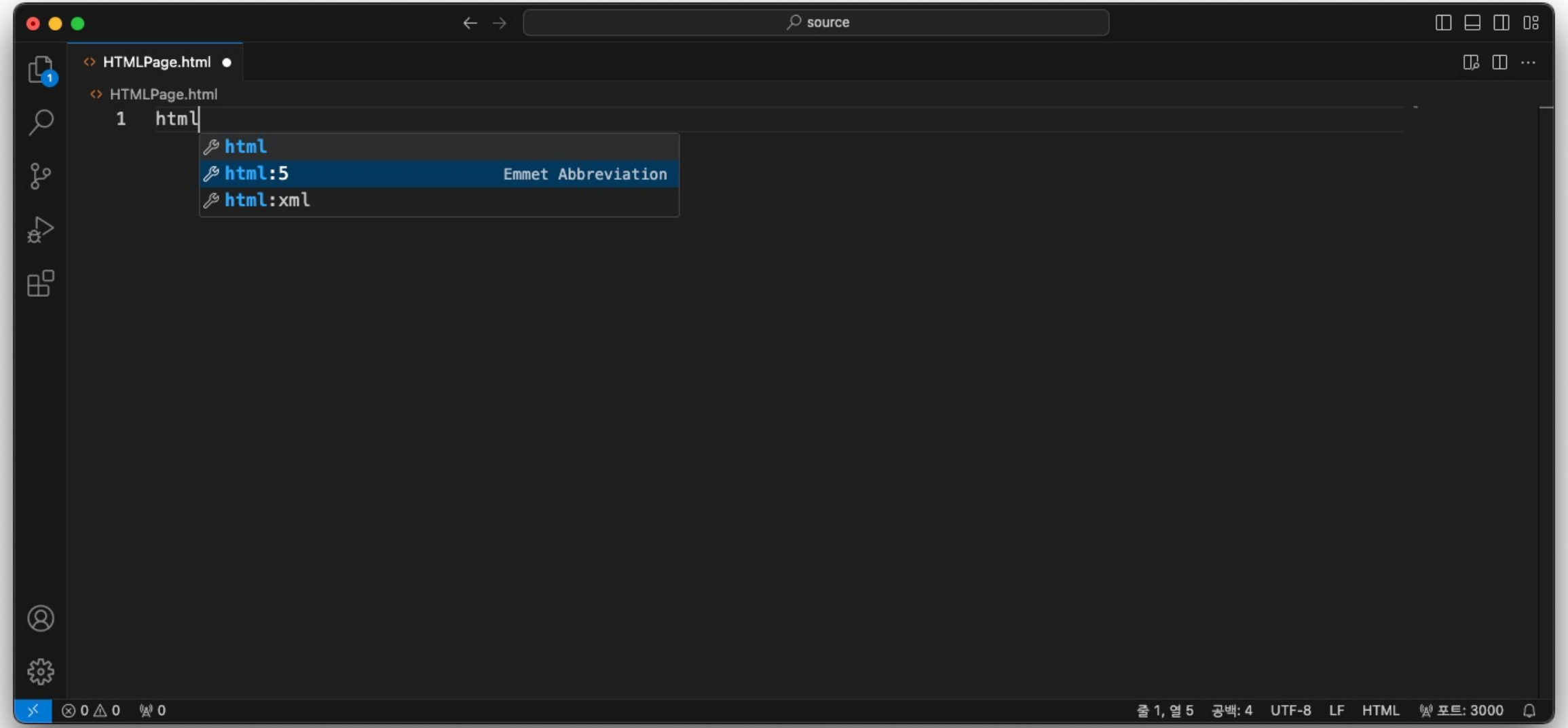
Part 1

Visual Studio Code 플러그인 설치



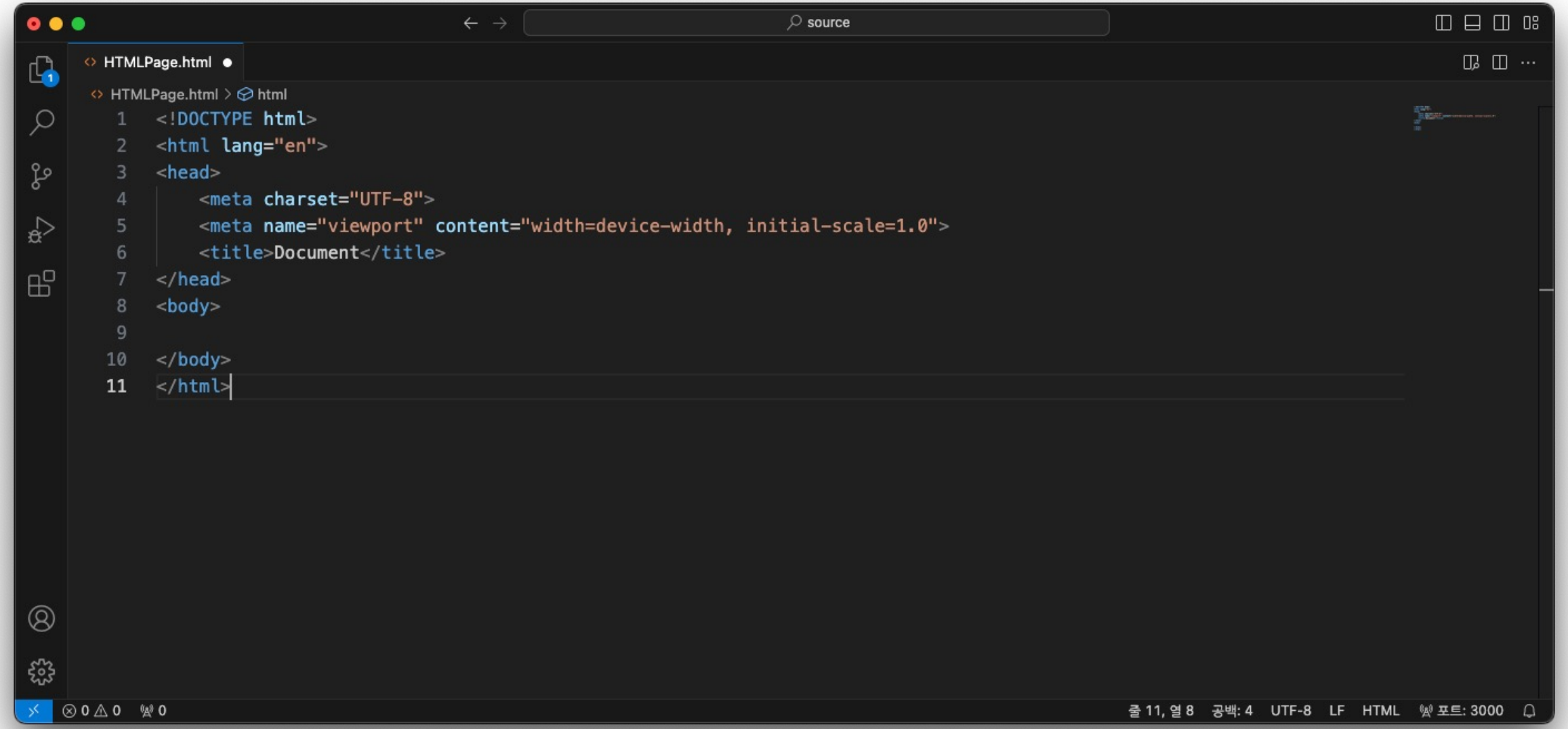
Part 1

HTML 페이지 작성



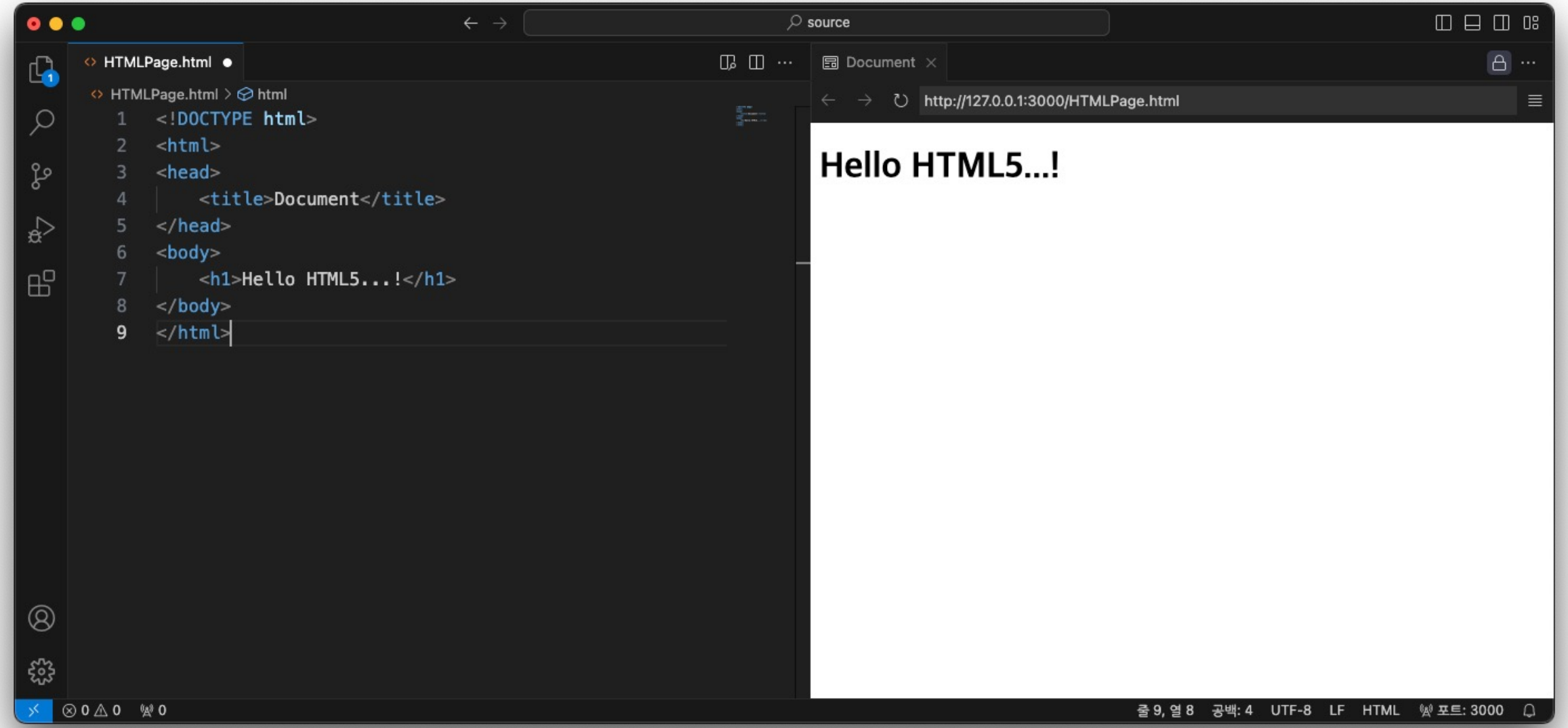
Part 1

HTML 페이지 작성



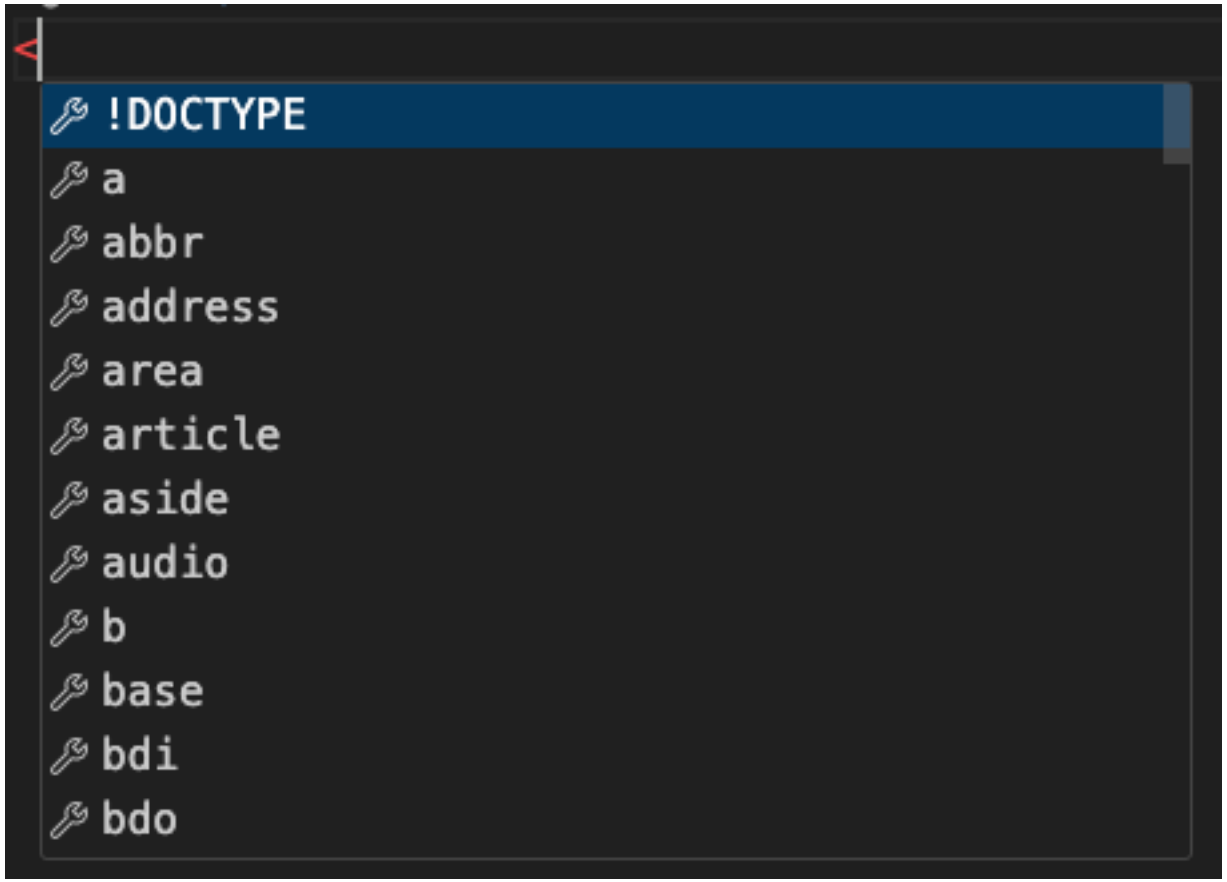
Part 1

HTML 페이지 작성



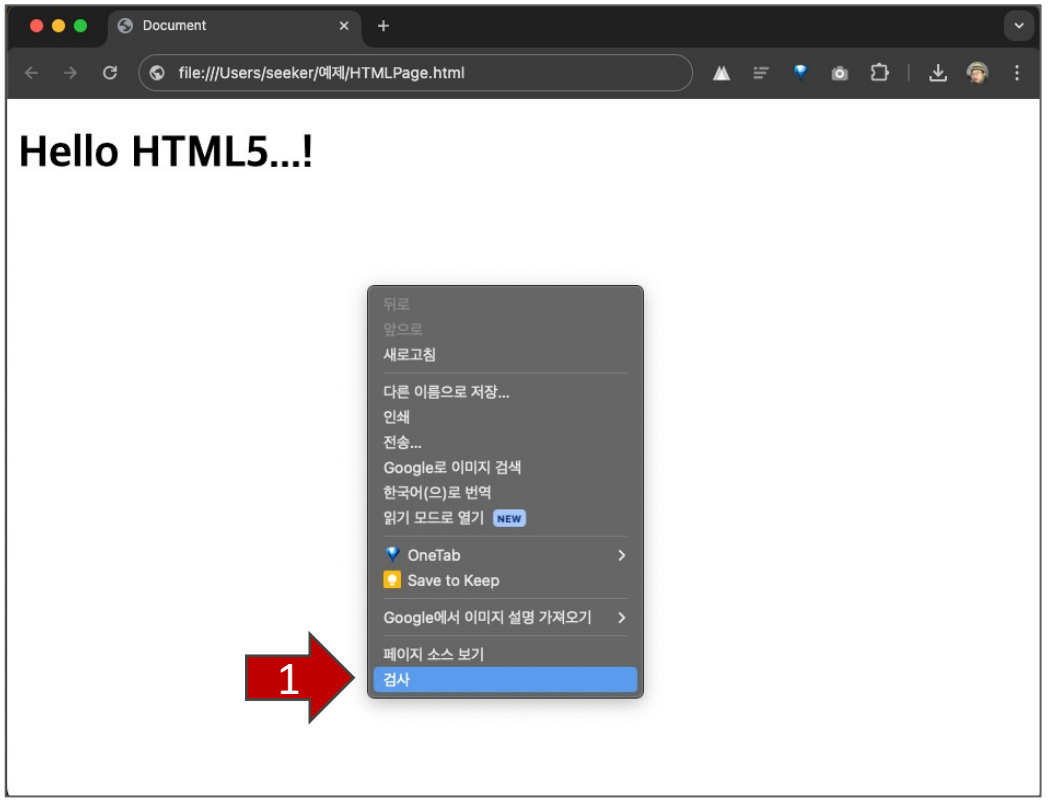
Part 1

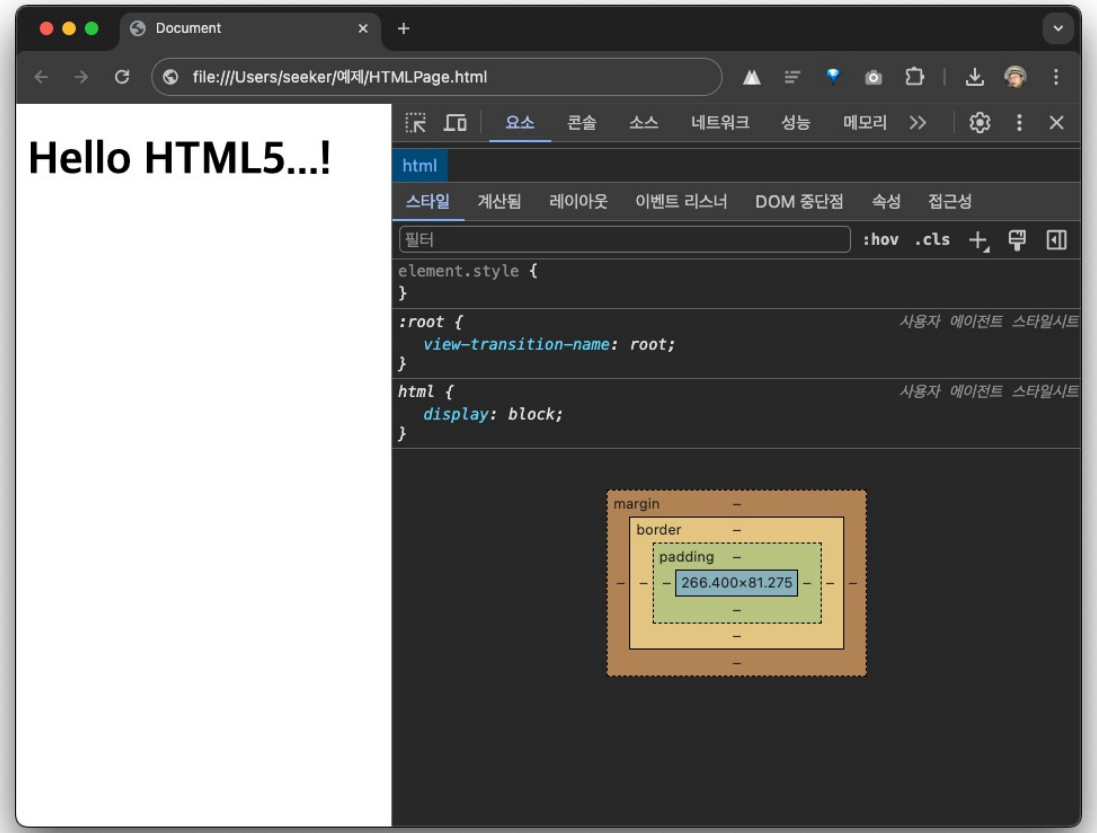
자동 완성 기능



Part 1

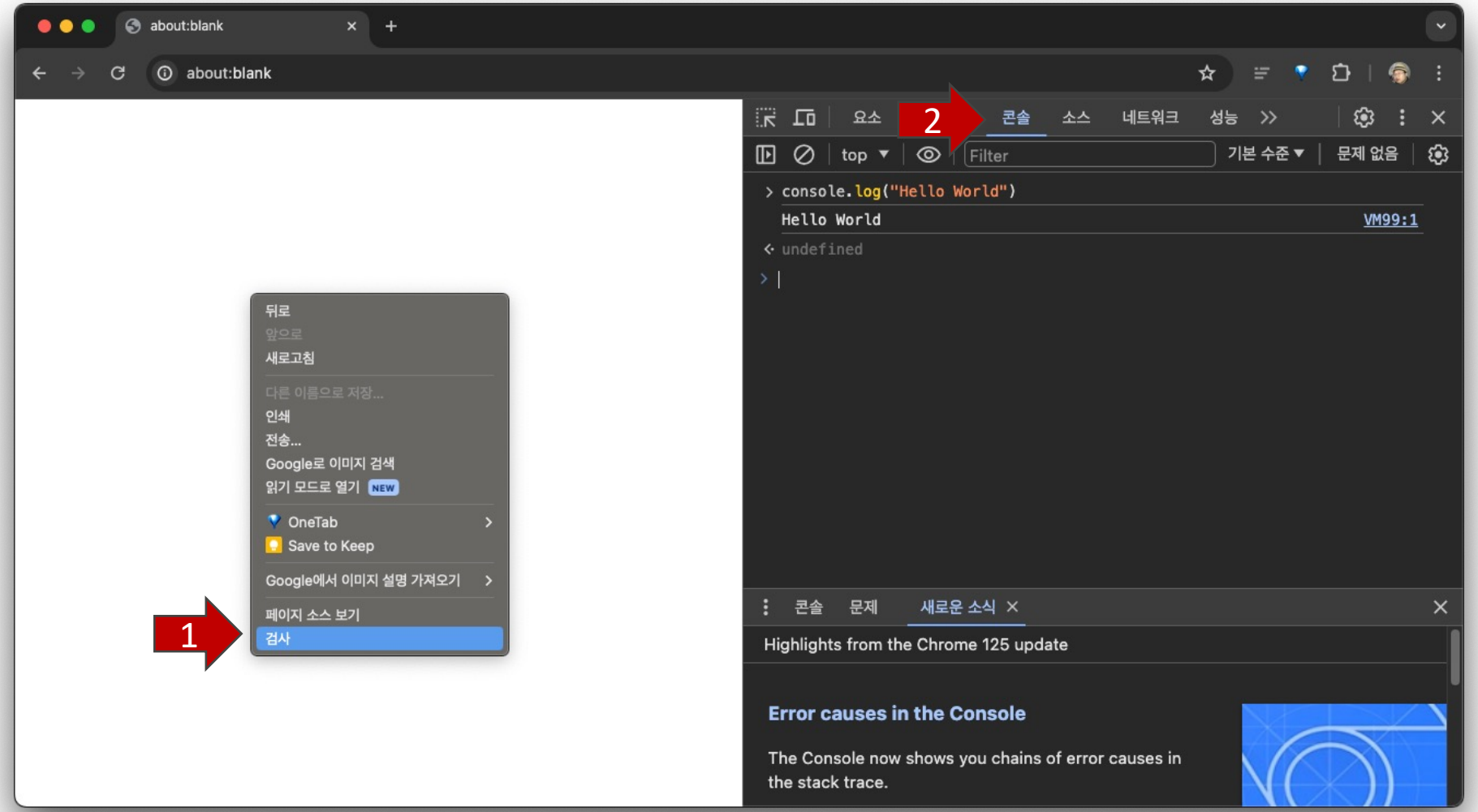
구글 크롬 개발자 도구





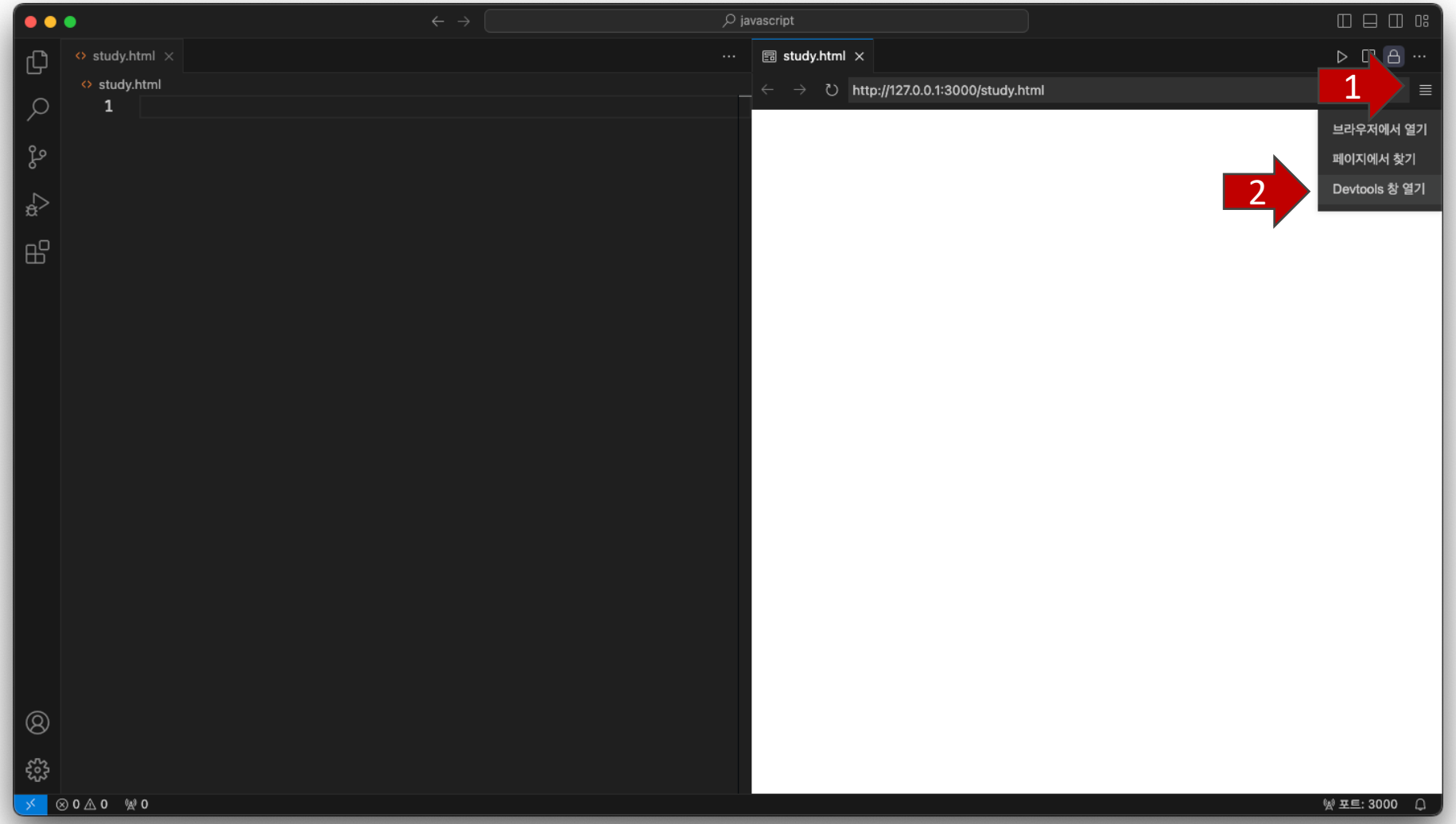
Part 1

구글 크롬 개발자 도구



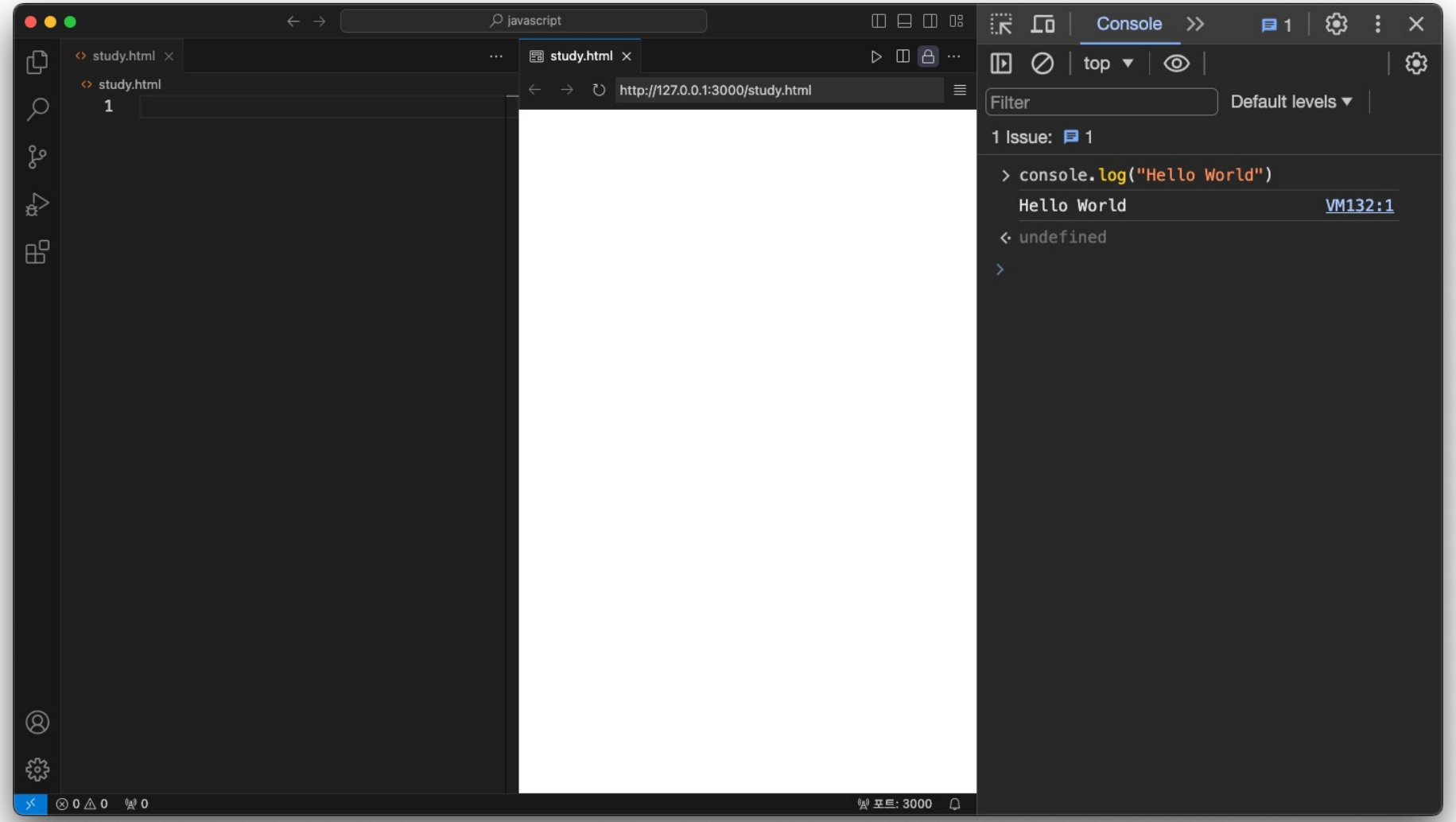
Part 1

Live Preview 개발자 도구



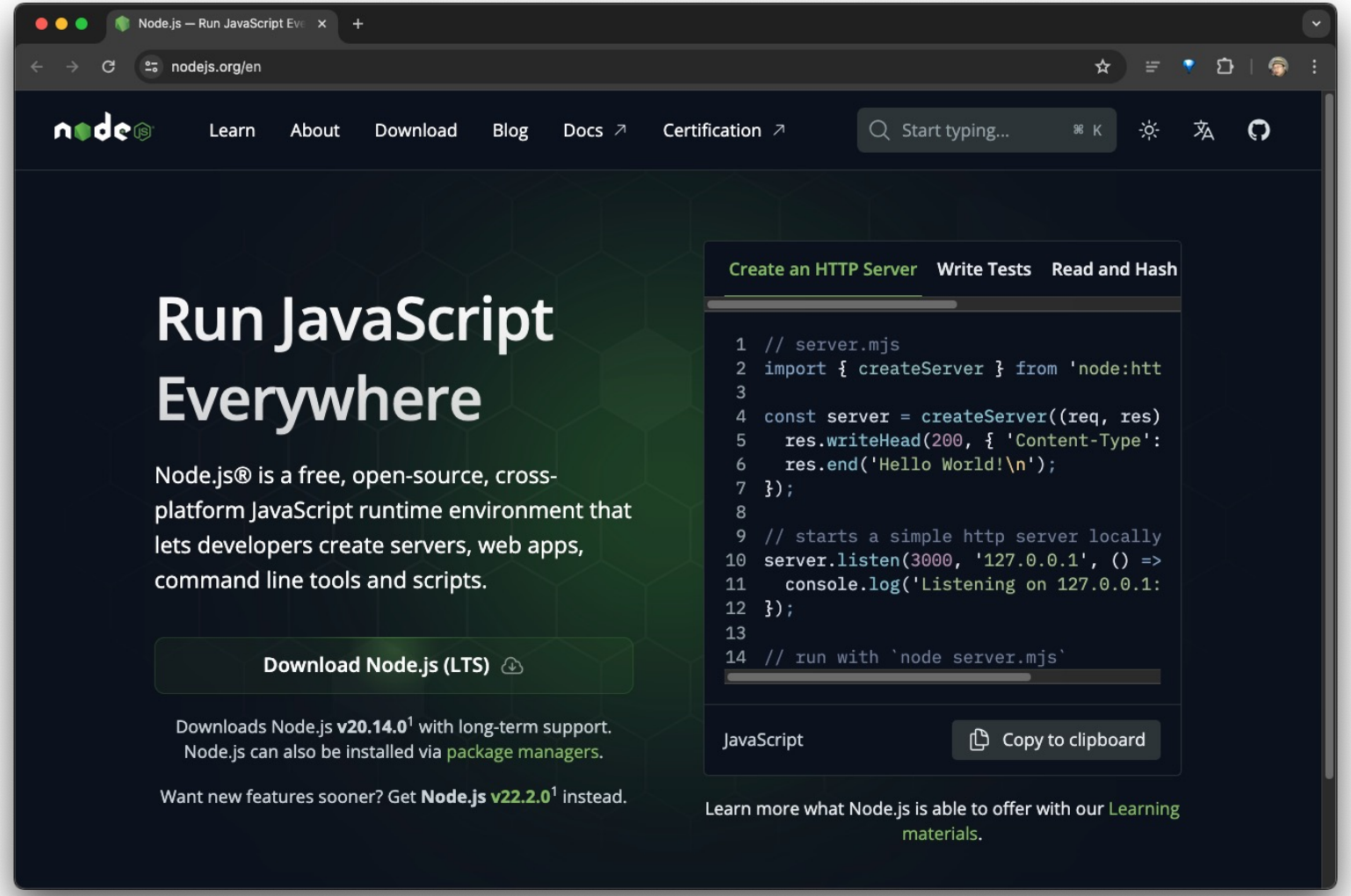
Part 1

Live Preview 개발자 도구

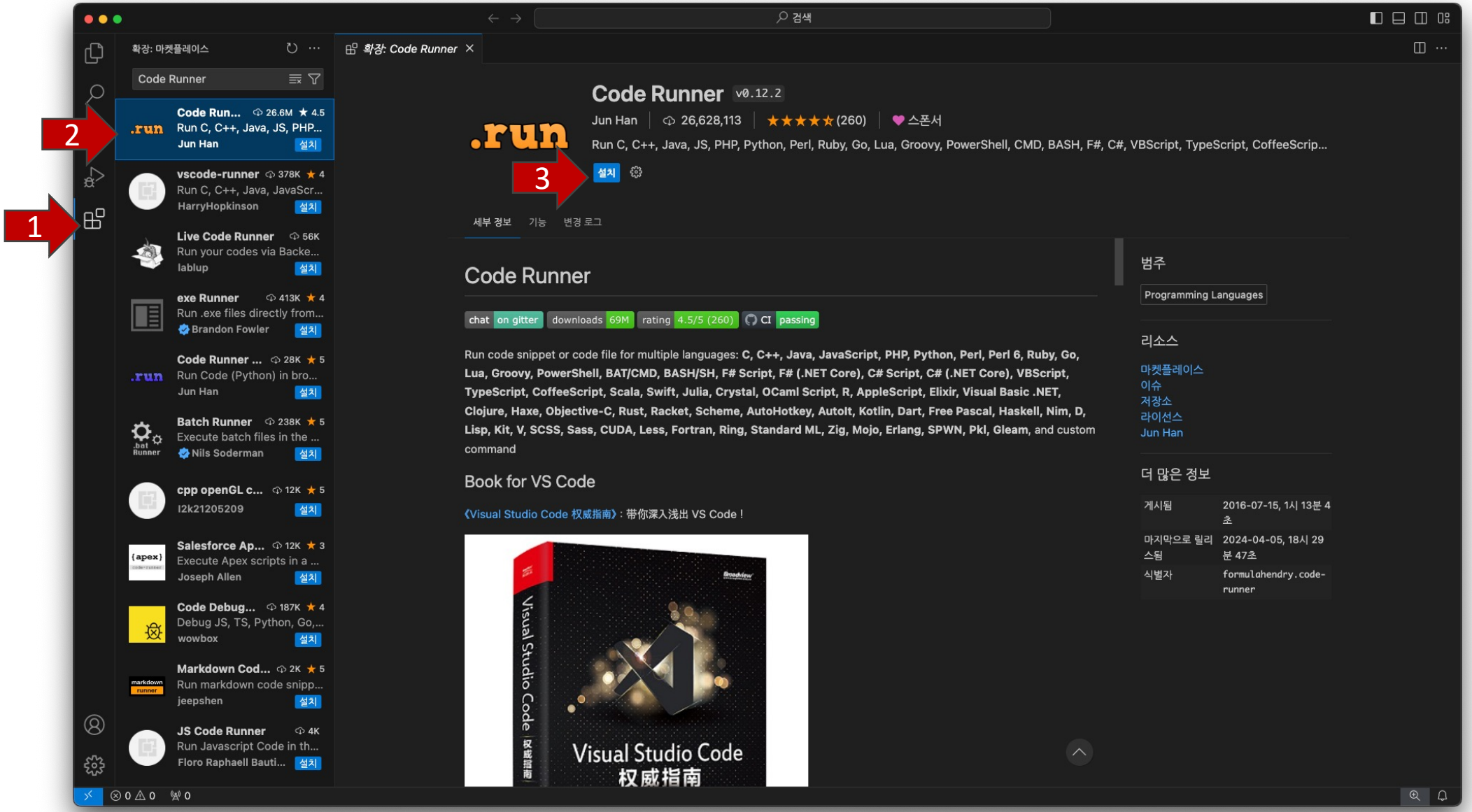


Part 1

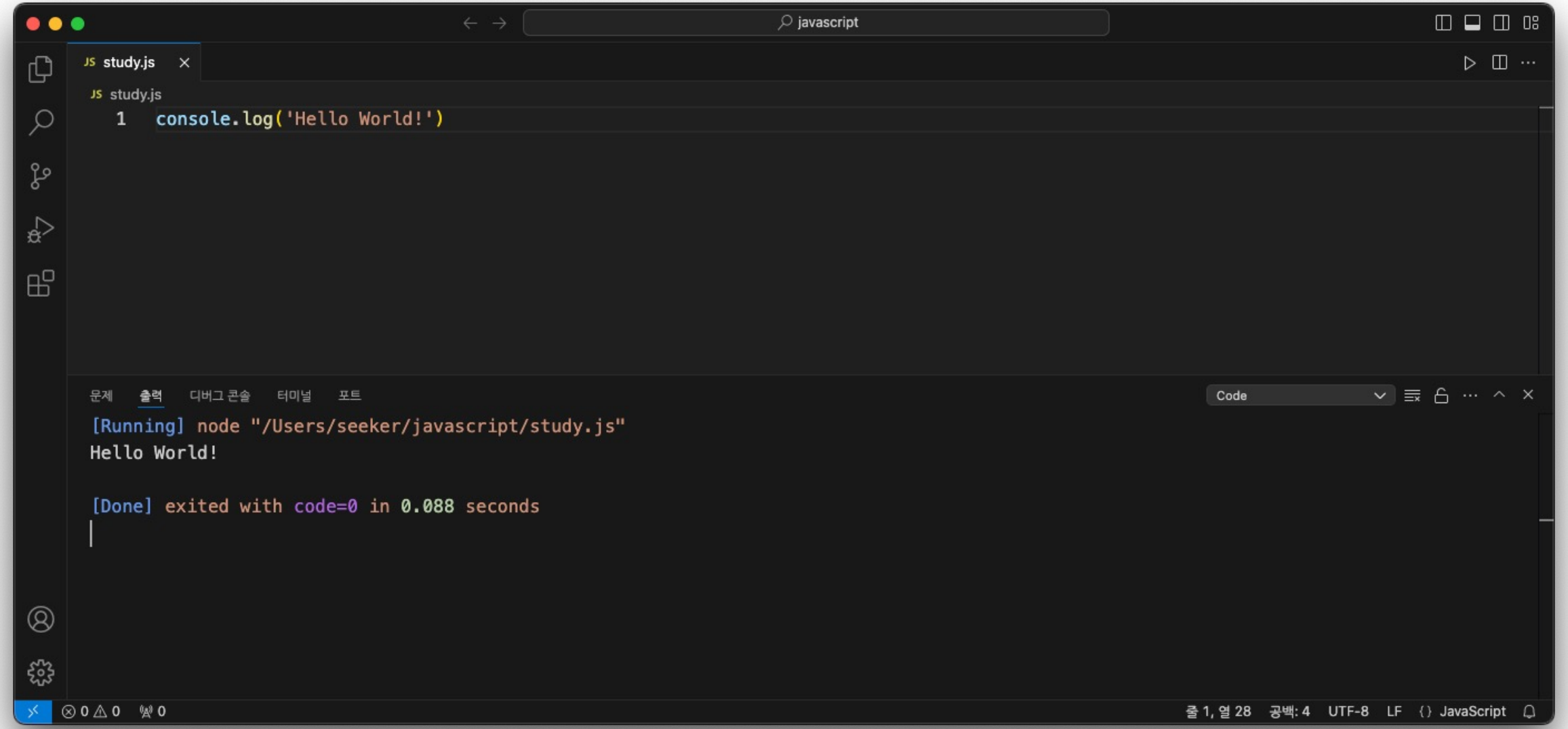
Node.js 설치



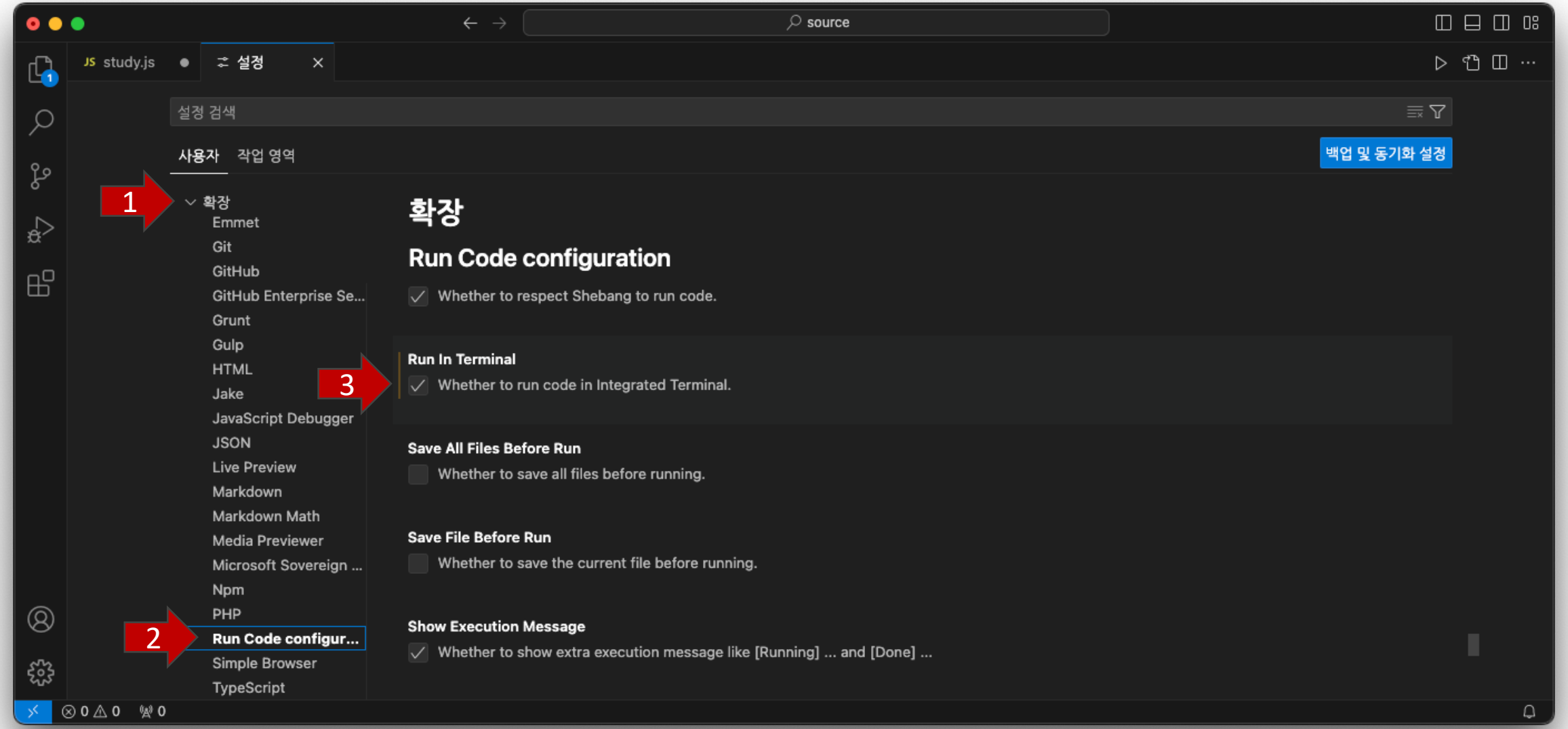
Part 1 Visual Studio Code 플러그인 설치



Part 1 JavaScript 작성

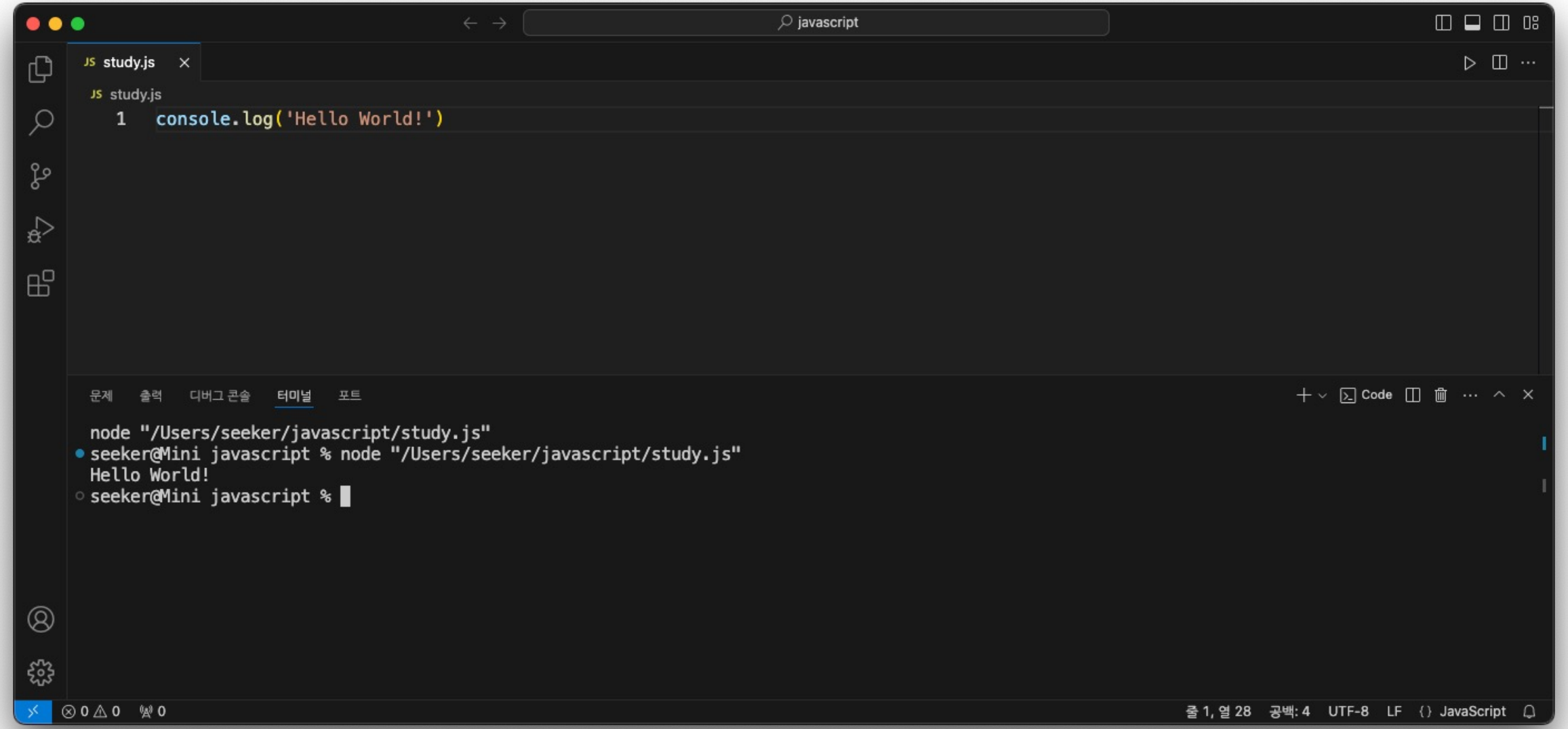


Part 1 Code Runner 환경설정 – 터미널 활성화



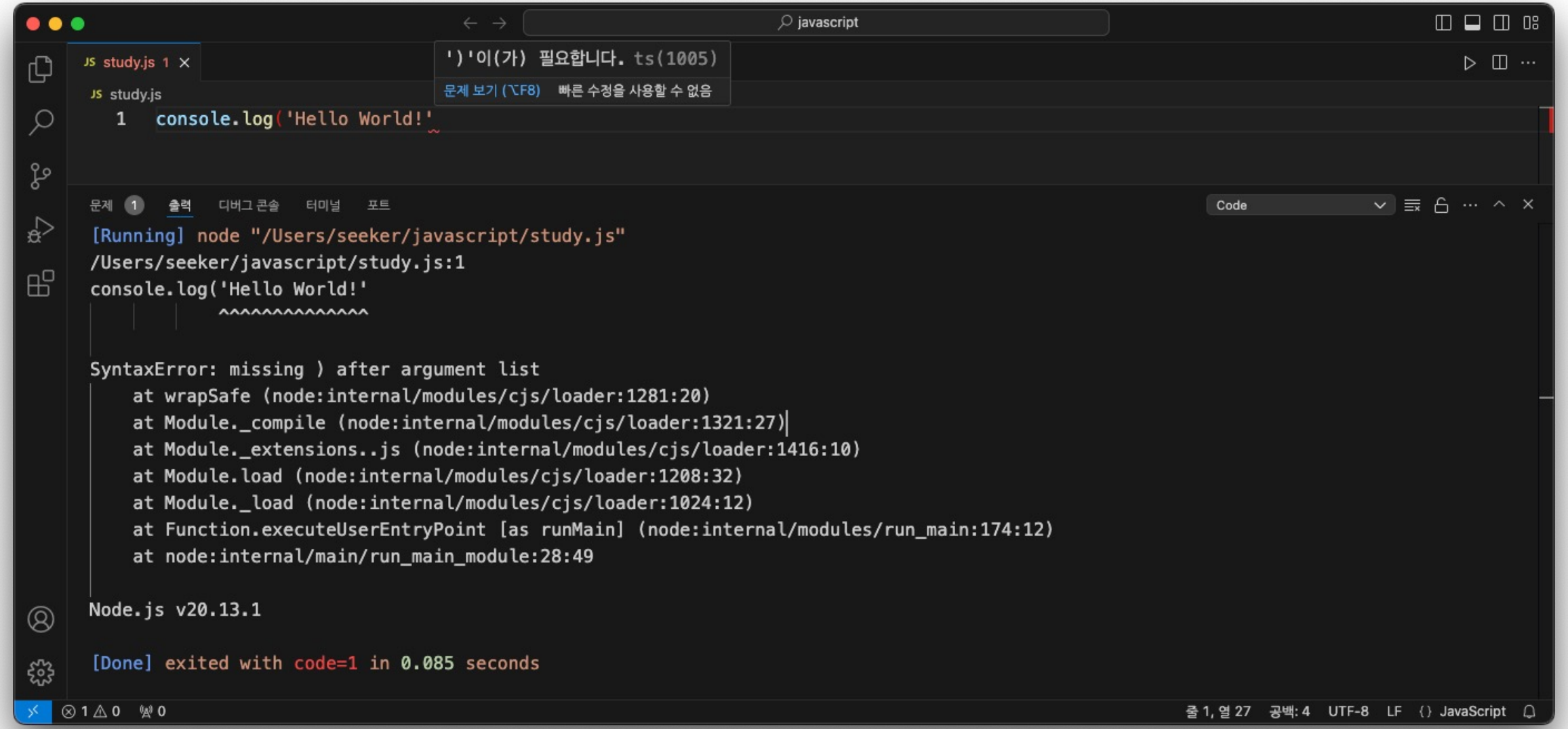
Part 1

터미널에서 JavaScript 실행

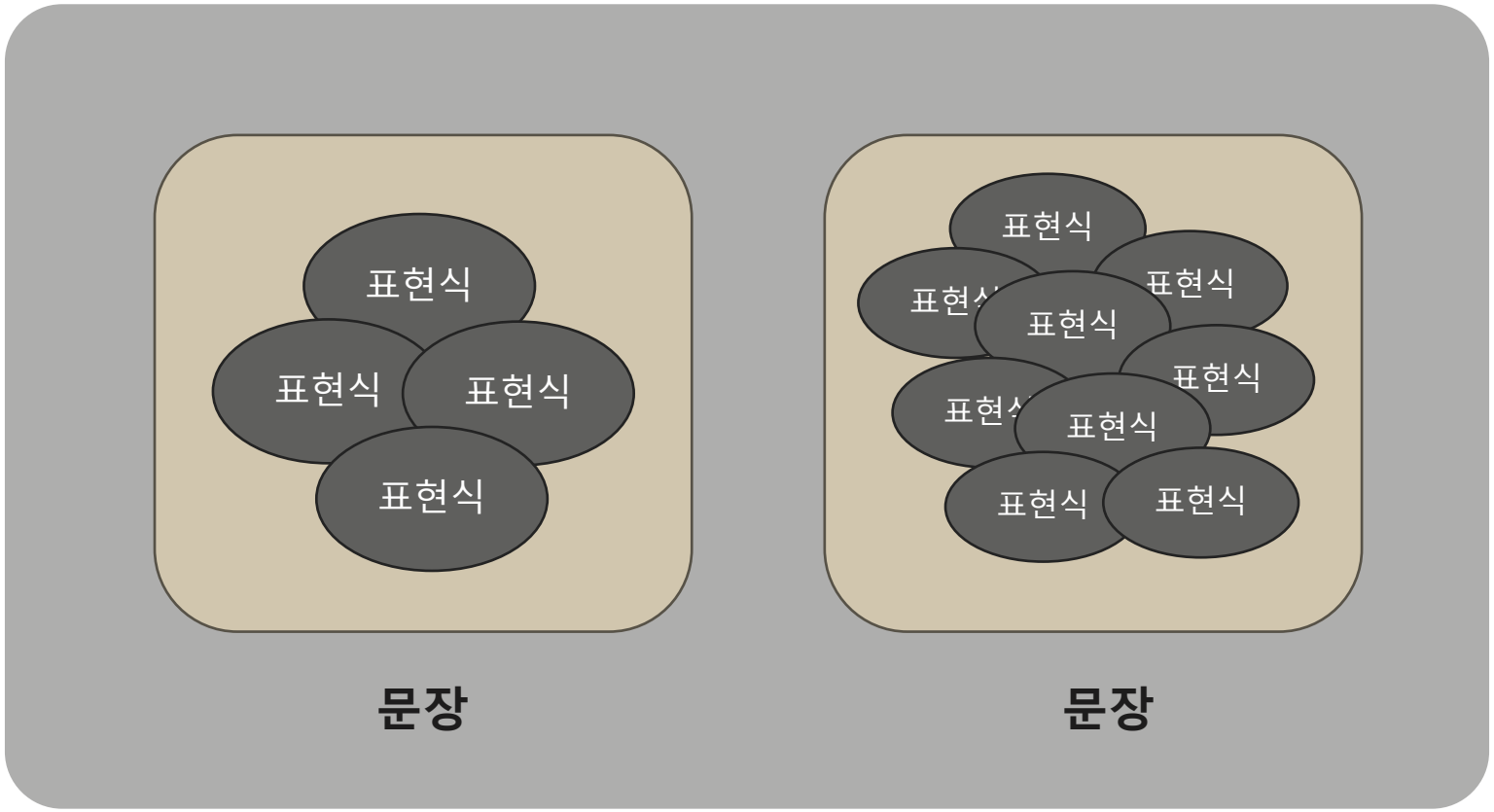


Part 1

JavaScript 오류 확인

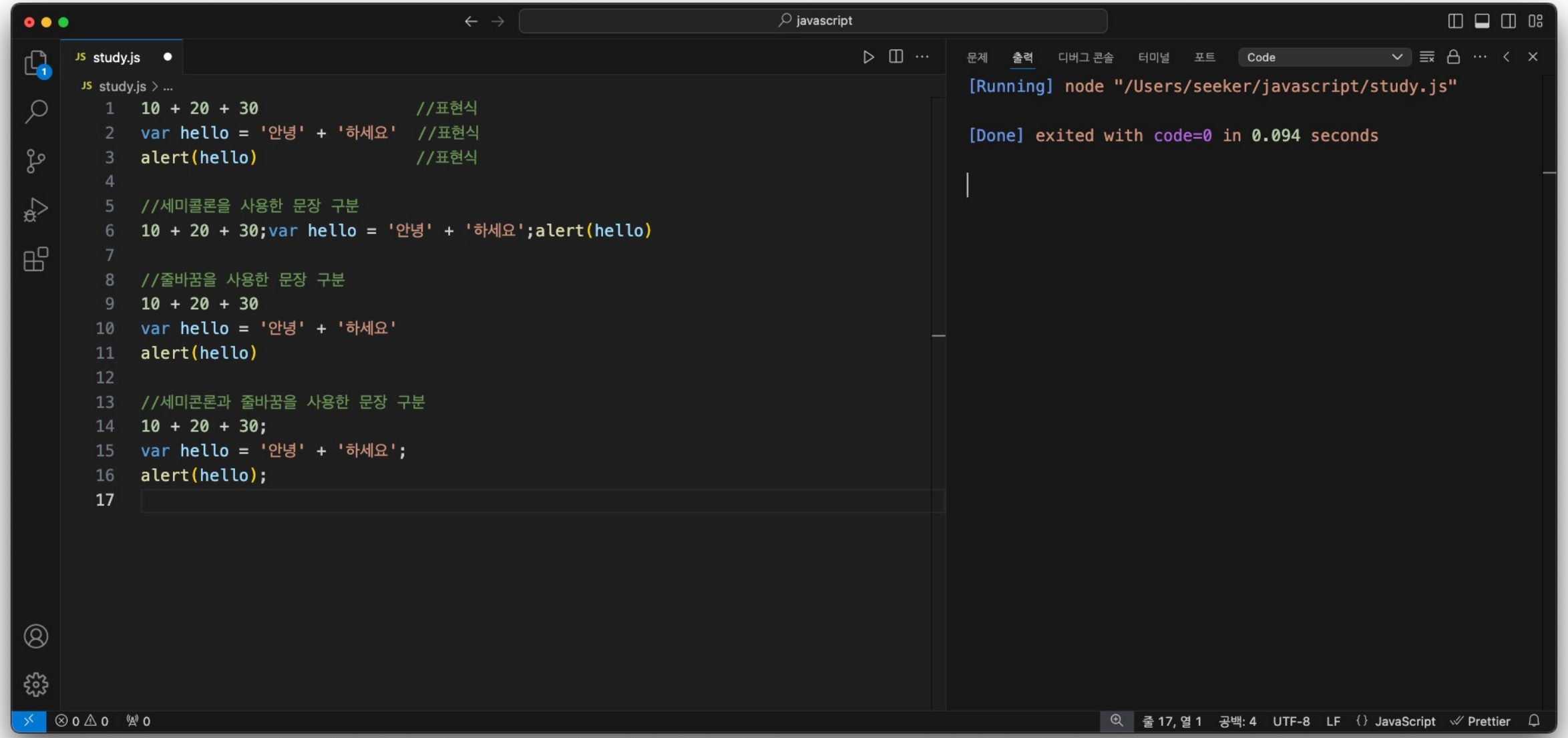


기본 용어



프로그램

Part 1 표현식과 문장



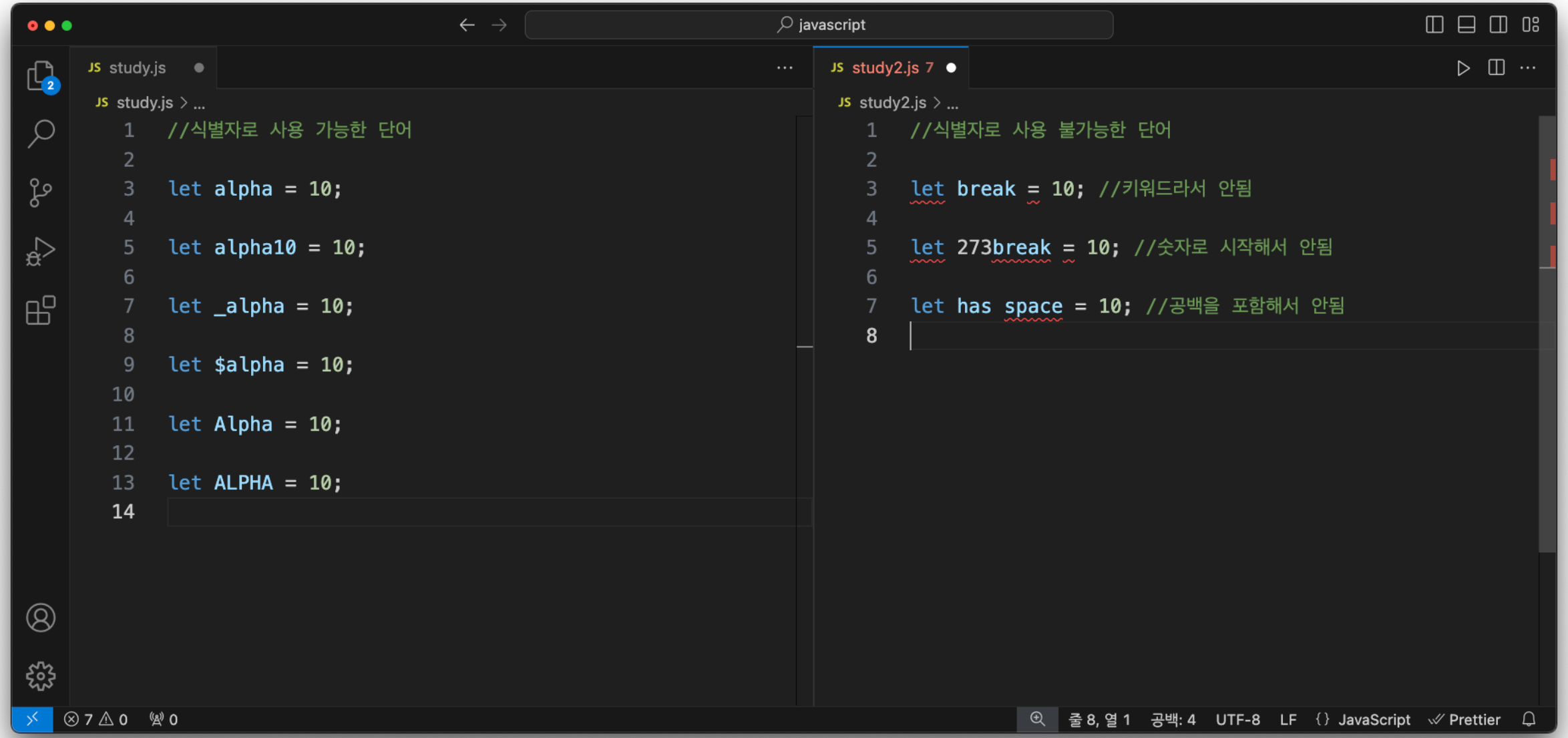
Part 1

키워드

await	break	case	catch
class	const	continue	debugger
default	delete	do	else
export	extends	finally	for
function	if	import	in
instanceof	new	return	super
switch	this	throw	try
typeof	var	void	while
with	yield	let	static
true	false	null	as
from	get	of	set
target			

Part 1

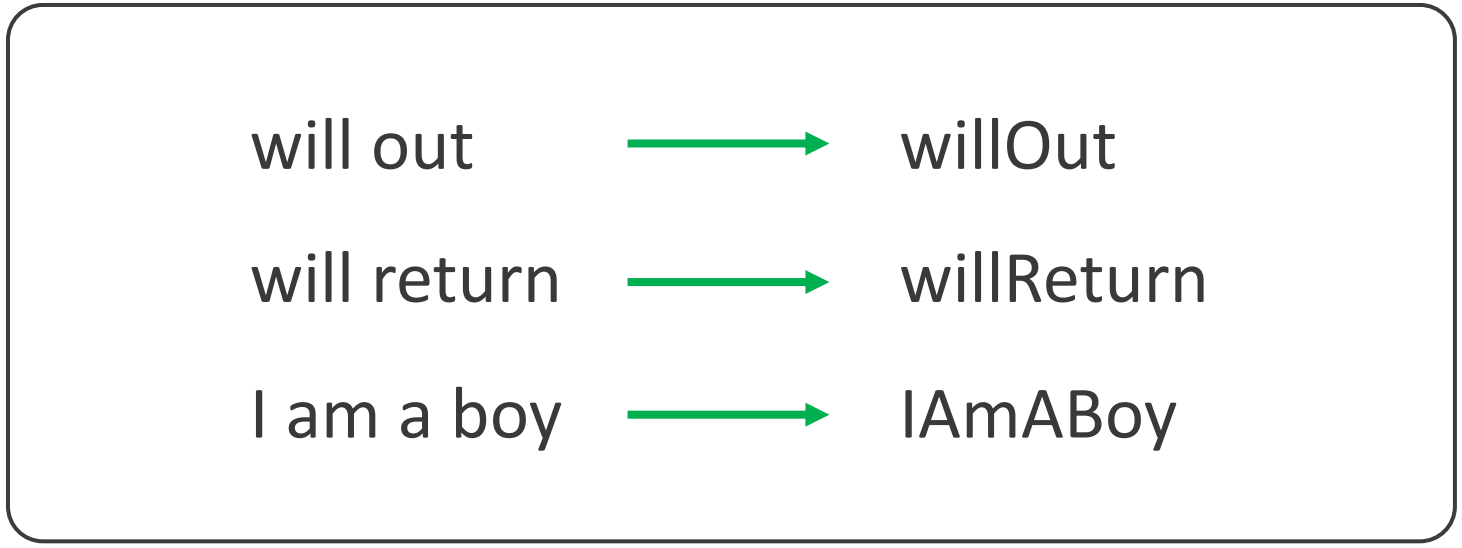
식별자


javascript
study.js
//식별자로 사용 가능한 단어
let alpha = 10;
let alpha10 = 10;
let _alpha = 10;
let $alpha = 10;
let Alpha = 10;
let ALPHA = 10;
study2.js 7
//식별자로 사용 불가능한 단어
let break = 10; //키워드라서 안됨
let 273break = 10; //숫자로 시작해서 안됨
let has space = 10; //공백을 포함해서 안됨
줄 8, 열 1
공백: 4
UTF-8
LF
JavaScript
Prettier

Part 1

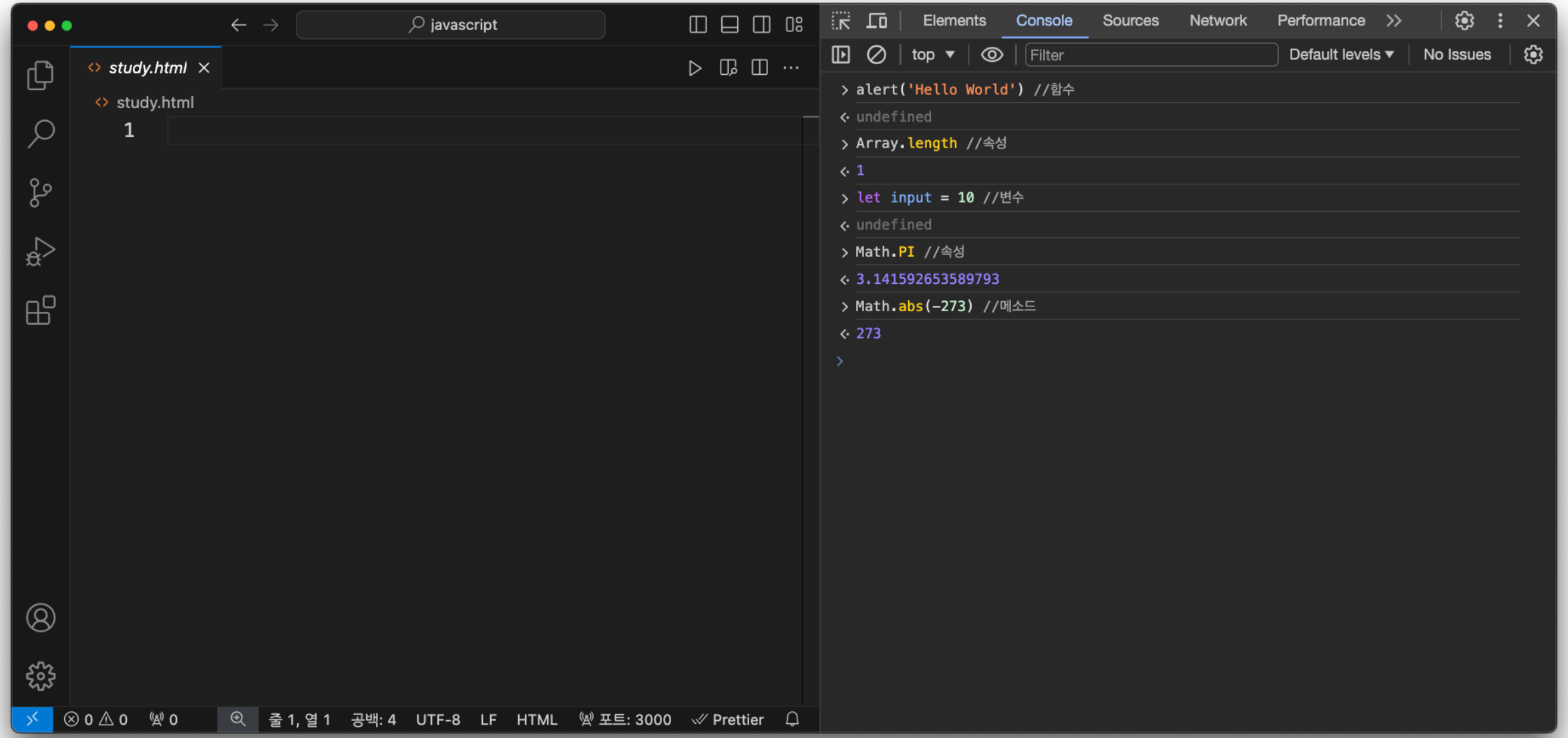
식별자 > 식별자를 만드는 일반적인 관례

- **클래스**의 이름은 항상 대문자로 시작합니다.
- **변수**와 **인스턴스**, **함수**, **메소드**의 이름은 항상 소문자로 시작합니다.
- 여러 단어로 이루어진 **식별자**는 각 단어의 첫 글자를 대문자로 합니다.



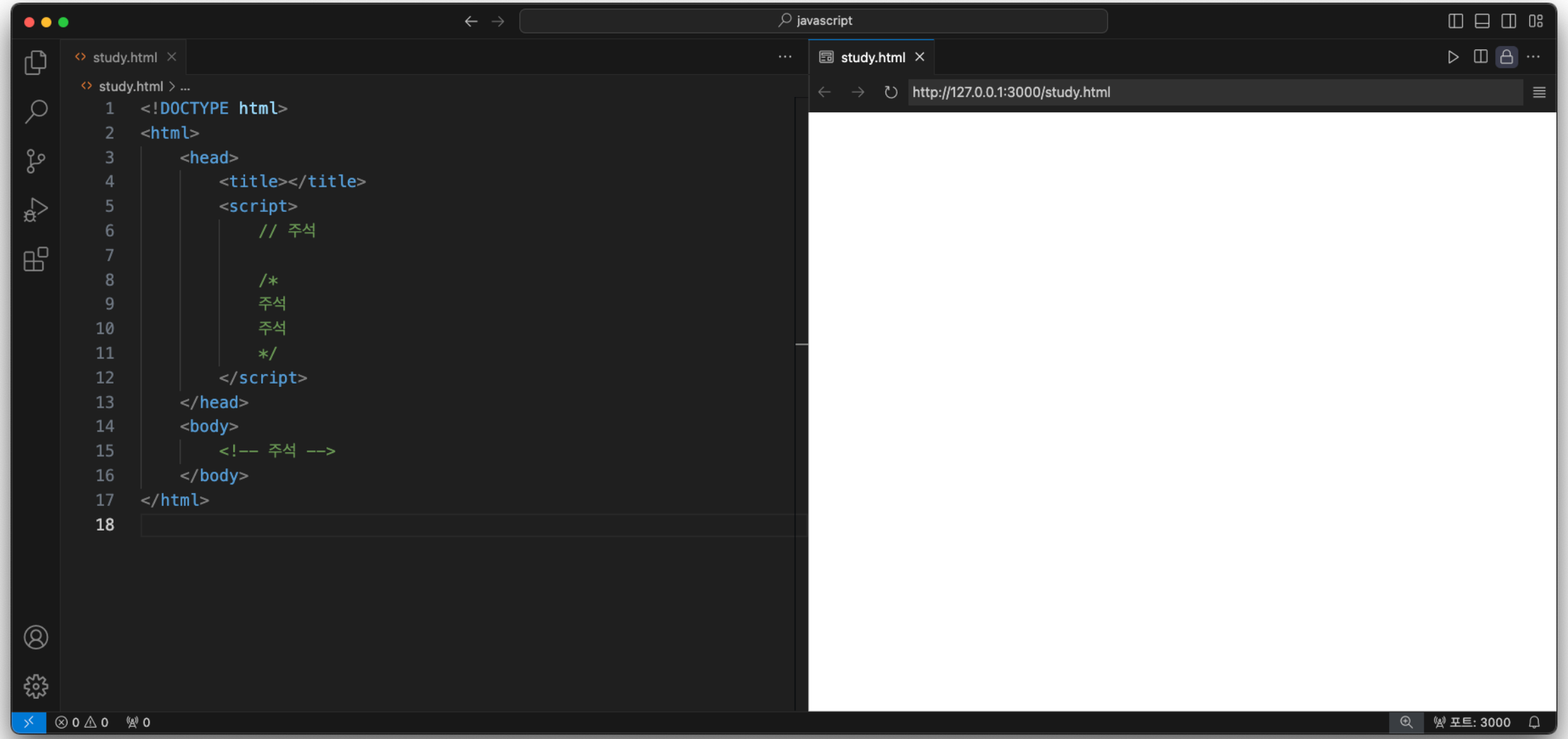
Part 1

식별자 > 식별자의 종류

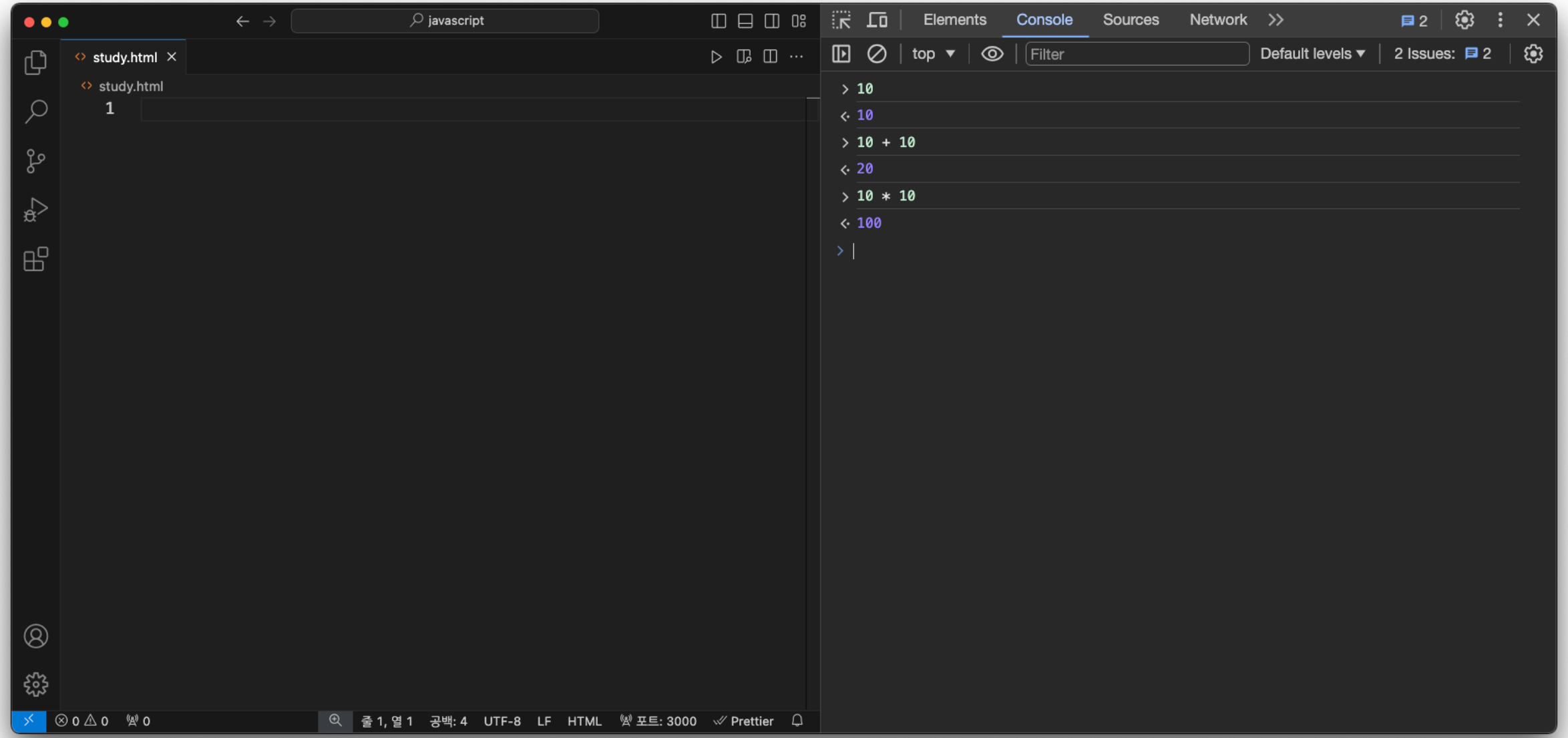


Part 1

주석

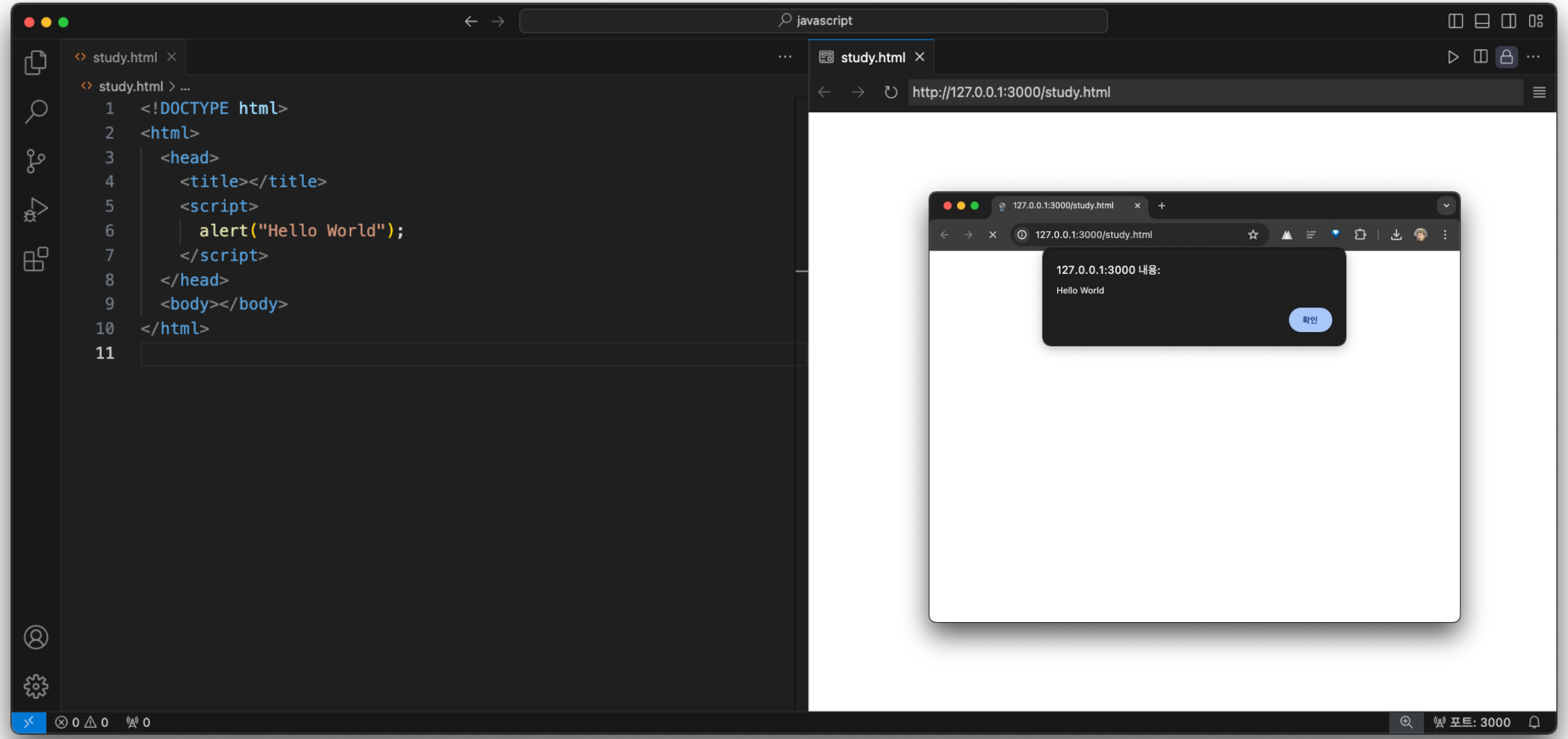

javascript
study.html
study.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title></title>
5 <script>
6 // 주석
7
8 /*
9 주석
10 주석
11 */
12 </script>
13 </head>
14 <body>
15 <!-- 주석 -->
16 </body>
17 </html>
18
study.html
http://127.0.0.1:3000/study.html
0 0 0
포트: 3000

Part 1 출력


javascript
study.html
study.html
1
Elements
Console
Sources
Network
top
Filter
Default levels
2 Issues: 2
10
10
10 + 10
20
10 * 10
100
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3000
Prettier

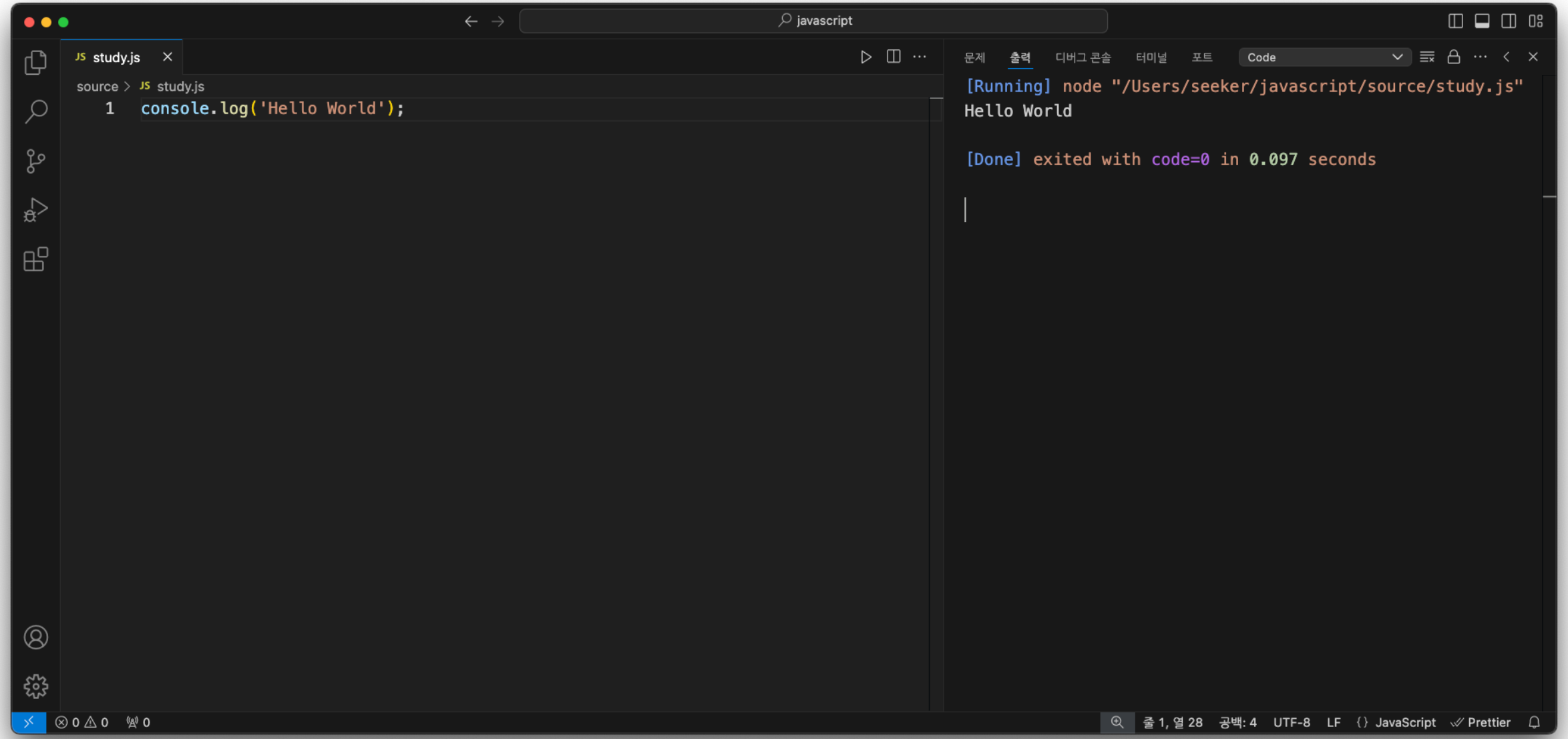
Part 1

출력 > 경고창에 출력하기

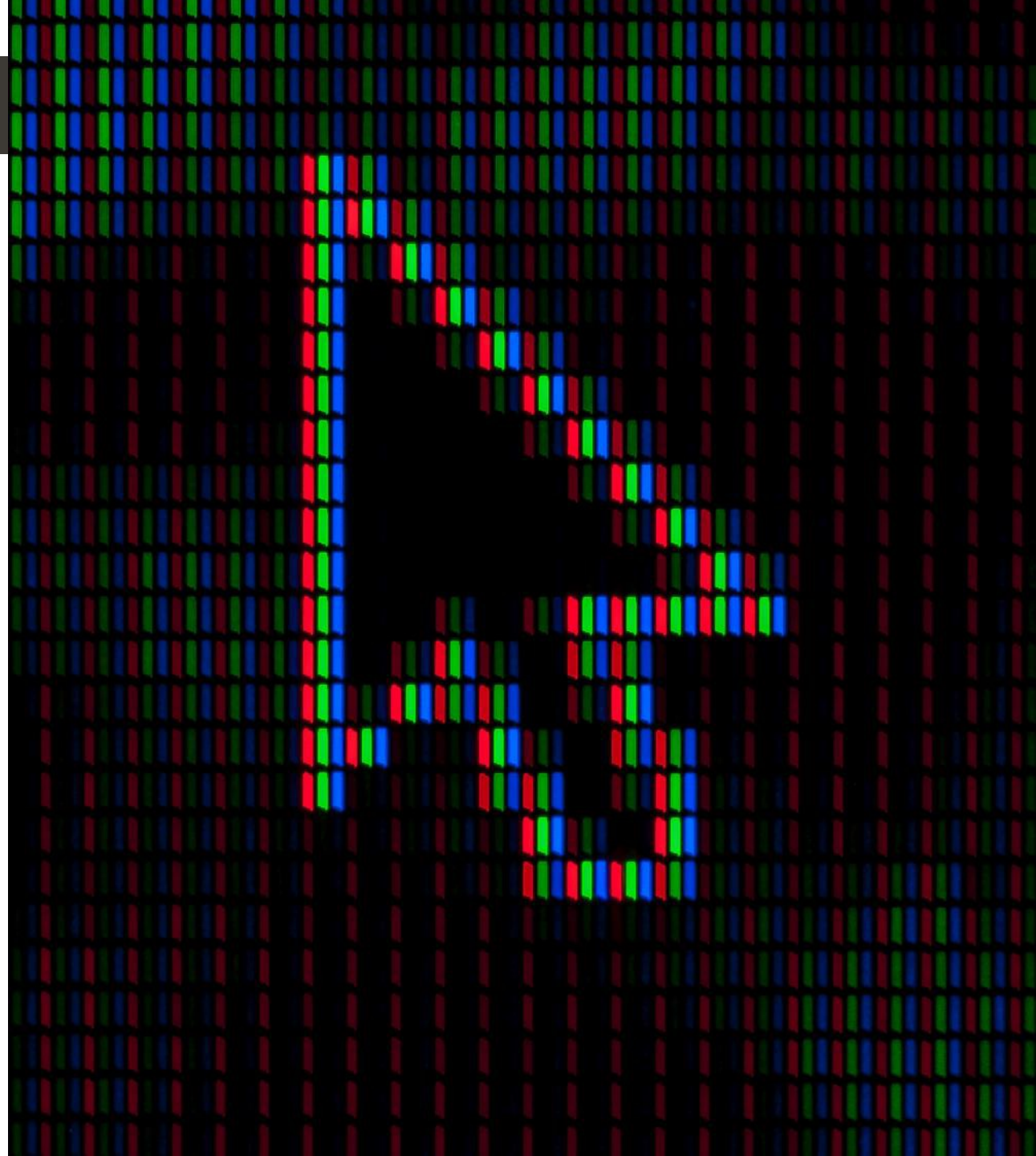


Part 1

출력 > 콘솔에서 출력하기



목차 a table of contents



기본 자료형

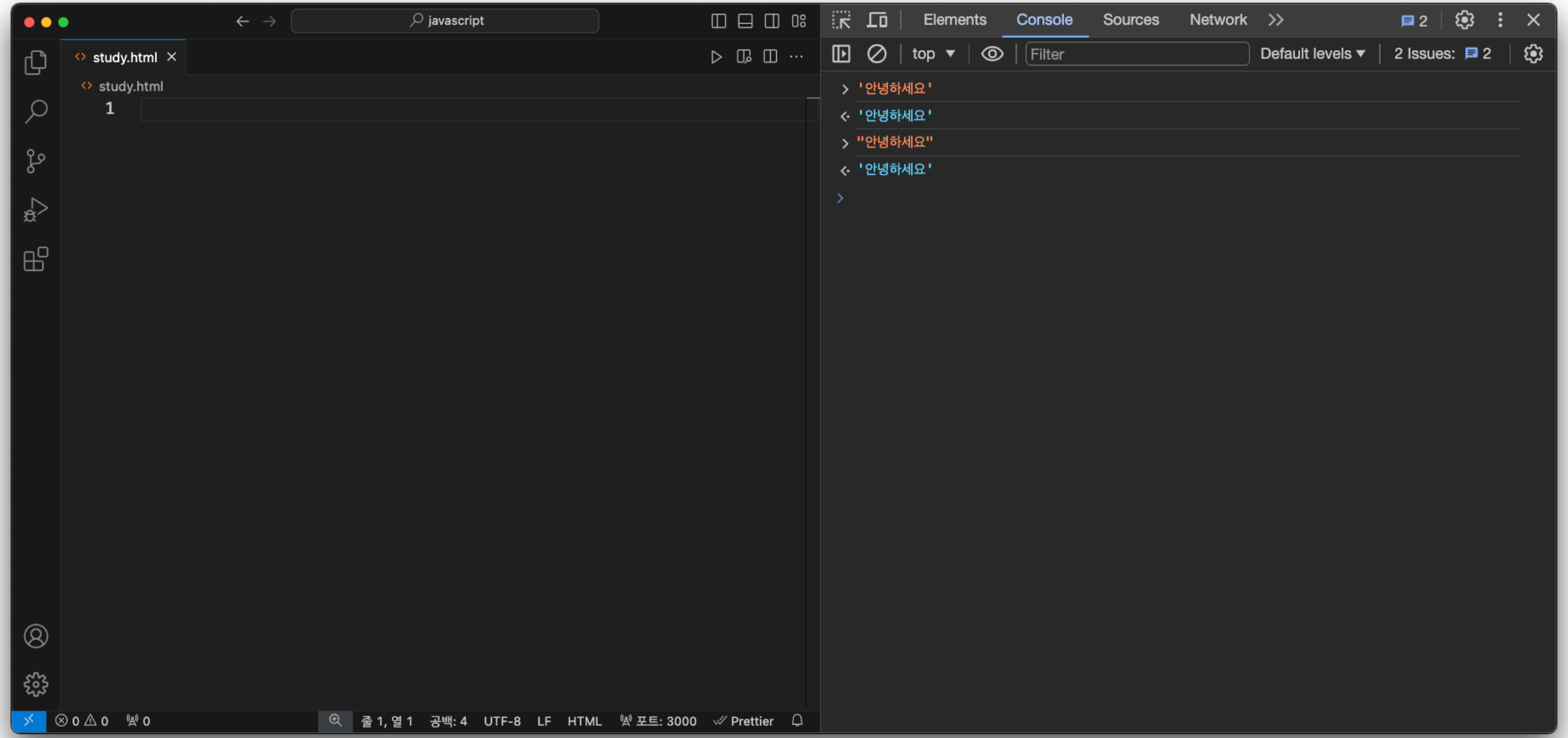
숫자
(number)

문자열
(string)

불
(boolean)

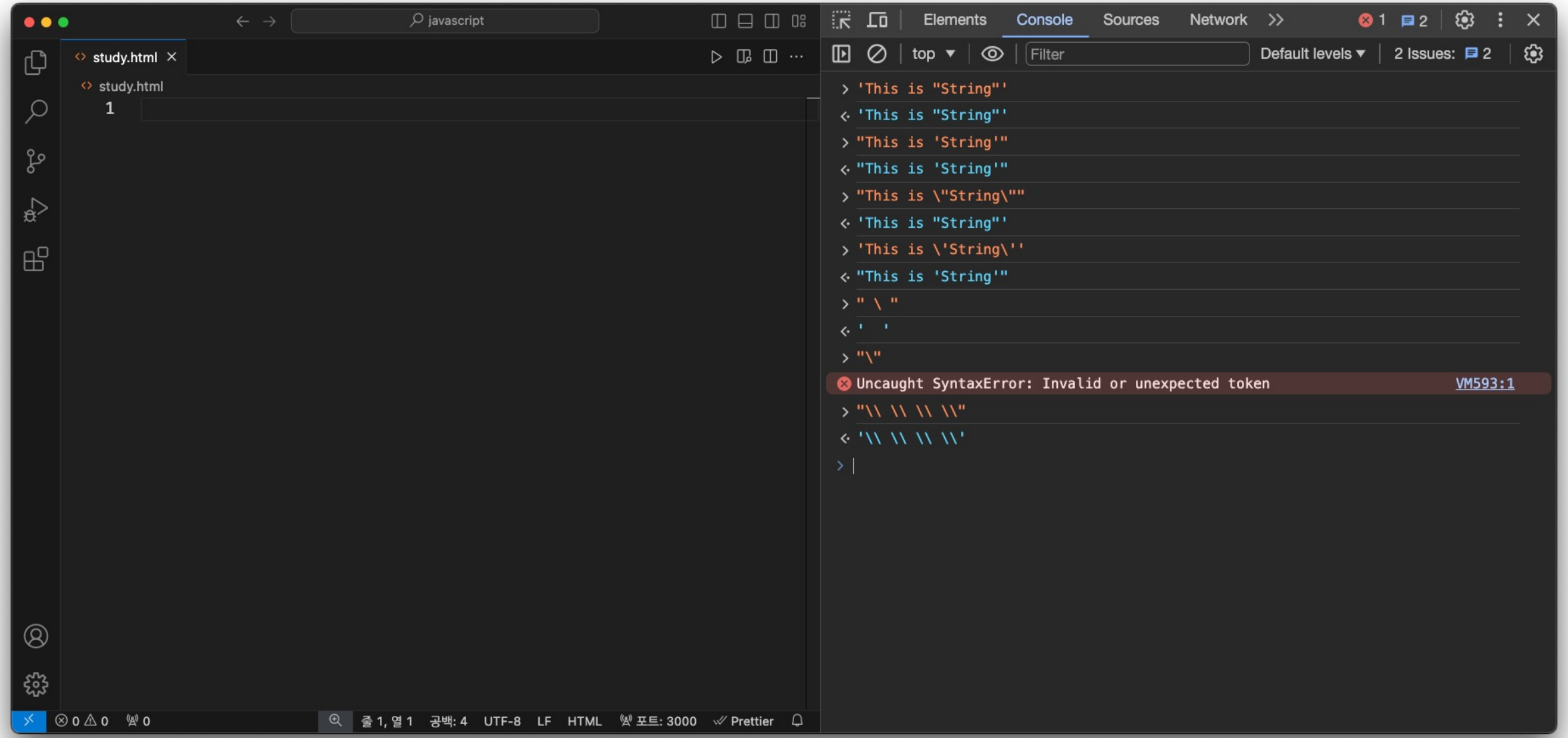
Part 2

문자열 자료형 > 만들기

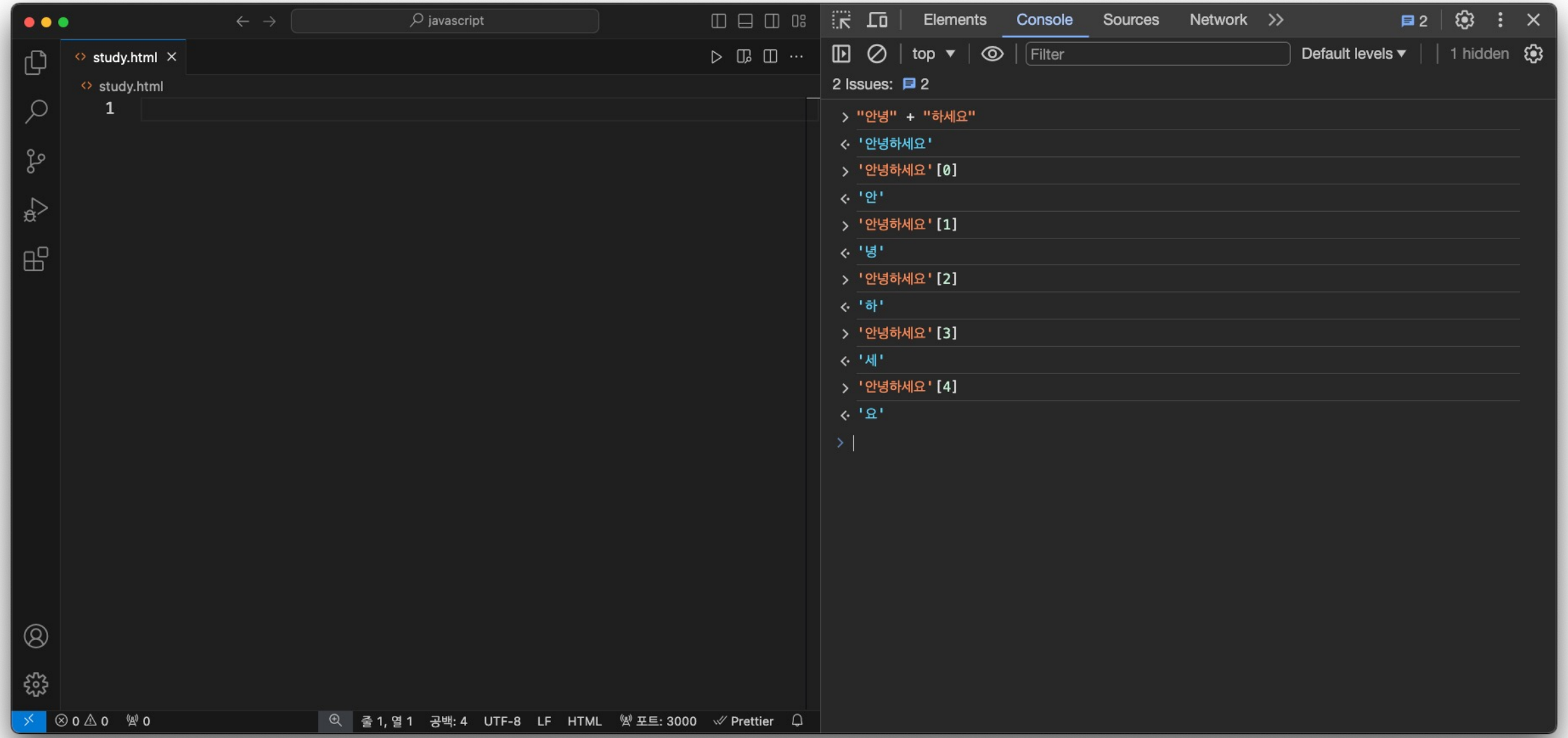


Part 2

문자열 자료형 > 만들기

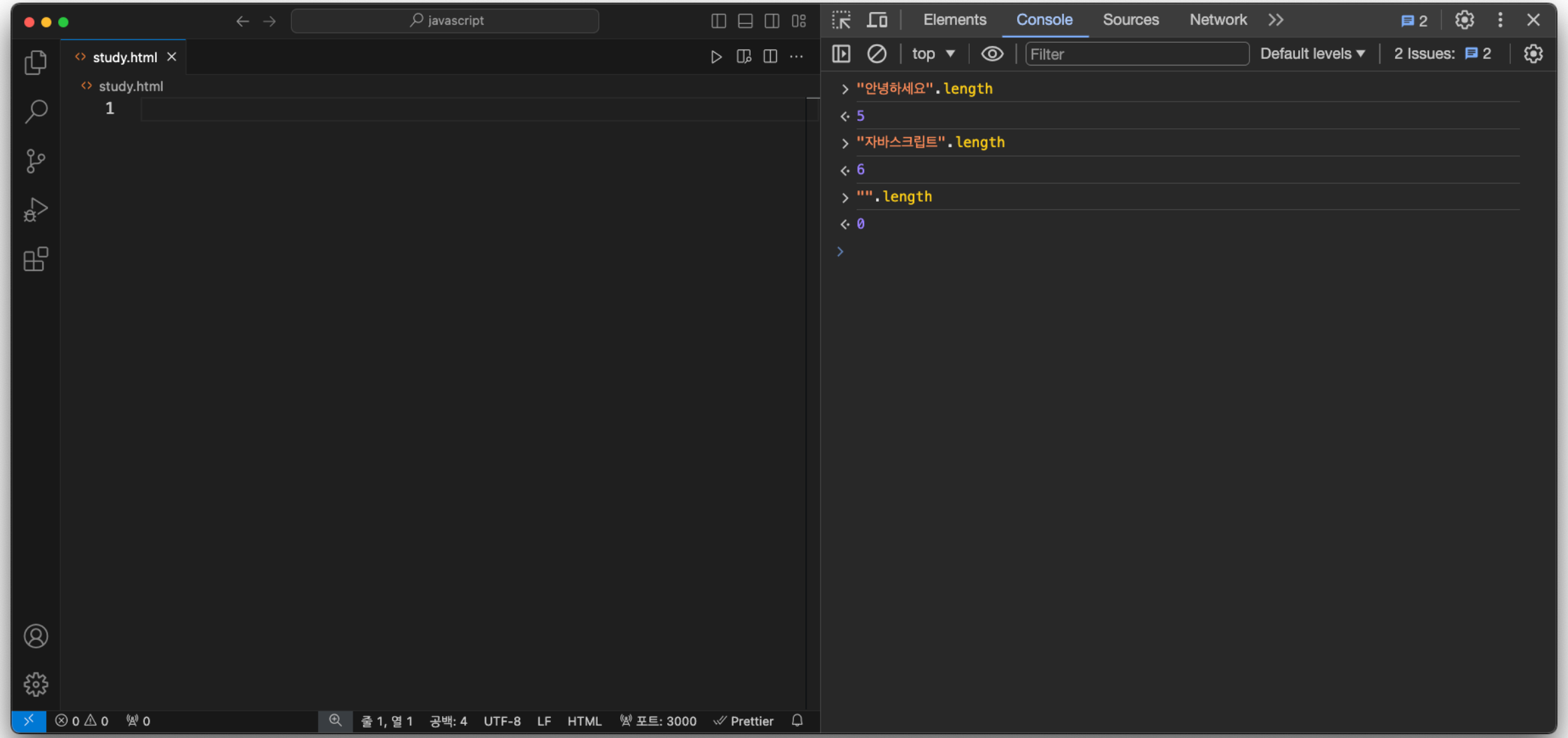


Part 2 문자열 자료형 > 연산



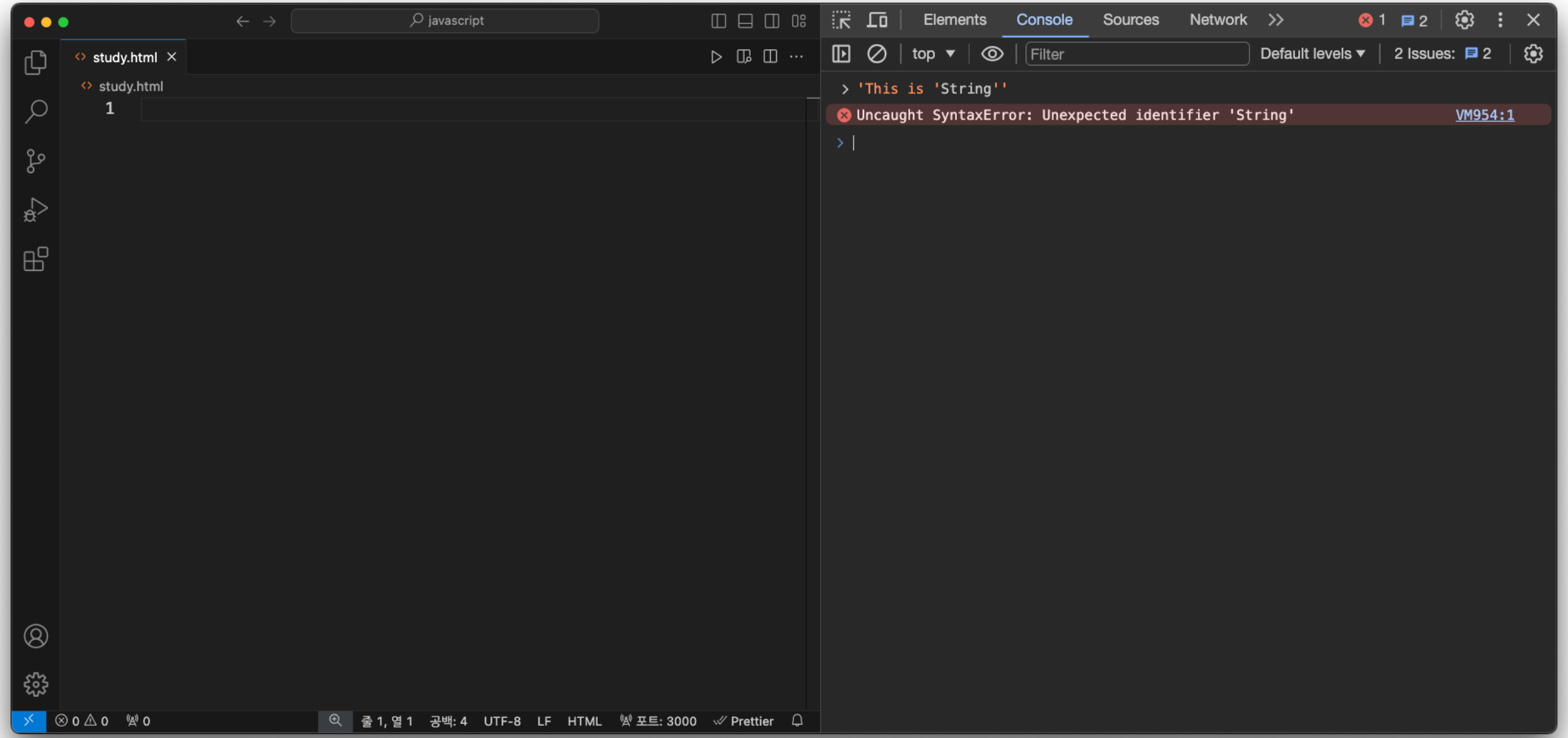
Part 2

문자열 자료형 > 길이 구하기



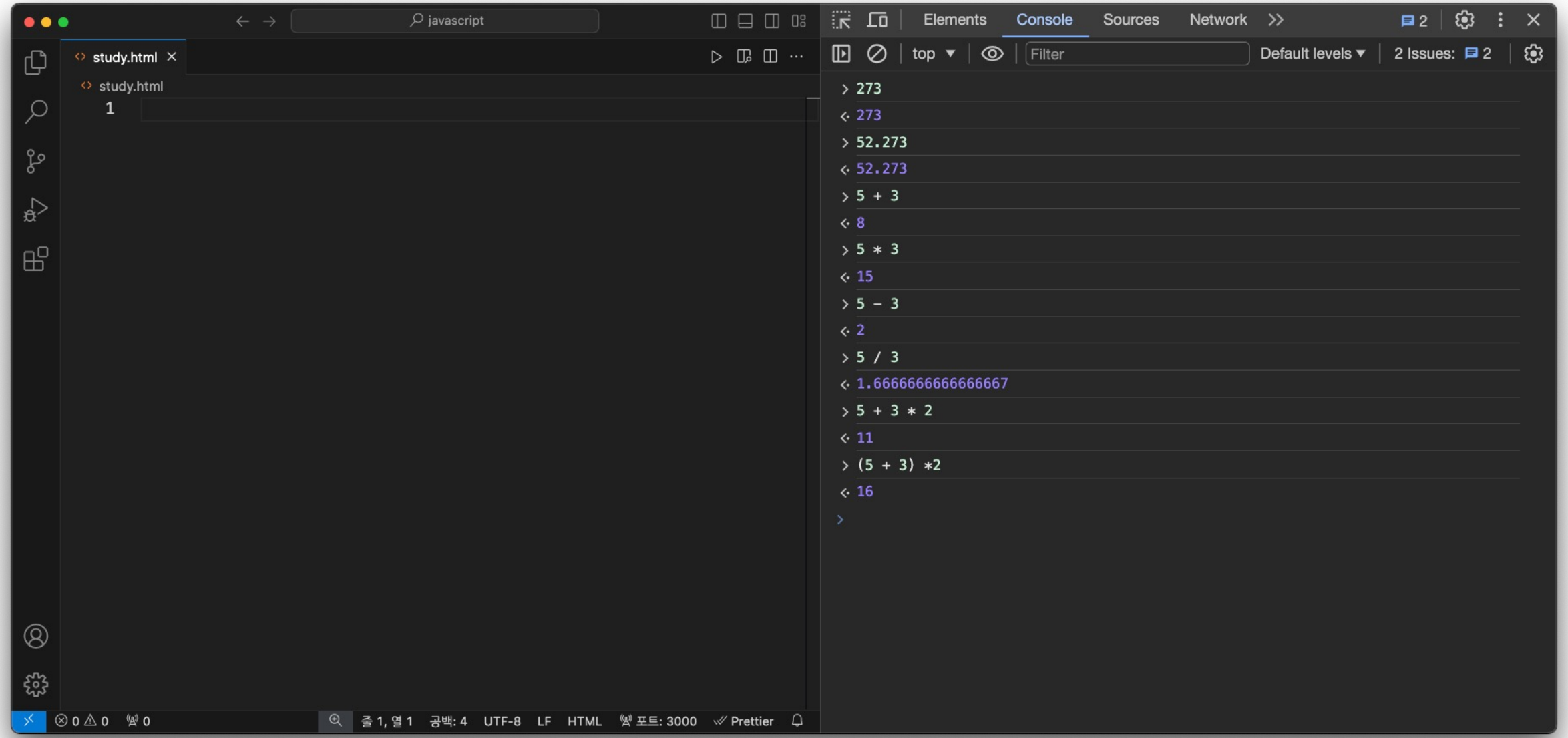
Part 2

문자열 자료형 > 오류

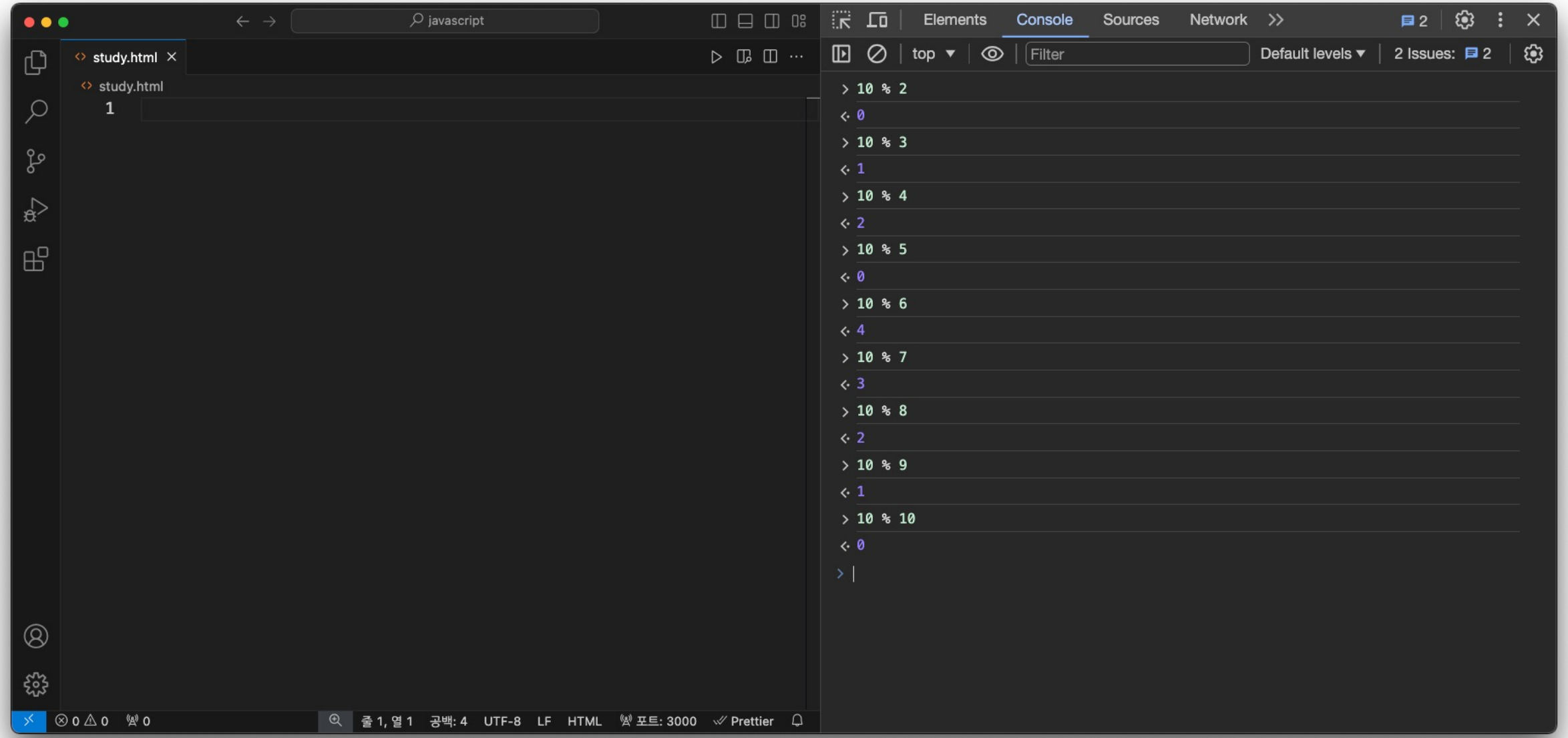


Part 2

숫자 자료형 > 연산자

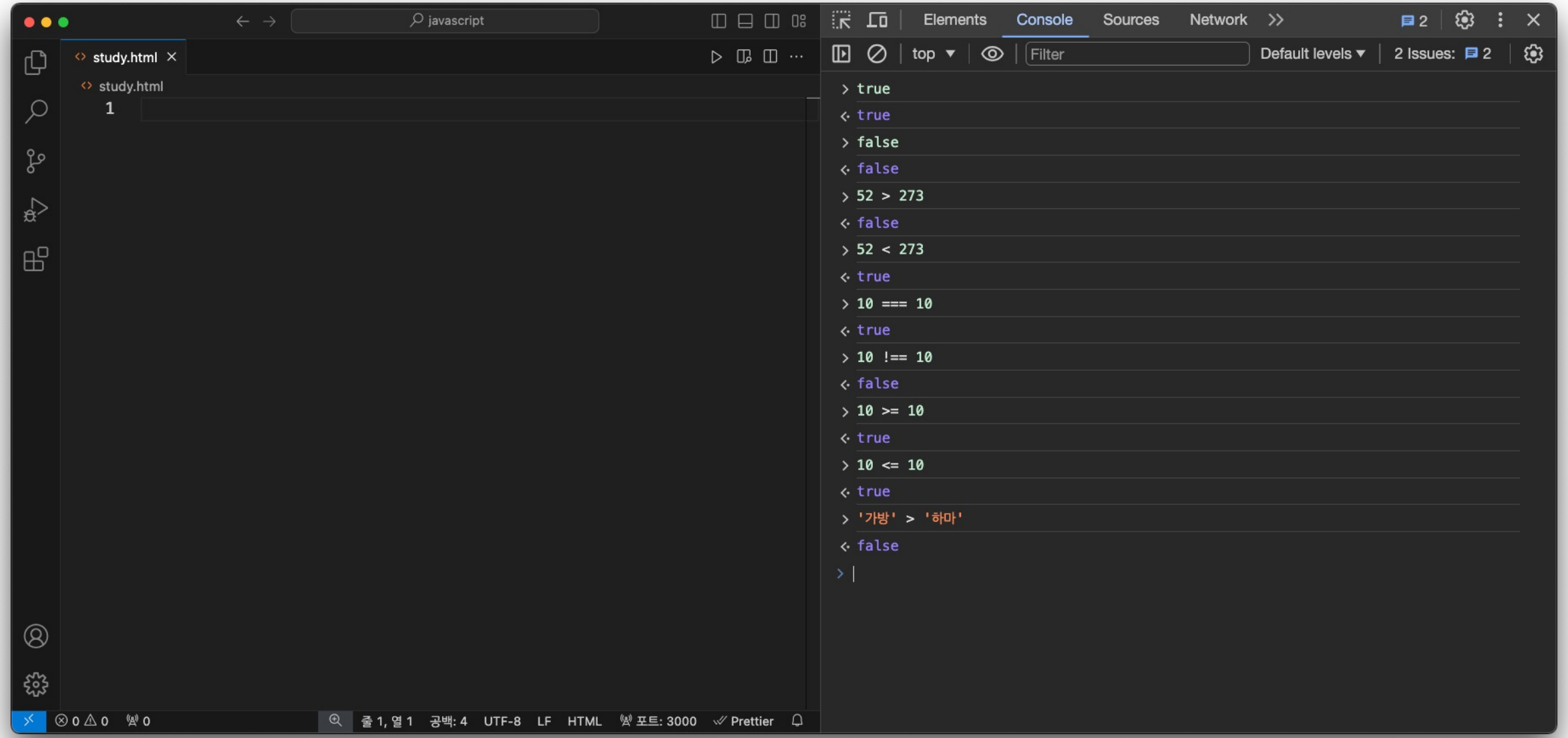


Part 2 숫자 자료형 > 연산자



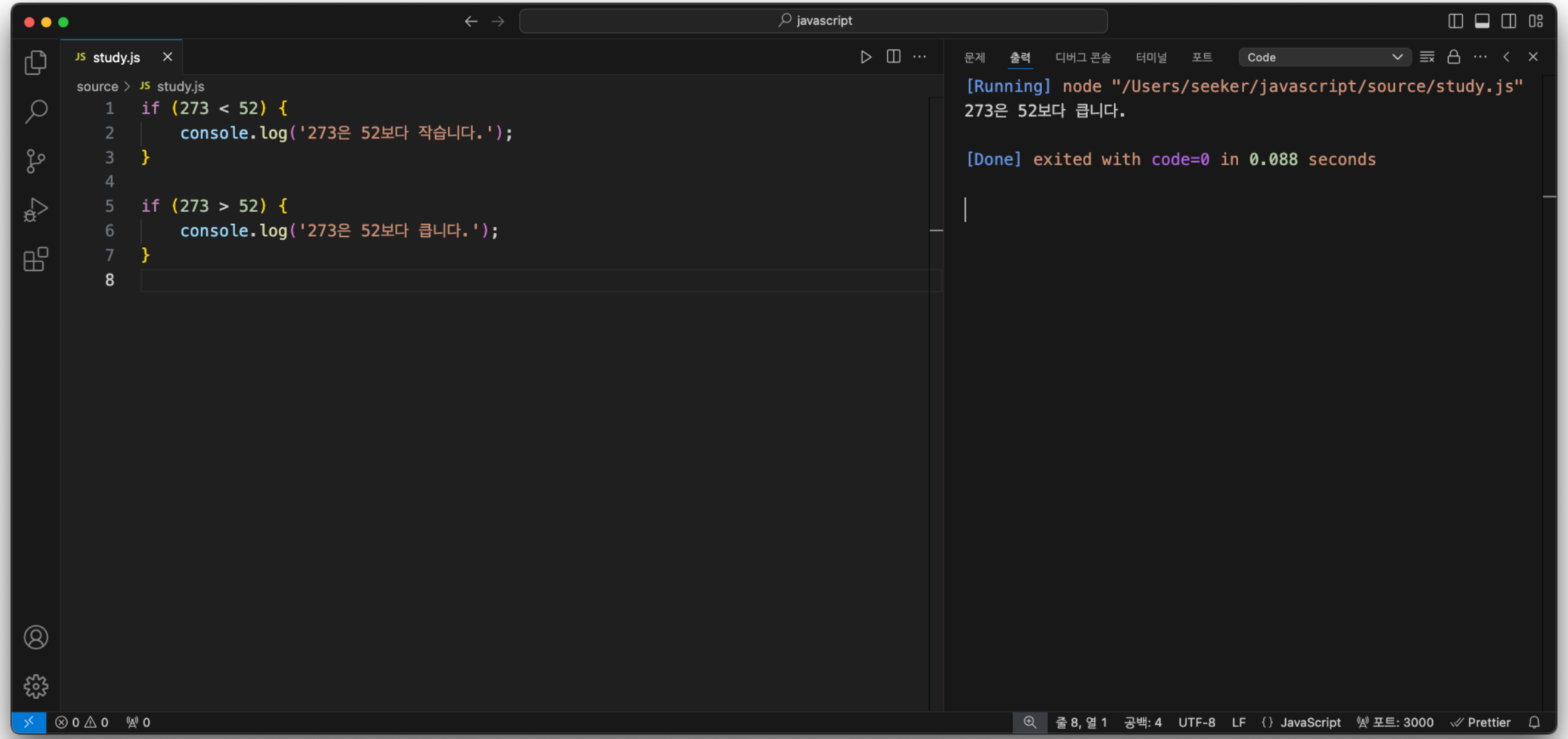
Part 2

불 자료형 > 만들기



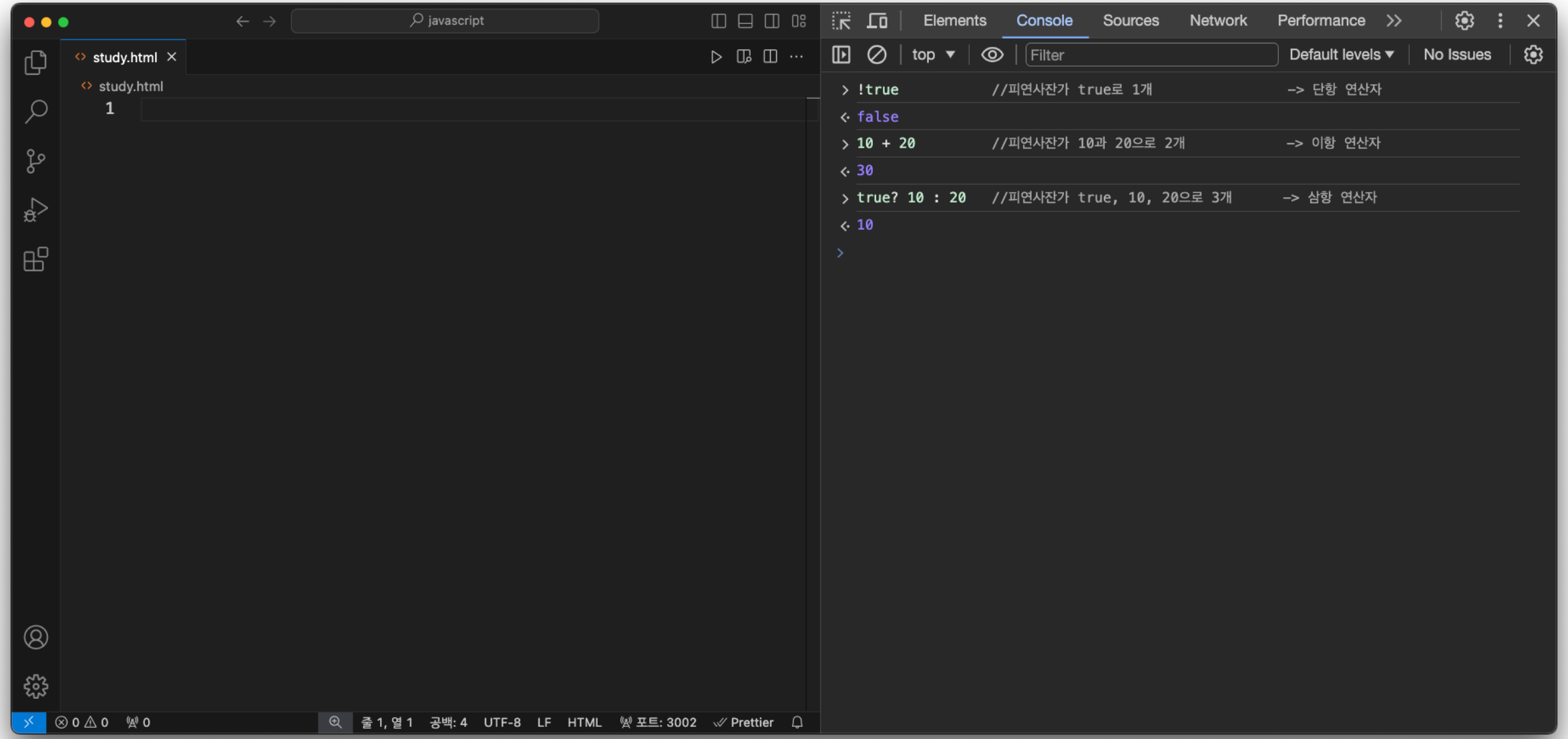
Part 2

불 자료형 > 만들기



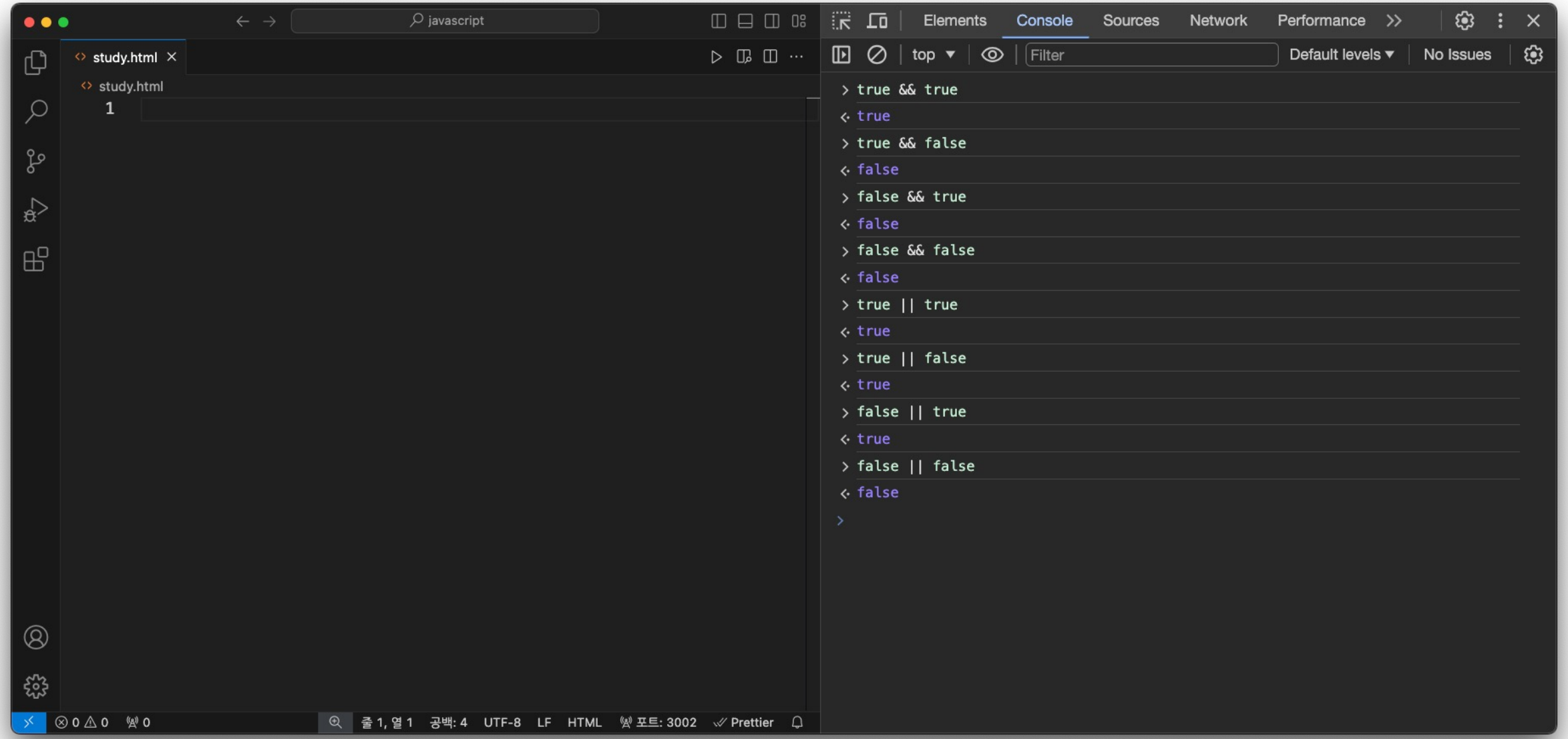
Part 2

불 자료형 > 부정 연산자



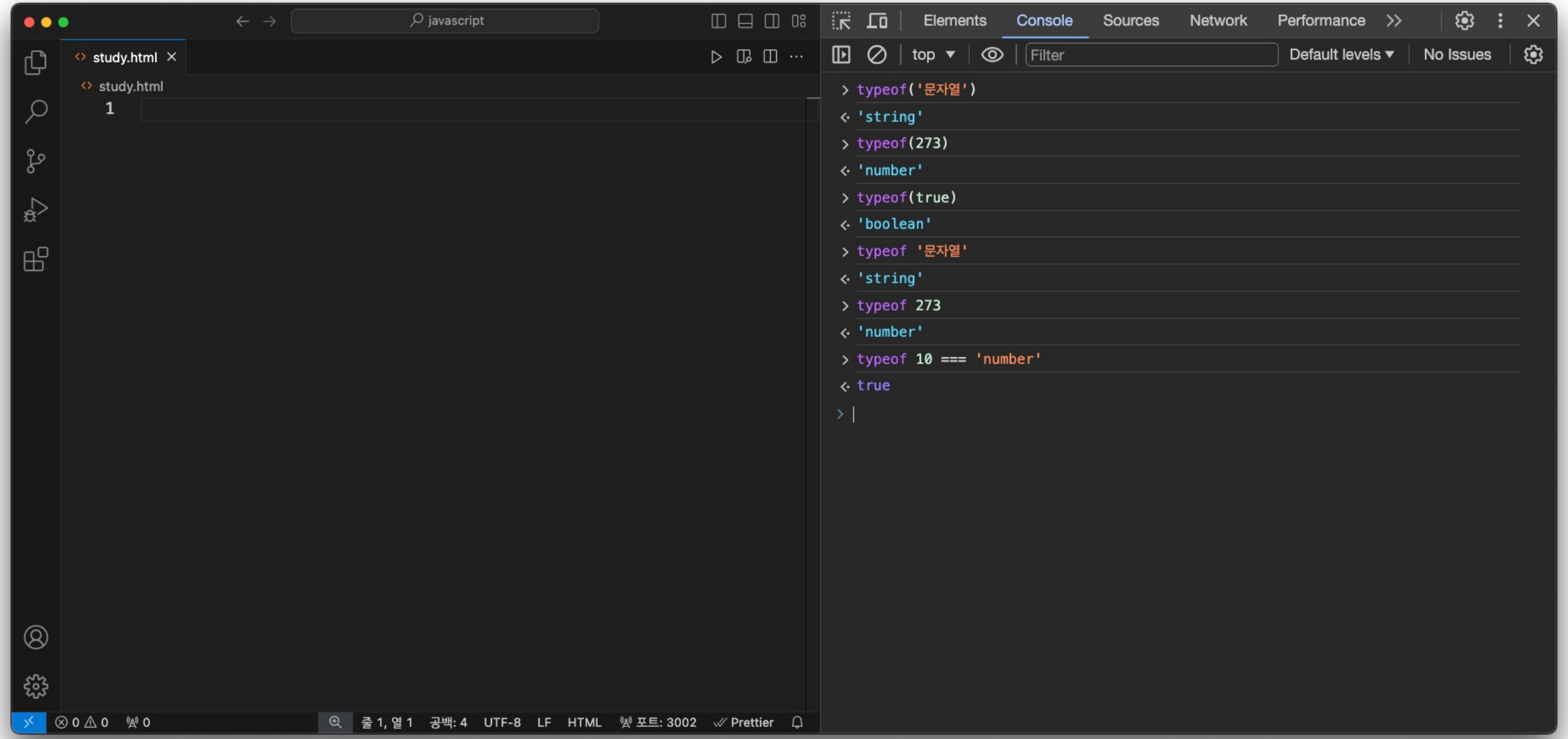
Part 2

불 자료형 > 논리합/논리곱 연산자



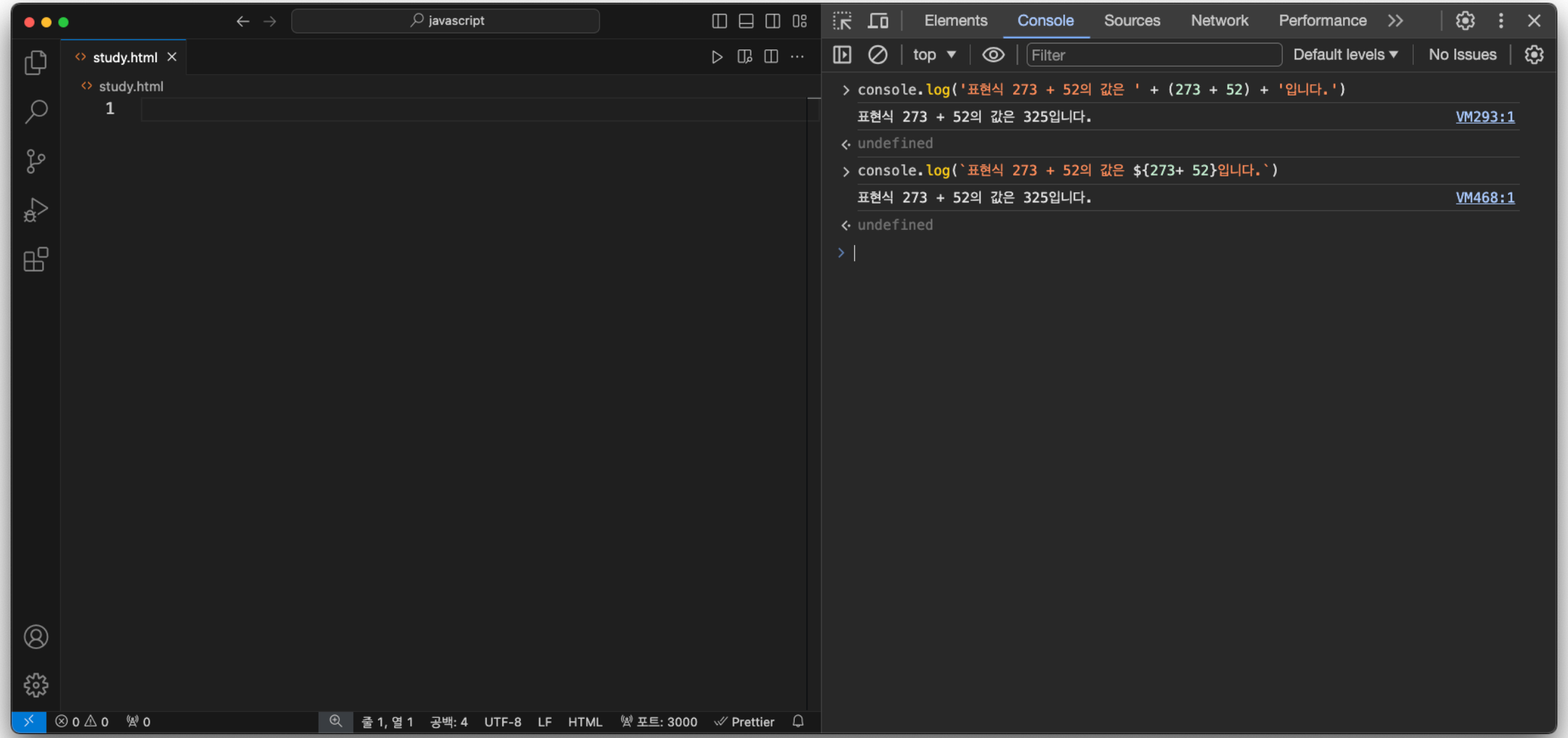
Part 2

자료형 검사


javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
> typeof('문자열')
'string'
> typeof(273)
'number'
> typeof(true)
'boolean'
> typeof '문자열'
'string'
> typeof 273
'number'
> typeof 10 === 'number'
true
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

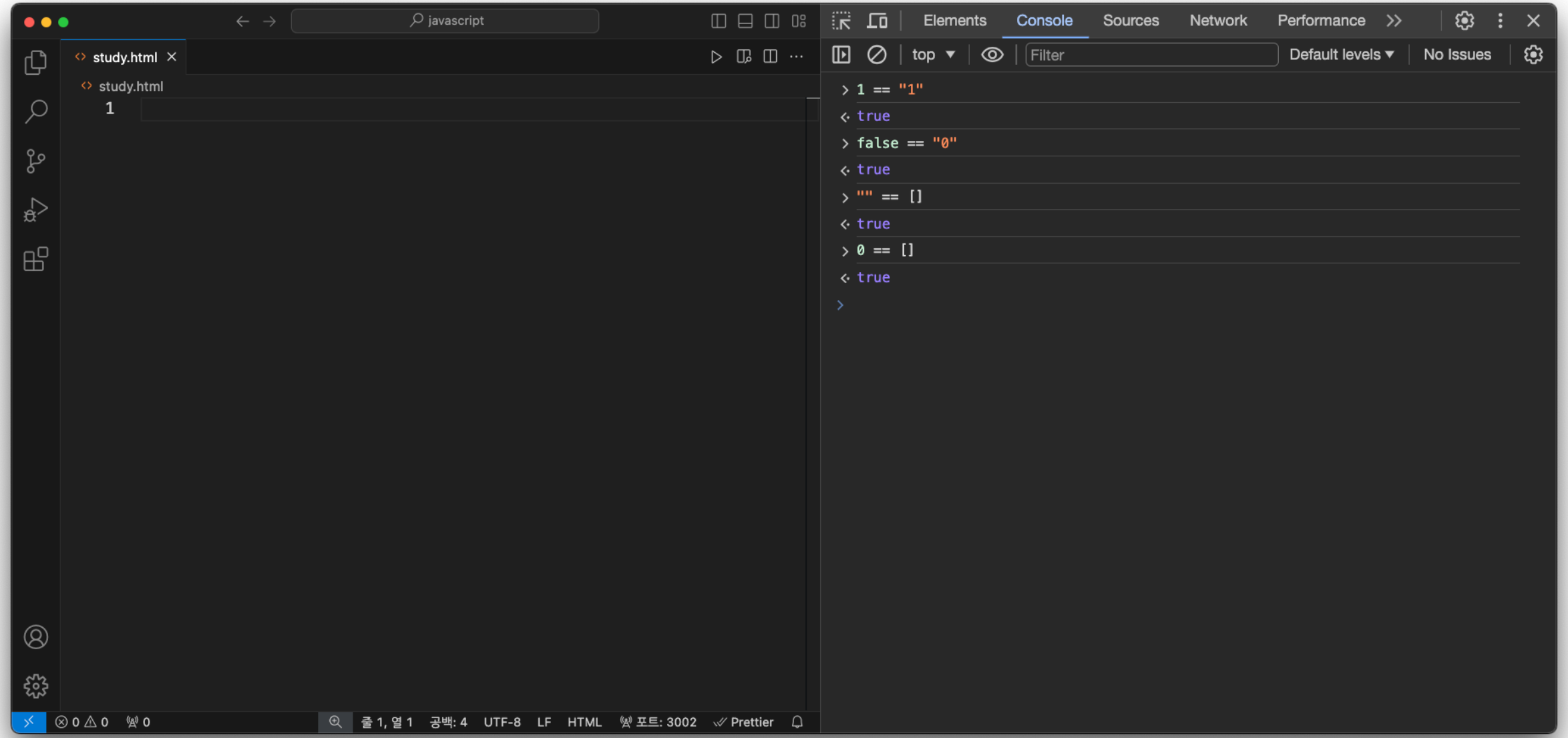
Part 2

템플릿 문자열



Part 2

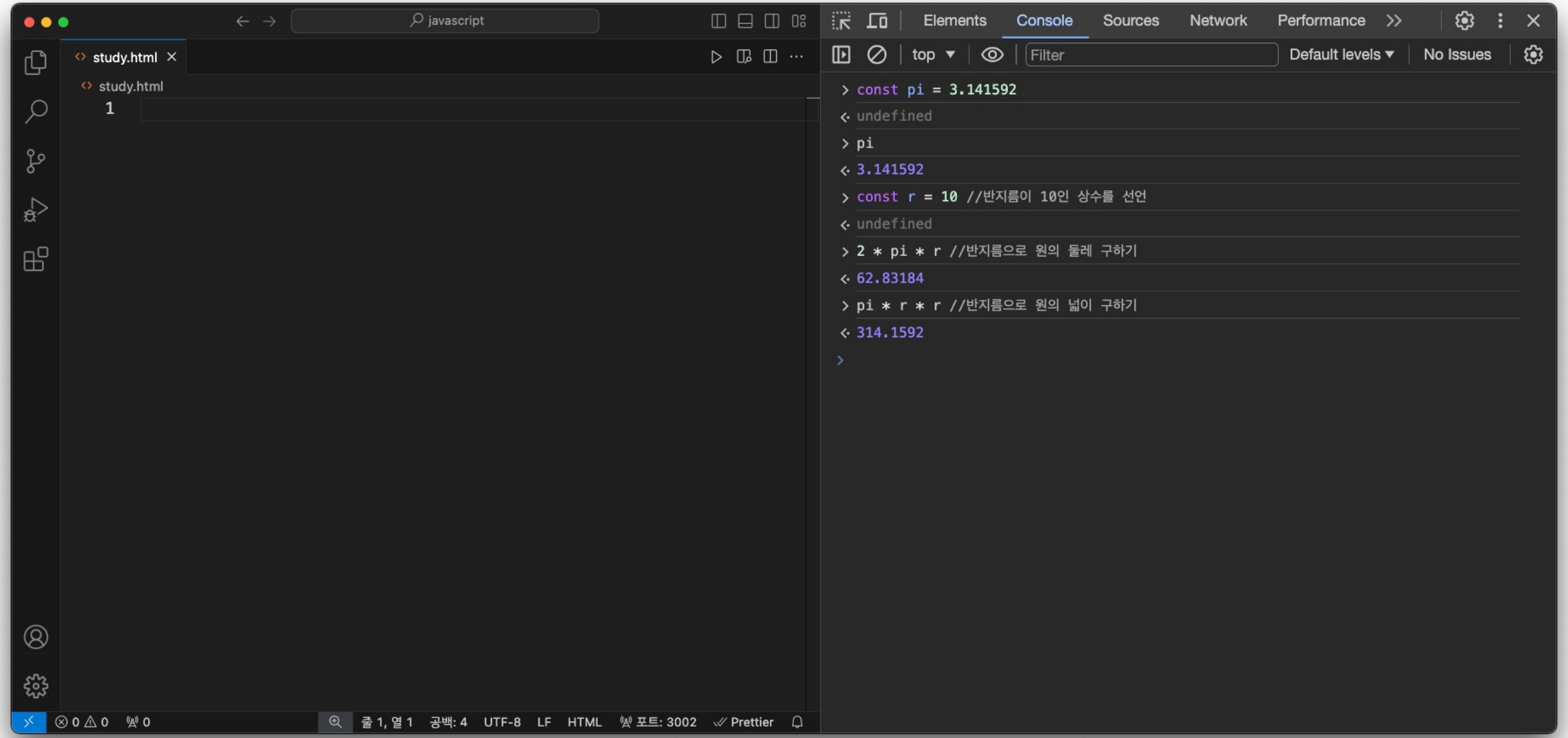
== 연산자와 != 연산자



상수와 변수

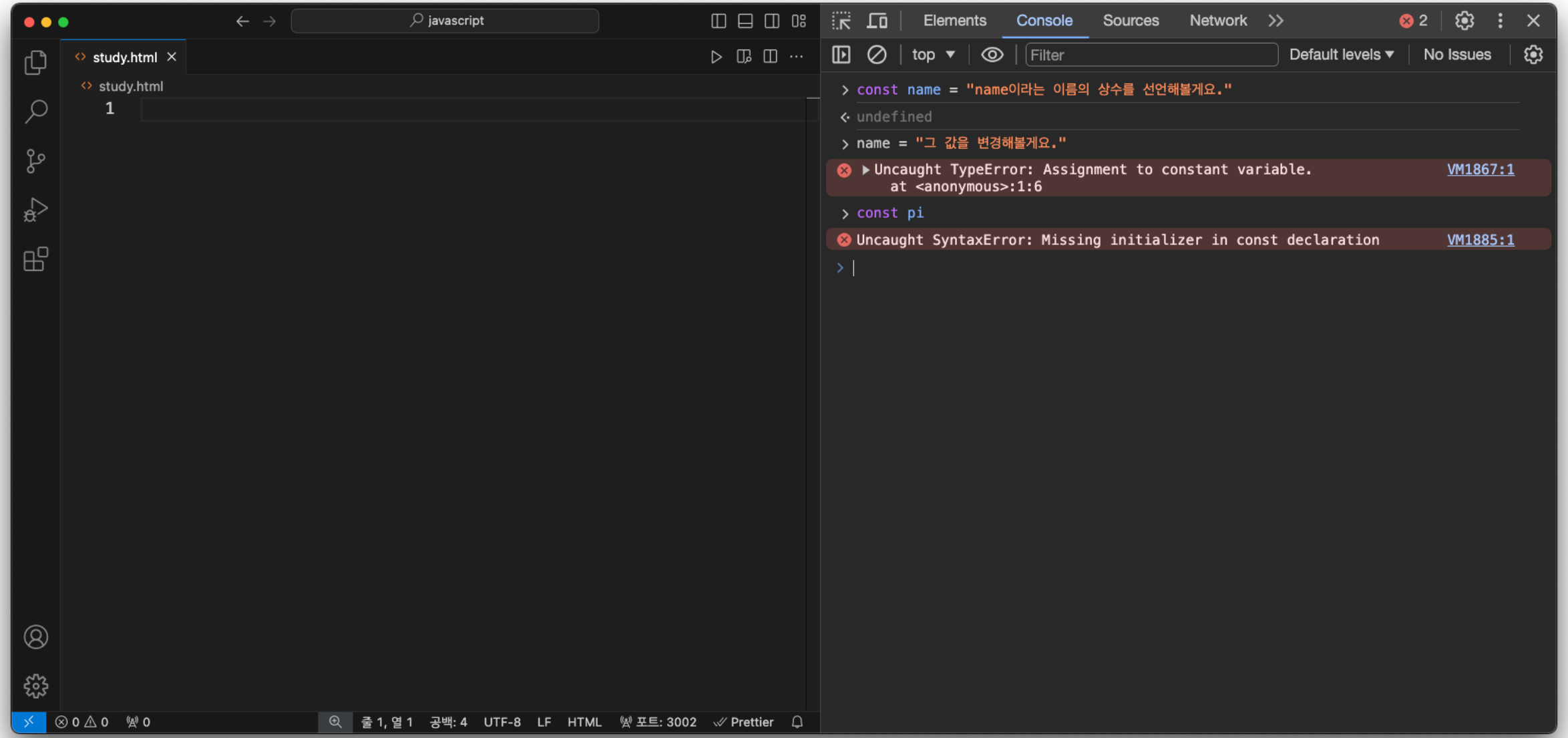
Part 2

상수


javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
> const pi = 3.141592
< undefined
> pi
< 3.141592
> const r = 10 //반지름이 10인 상수를 선언
< undefined
> 2 * pi * r //반지름으로 원의 둘레 구하기
< 62.83184
> pi * r * r //반지름으로 원의 넓이 구하기
< 314.1592
>
⊗ 0 ⚠ 0
0
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

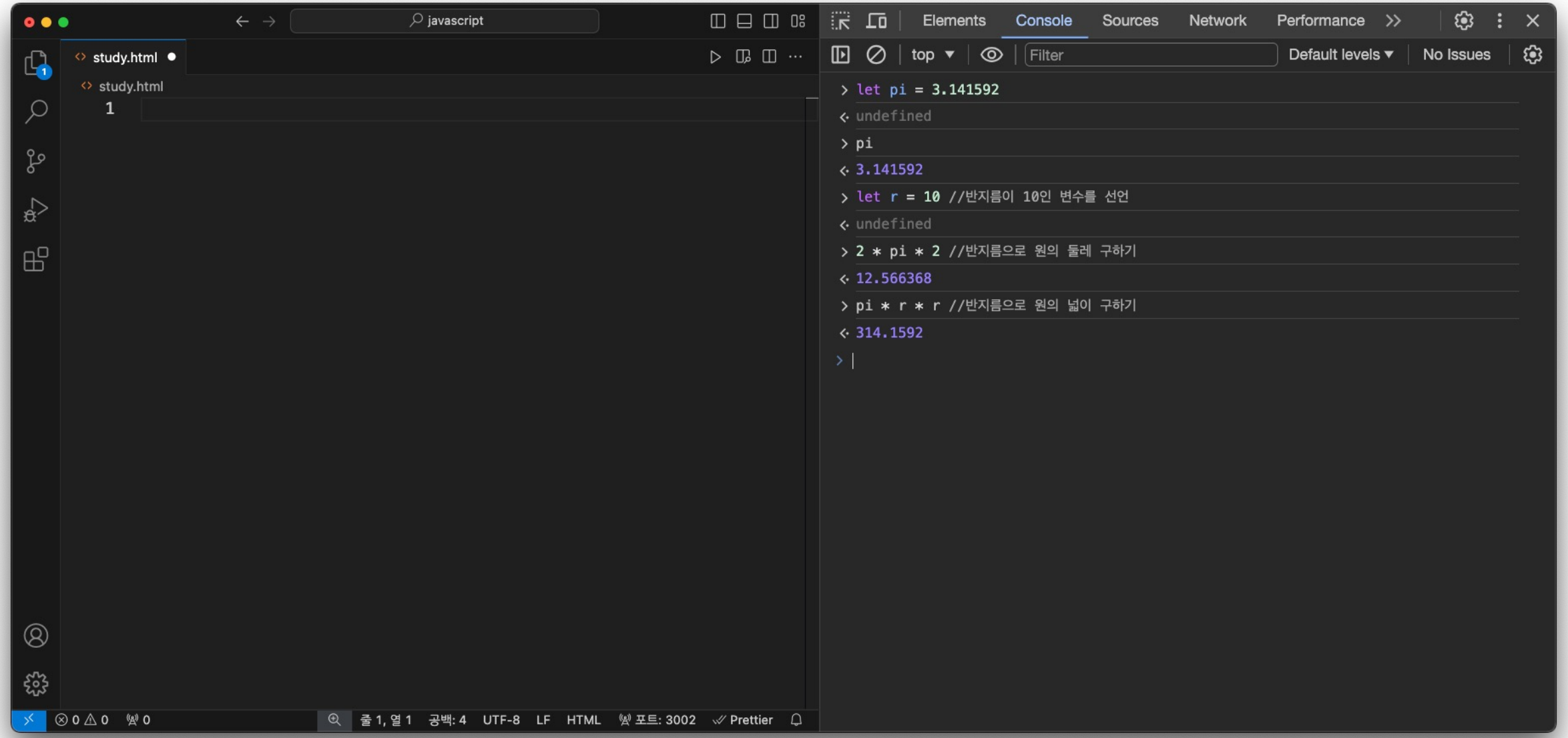
Part 2

상수 > 오류


javascript
study.html
study.html
1
Elements
Console
Sources
Network
2
top
Filter
Default levels
No Issues
const name = "name이라는 이름의 상수를 선언해볼게요."
undefined
name = "그 값을 변경해볼게요."
Uncaught TypeError: Assignment to constant variable.
at <anonymous>:1:6
VM1867:1
const pi
Uncaught SyntaxError: Missing initializer in const declaration
VM1885:1
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

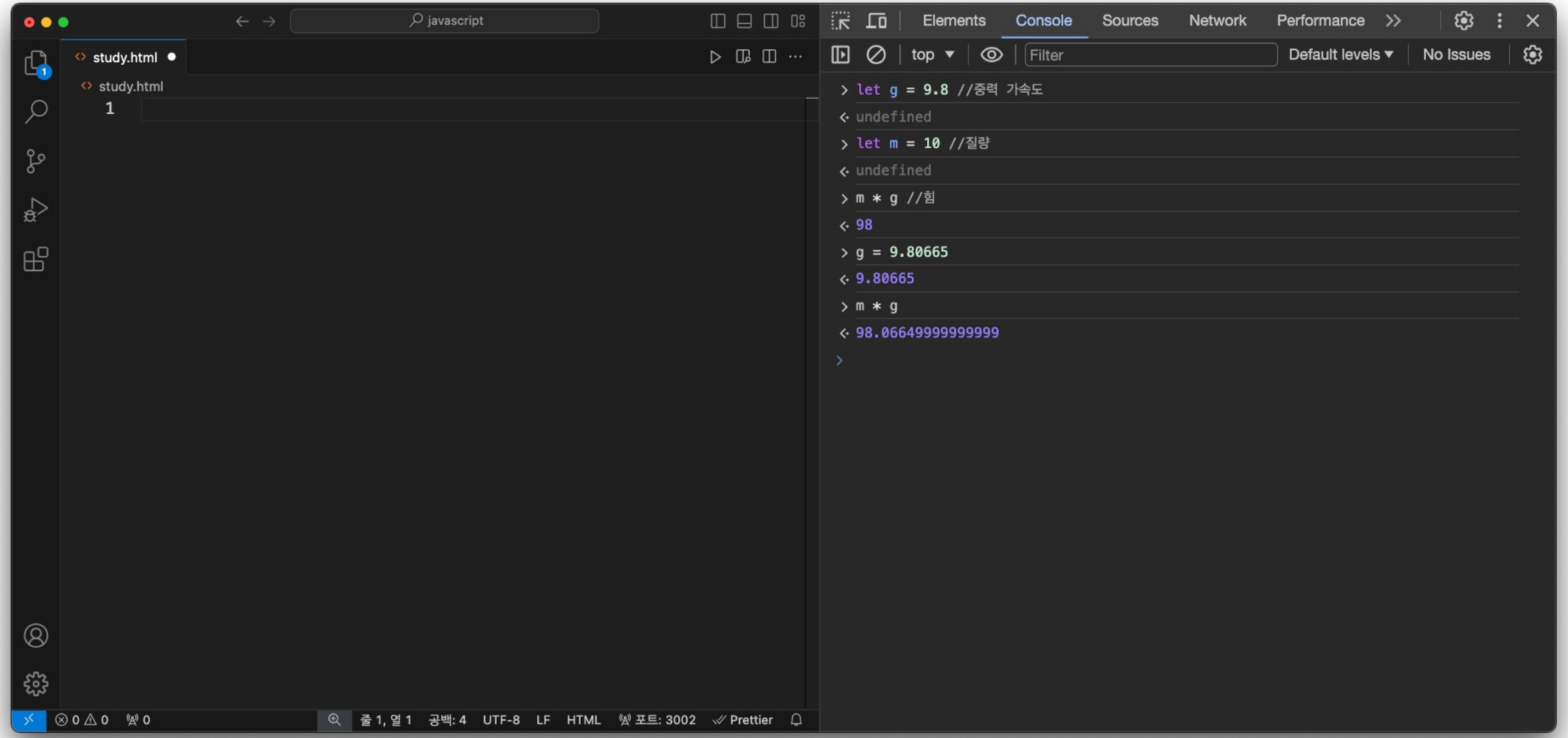
Part 2

변수


javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
> let pi = 3.141592
< undefined
> pi
< 3.141592
> let r = 10 //반지름이 10인 변수를 선언
< undefined
> 2 * pi * 2 //반지름으로 원의 둘레 구하기
< 12.566368
> pi * r * r //반지름으로 원의 넓이 구하기
< 314.1592
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

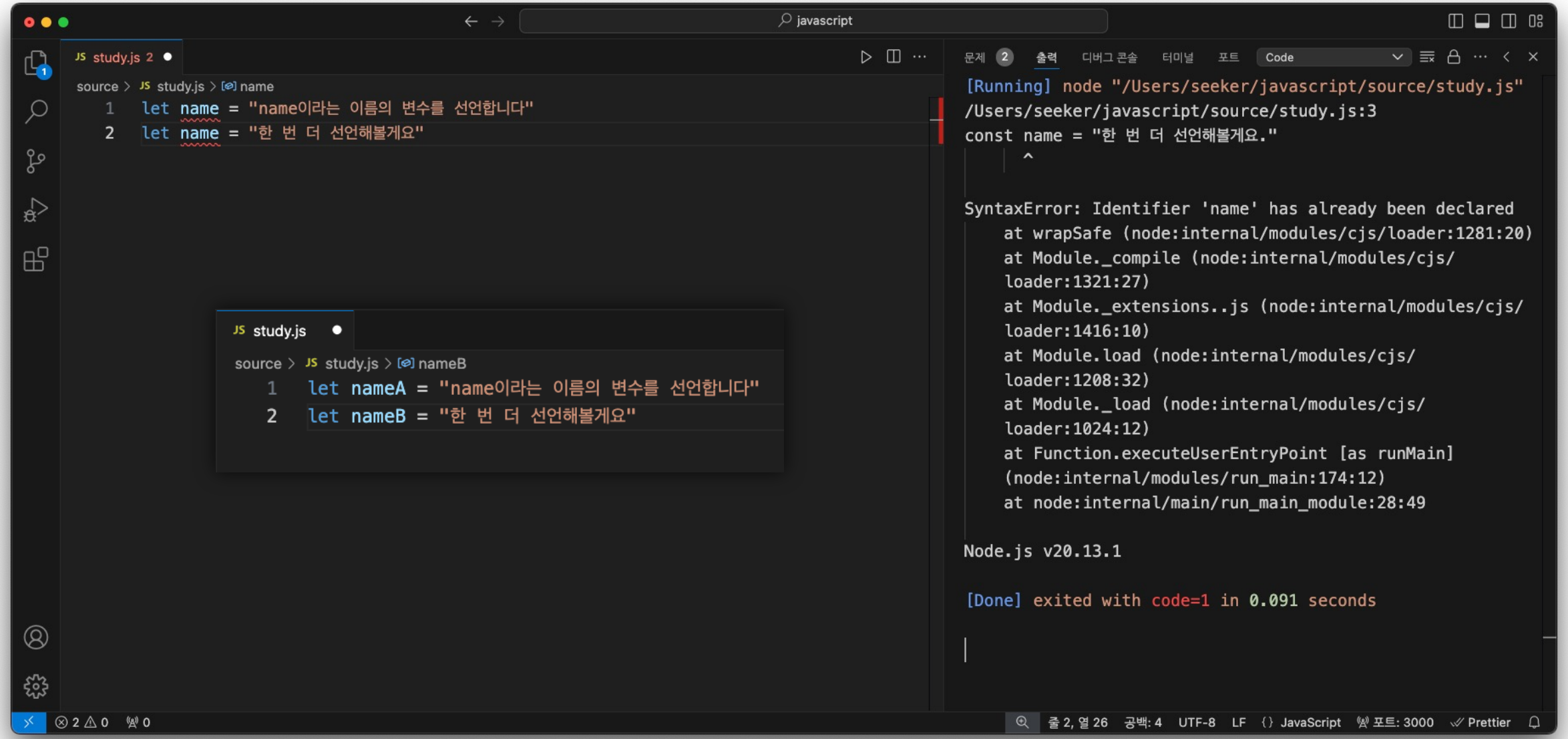
Part 2

변수 > 변경


javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
> let g = 9.8 //중력 가속도
undefined
> let m = 10 //질량
undefined
> m * g //힘
98
> g = 9.80665
9.80665
> m * g
98.06649999999999
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

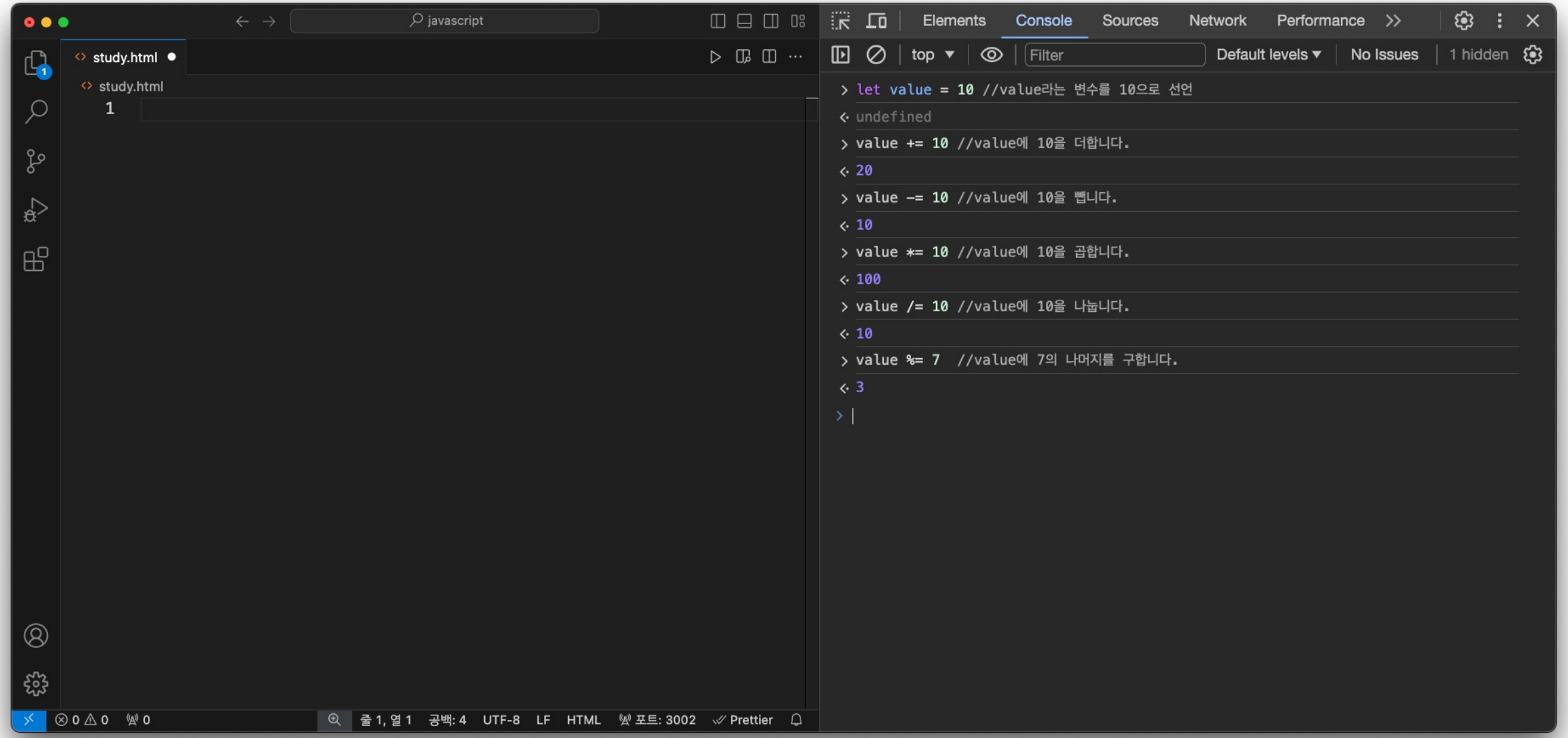
Part 2

변수 > 오류


javascript
study.js 2
source > study.js > name
1 let name = "name이라는 이름의 변수를 선언합니다"
2 let name = "한 번 더 선언해볼게요"
study.js
source > study.js > nameB
1 let nameA = "name이라는 이름의 변수를 선언합니다"
2 let nameB = "한 번 더 선언해볼게요"
문제 2
출력
디버그 콘솔
터미널
포트
Code
[Running] node "/Users/seeker/javascript/source/study.js"
/Users/seeker/javascript/source/study.js:3
const name = "한 번 더 선언해볼게요."
^
SyntaxError: Identifier 'name' has already been declared
at wrapSafe (node:internal/modules/cjs/loader:1281:20)
at Module._compile (node:internal/modules/cjs/loader:1321:27)
at Module._extensions..js (node:internal/modules/cjs/loader:1416:10)
at Module.load (node:internal/modules/cjs/loader:1208:32)
at Module._load (node:internal/modules/cjs/loader:1024:12)
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:174:12)
at node:internal/main/run_main_module:28:49
Node.js v20.13.1
[Done] exited with code=1 in 0.091 seconds
줄 2, 열 26
공백: 4
UTF-8
LF
JavaScript
포트: 3000
Prettier

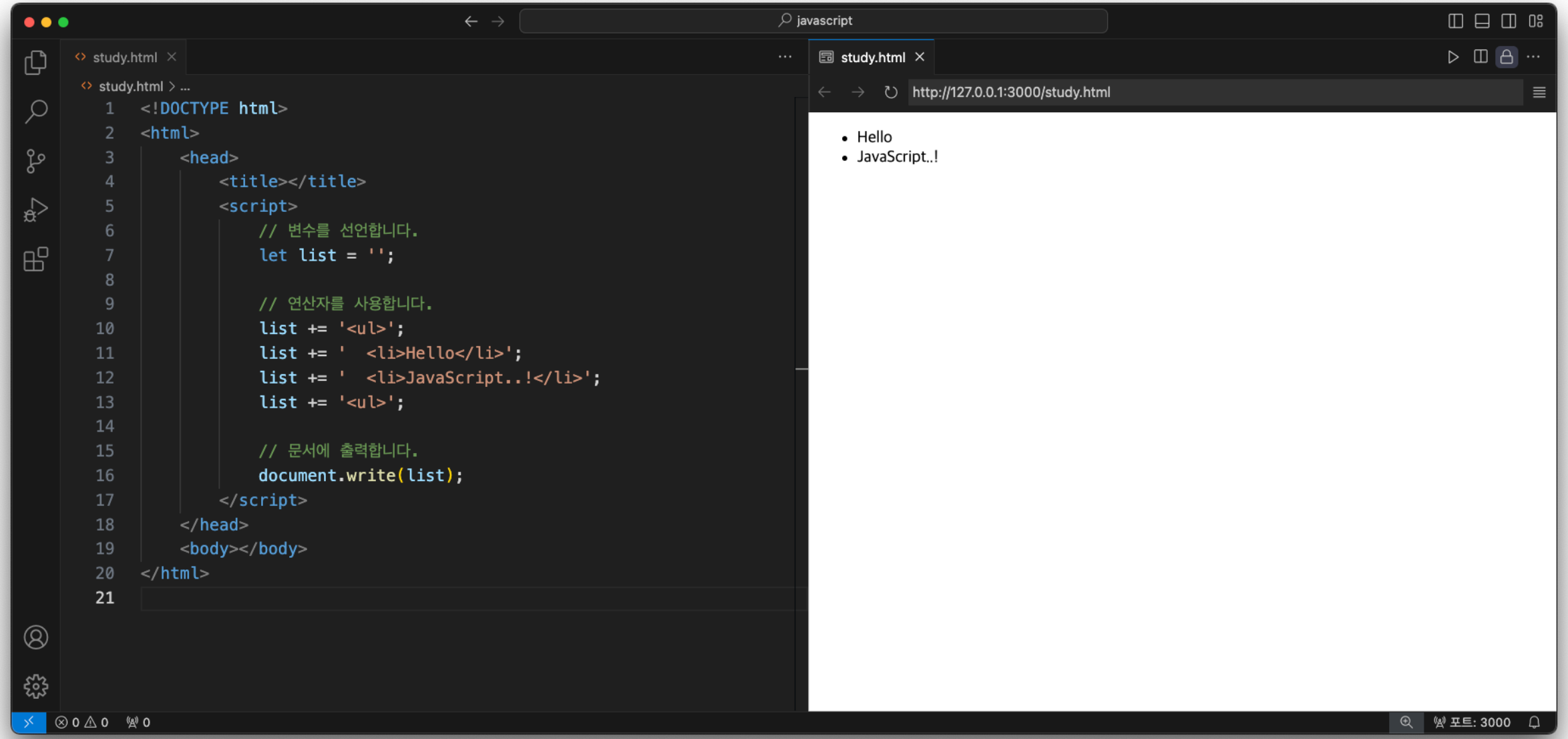
Part 2

변수에 적용할 수 있는 연산자 > 복합 대입 연산자



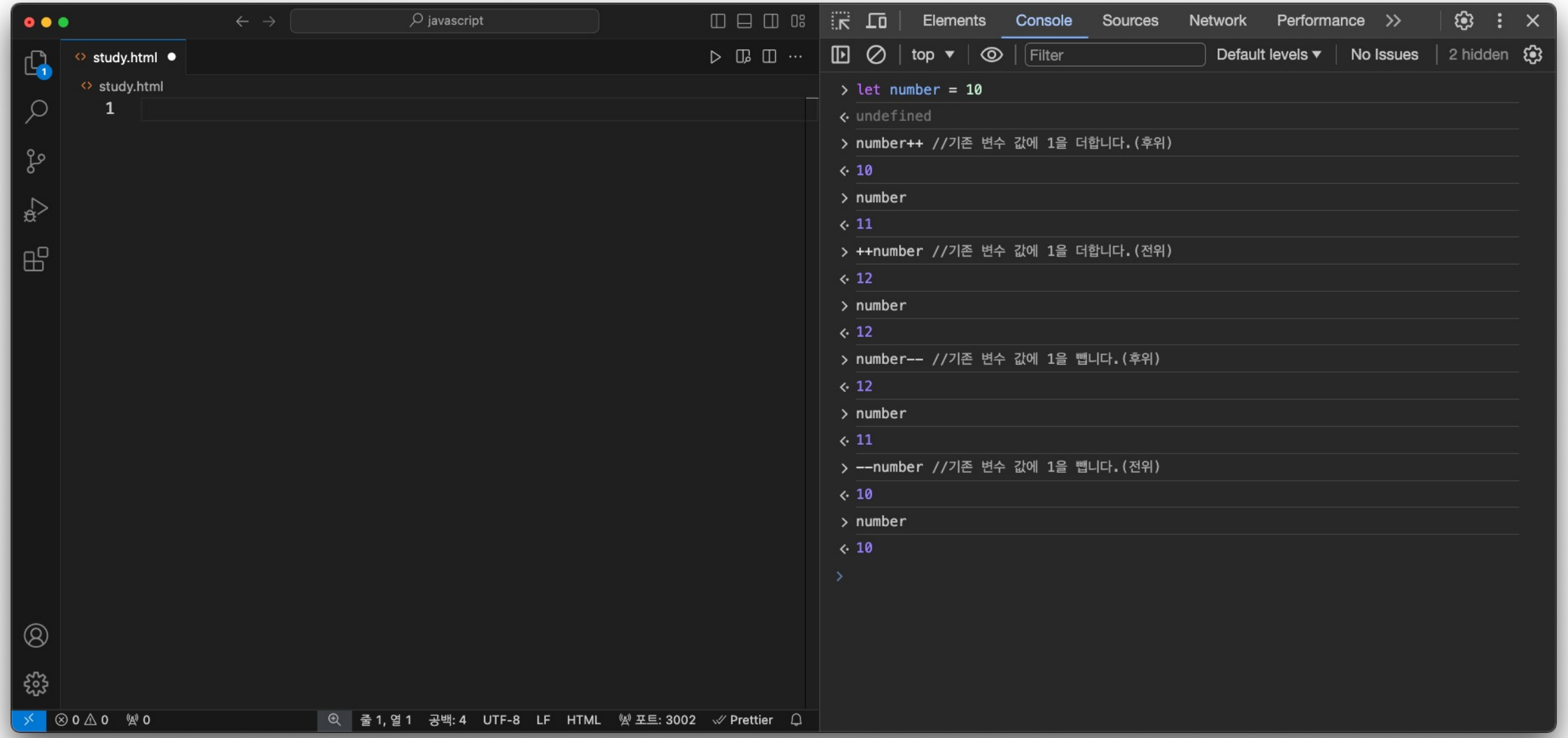
Part 2

변수에 적용할 수 있는 연산자 > 복합 대입 연산자



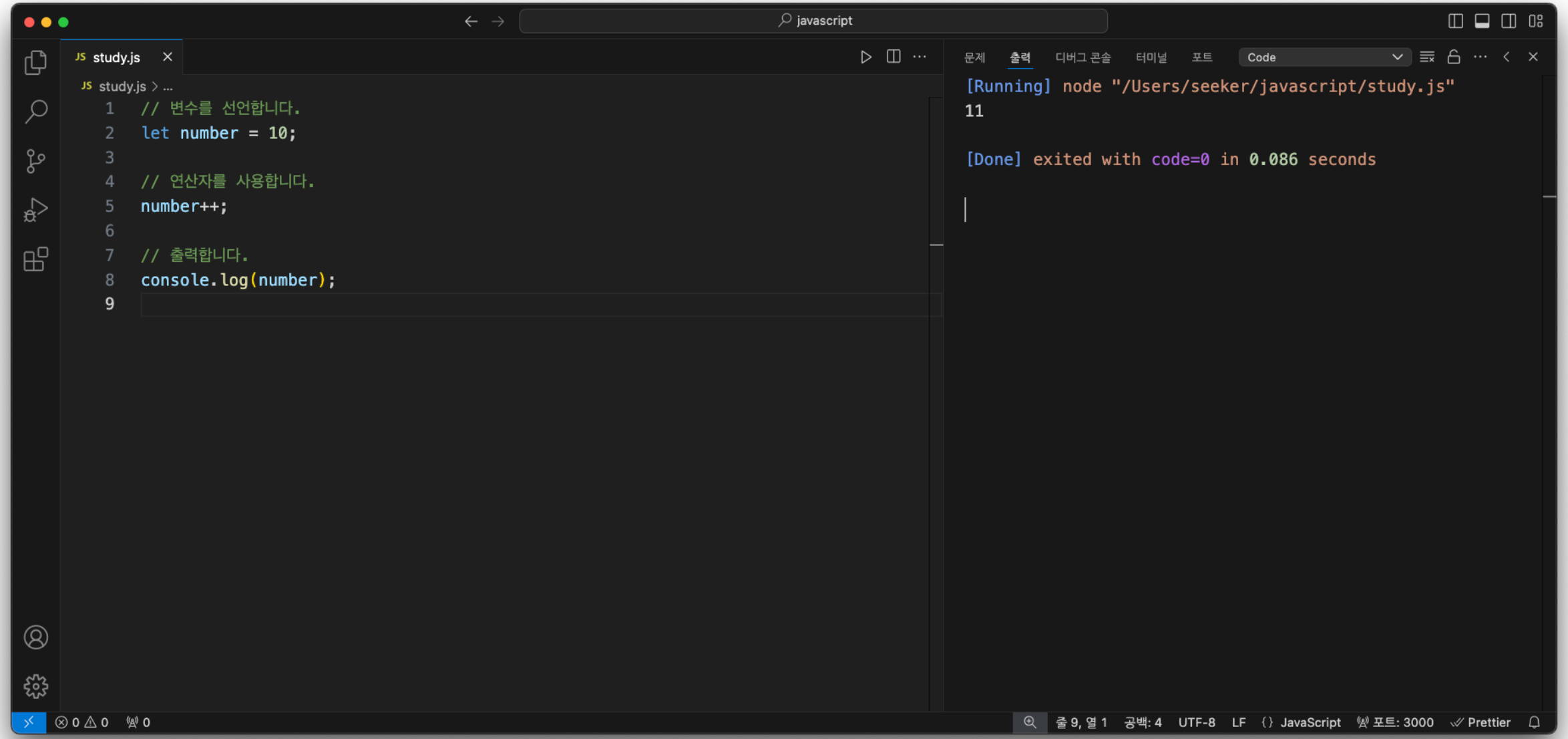
Part 2

변수에 적용할 수 있는 연산자 > 증감 연산자



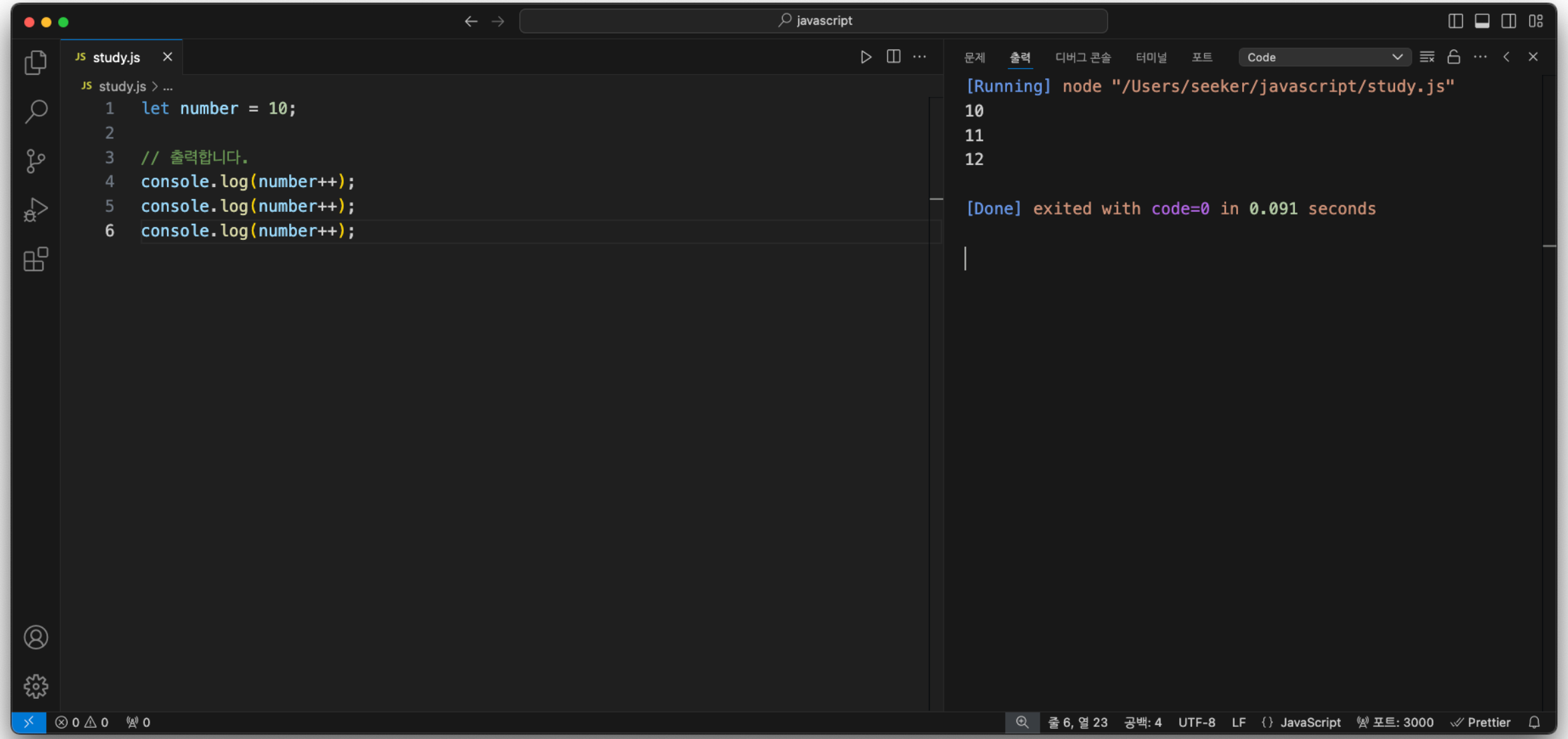
Part 2

변수에 적용할 수 있는 연산자 > 증감 연산자



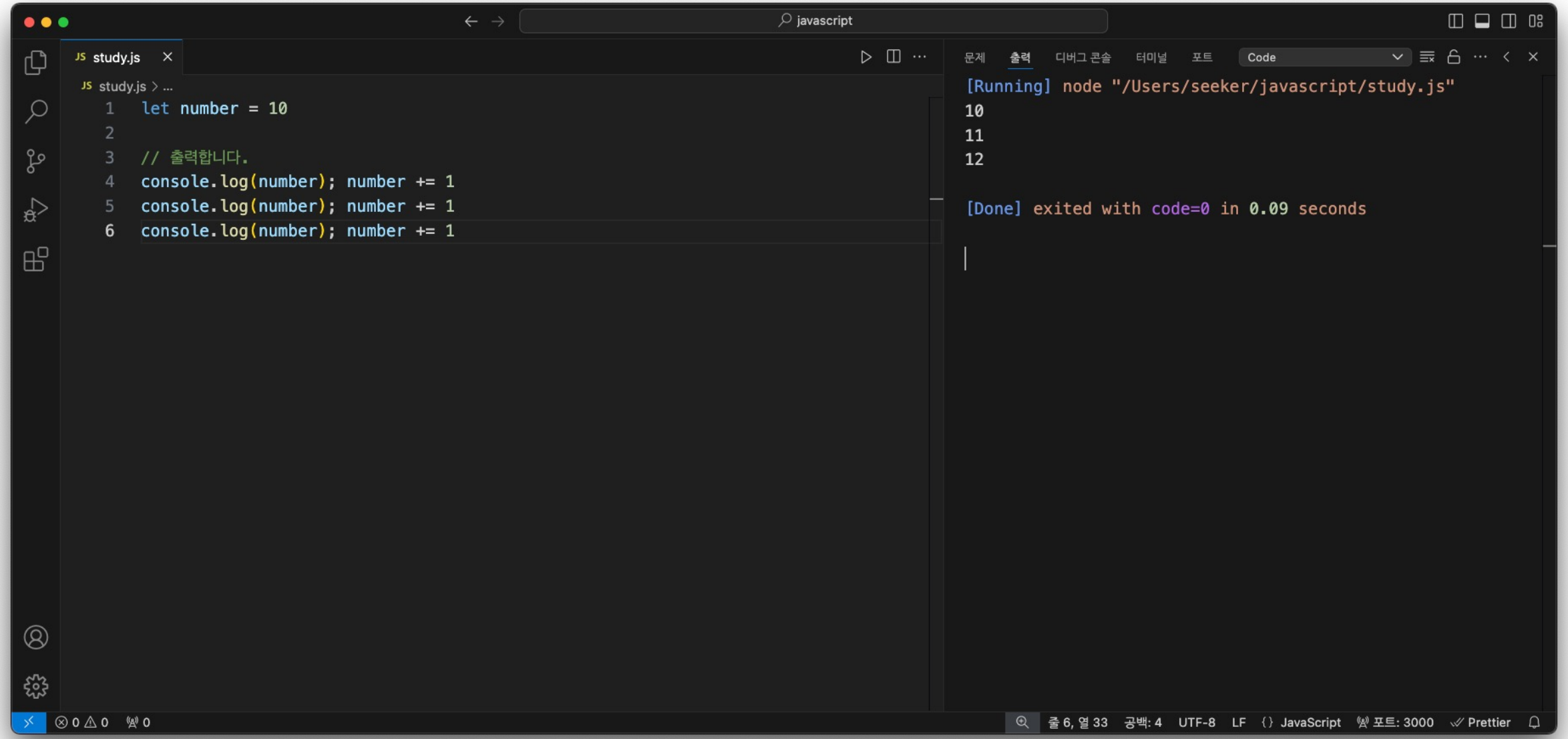
Part 2

변수에 적용할 수 있는 연산자 > 증감 연산자



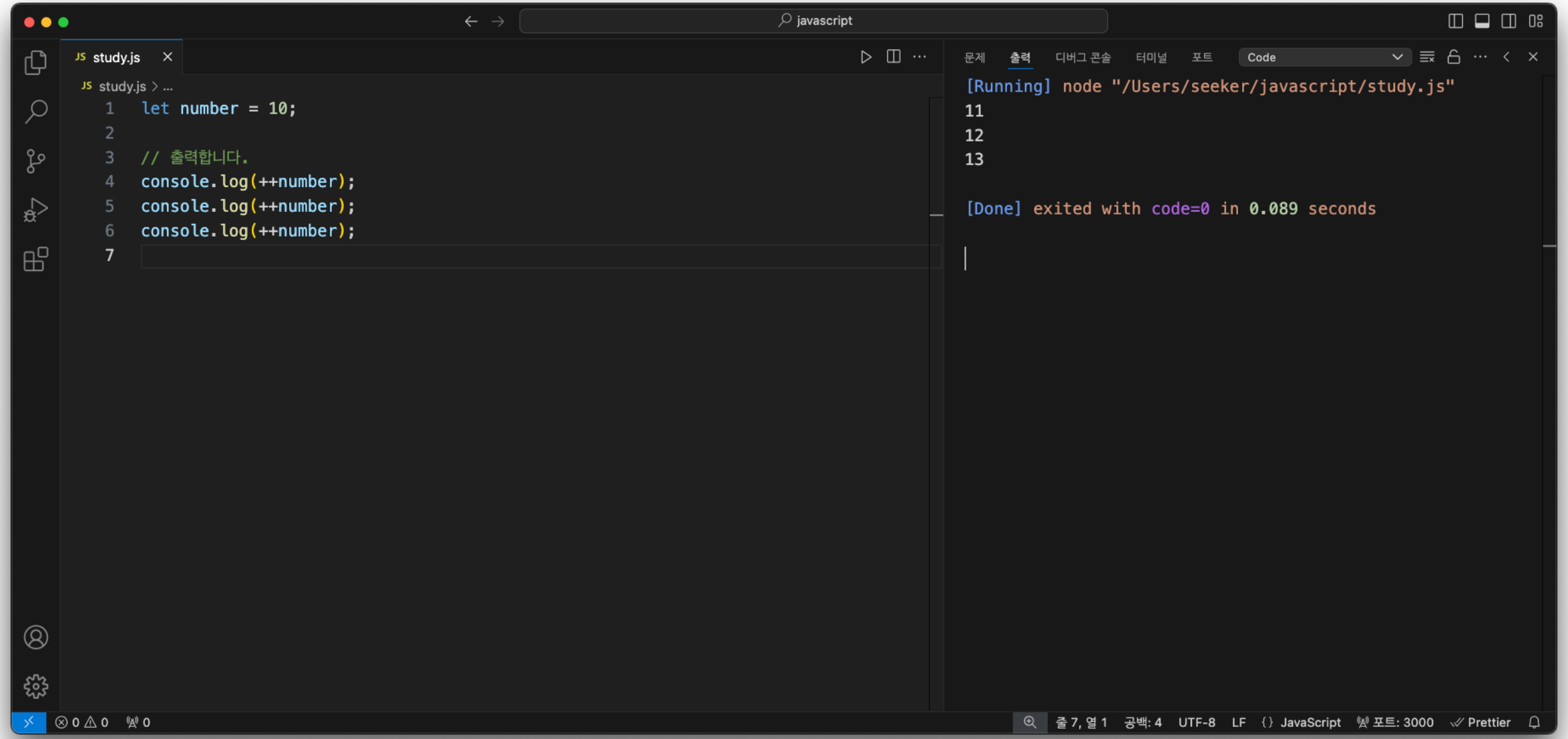
Part 2

변수에 적용할 수 있는 연산자 > 증감 연산자



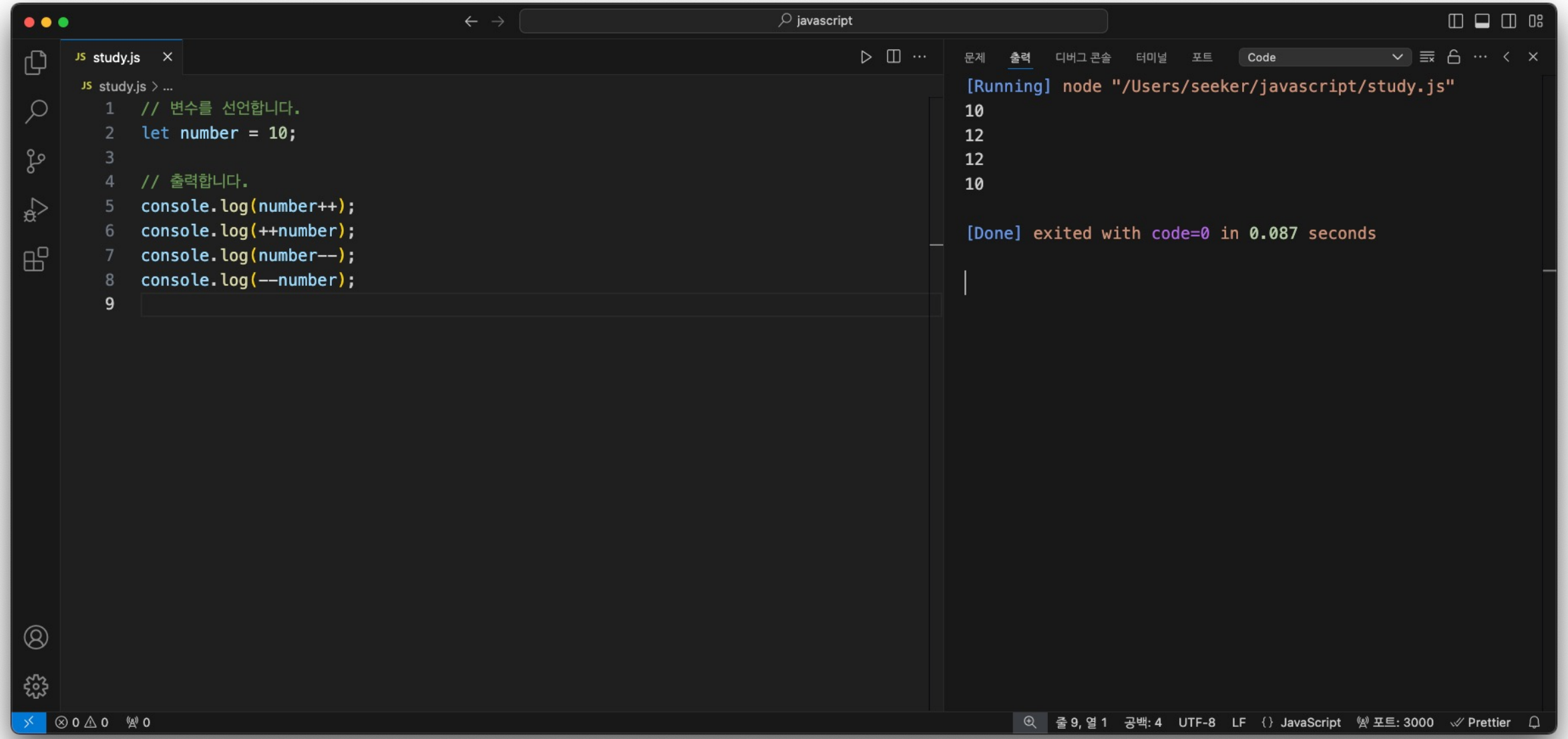
Part 2

변수에 적용할 수 있는 연산자 > 증감 연산자



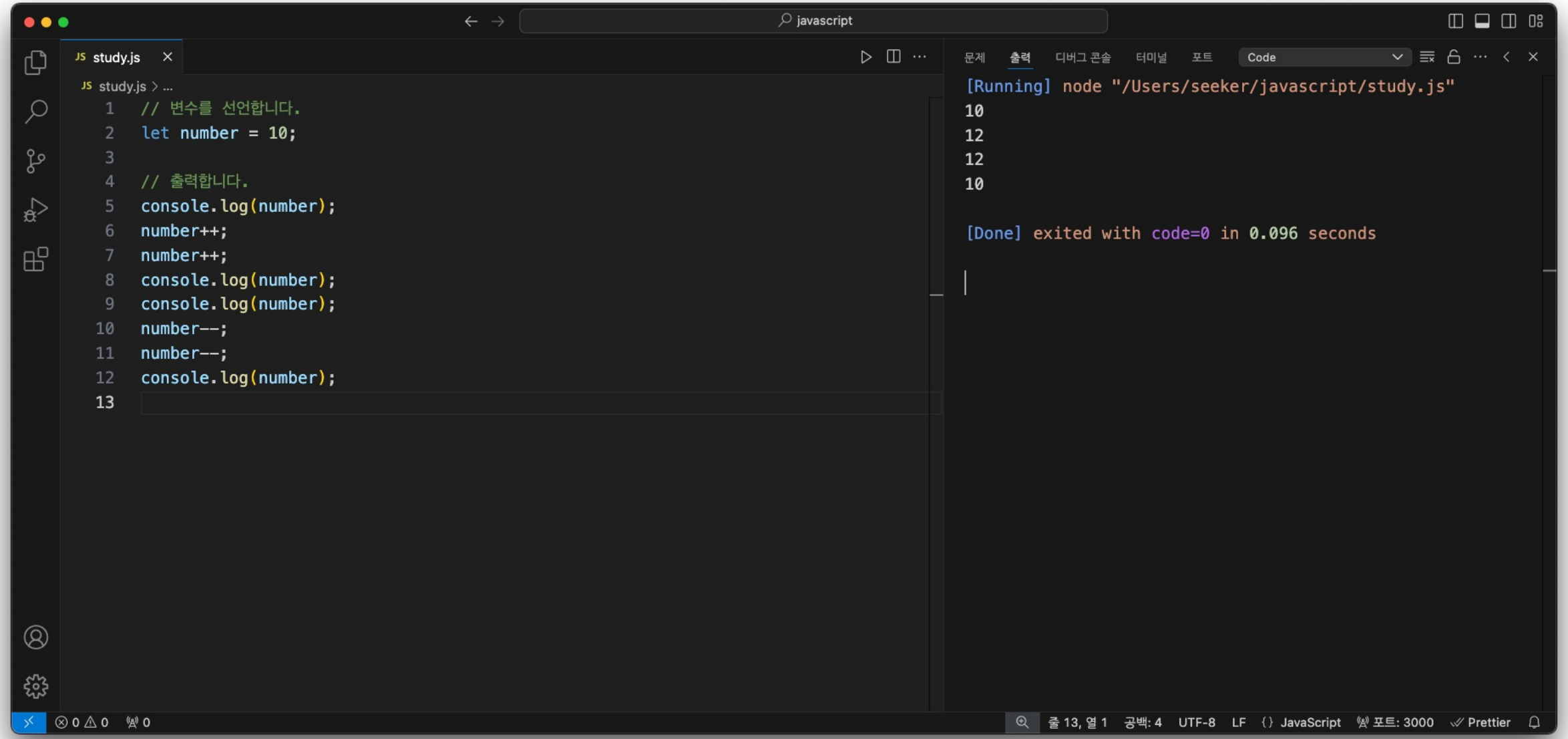
Part 2

변수에 적용할 수 있는 연산자 > 증감 연산자

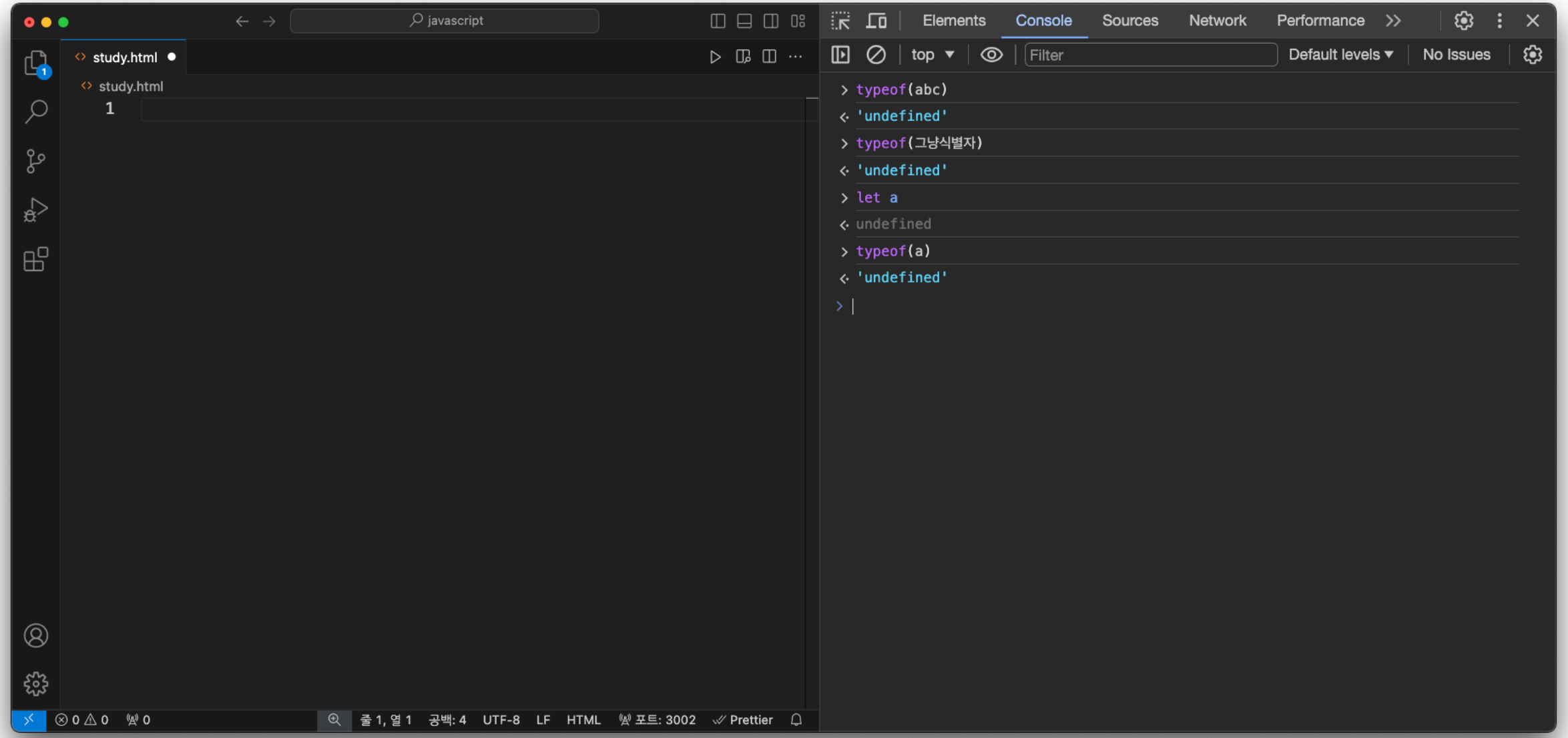


Part 2

변수에 적용할 수 있는 연산자 > 증감 연산자



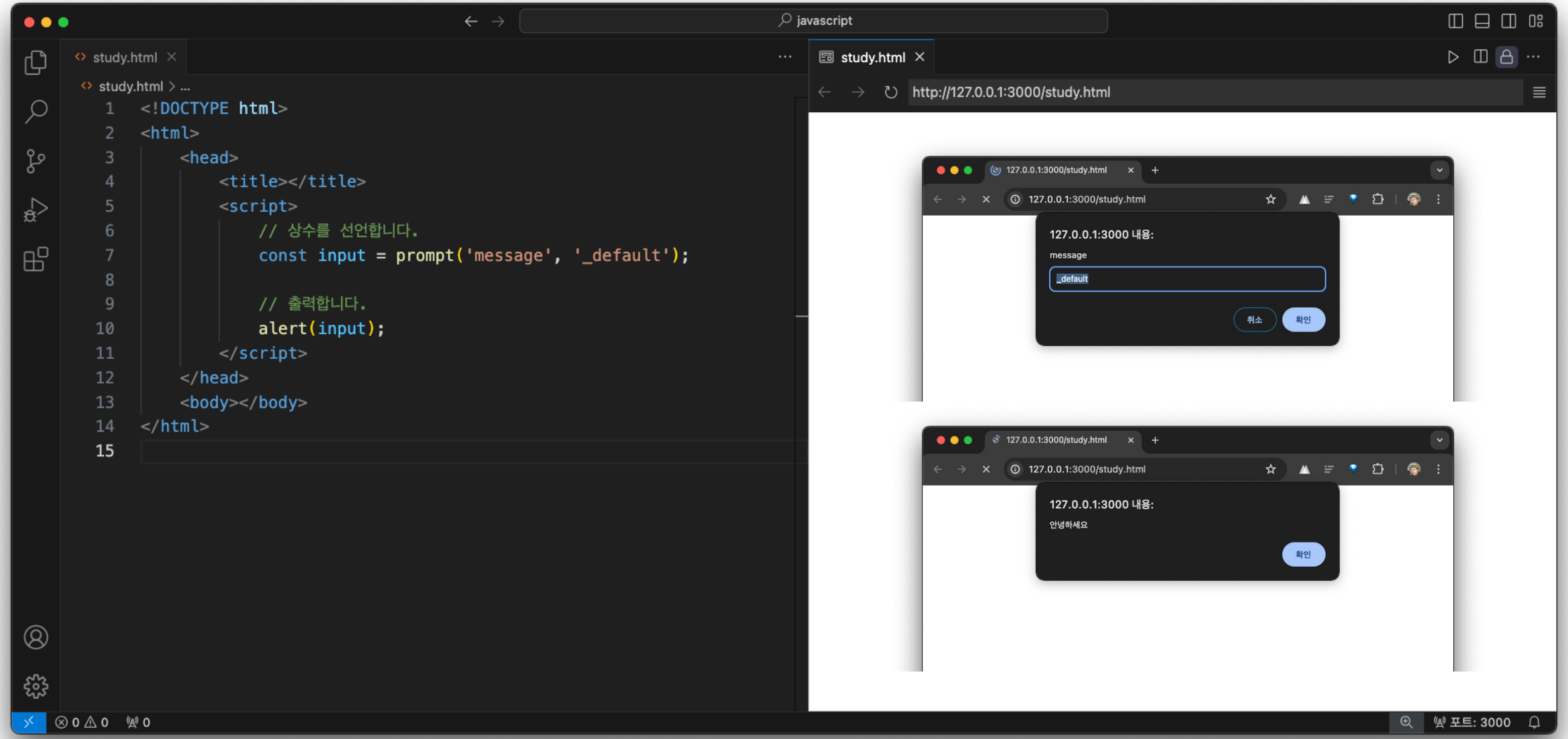
Part 2 undefined 자료형



자료형 변환

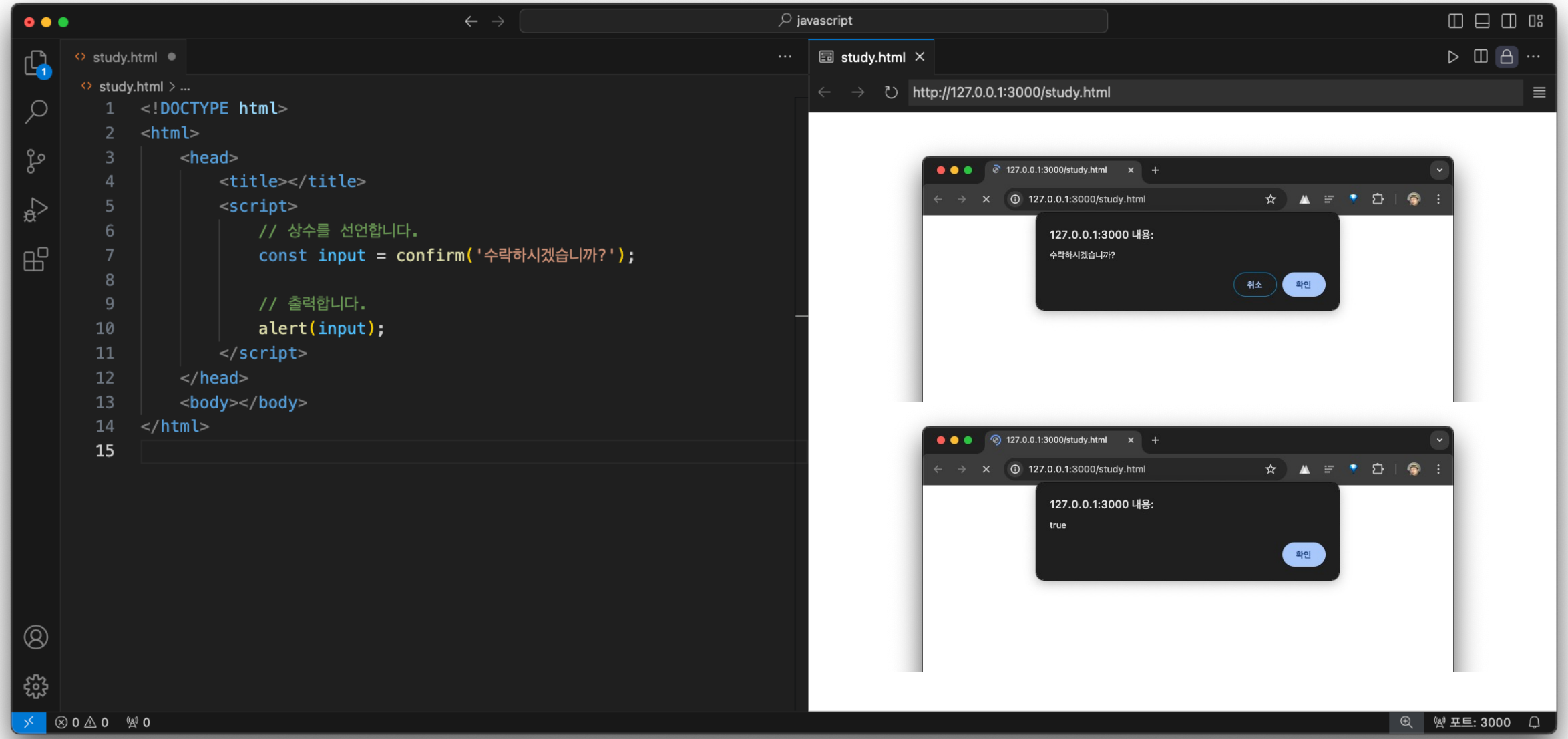
Part 2

문자열 변환


javascript
study.html
study.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title></title>
5 <script>
6 // 상수를 선언합니다.
7 const input = prompt('message', '_default');
8
9 // 출력합니다.
10 alert(input);
11 </script>
12 </head>
13 <body></body>
14 </html>
15
study.html
http://127.0.0.1:3000/study.html
127.0.0.1:3000/study.html
127.0.0.1:3000 내용:
message
_default
취소
확인
127.0.0.1:3000/study.html
127.0.0.1:3000 내용:
안녕하세요
확인
포트: 3000

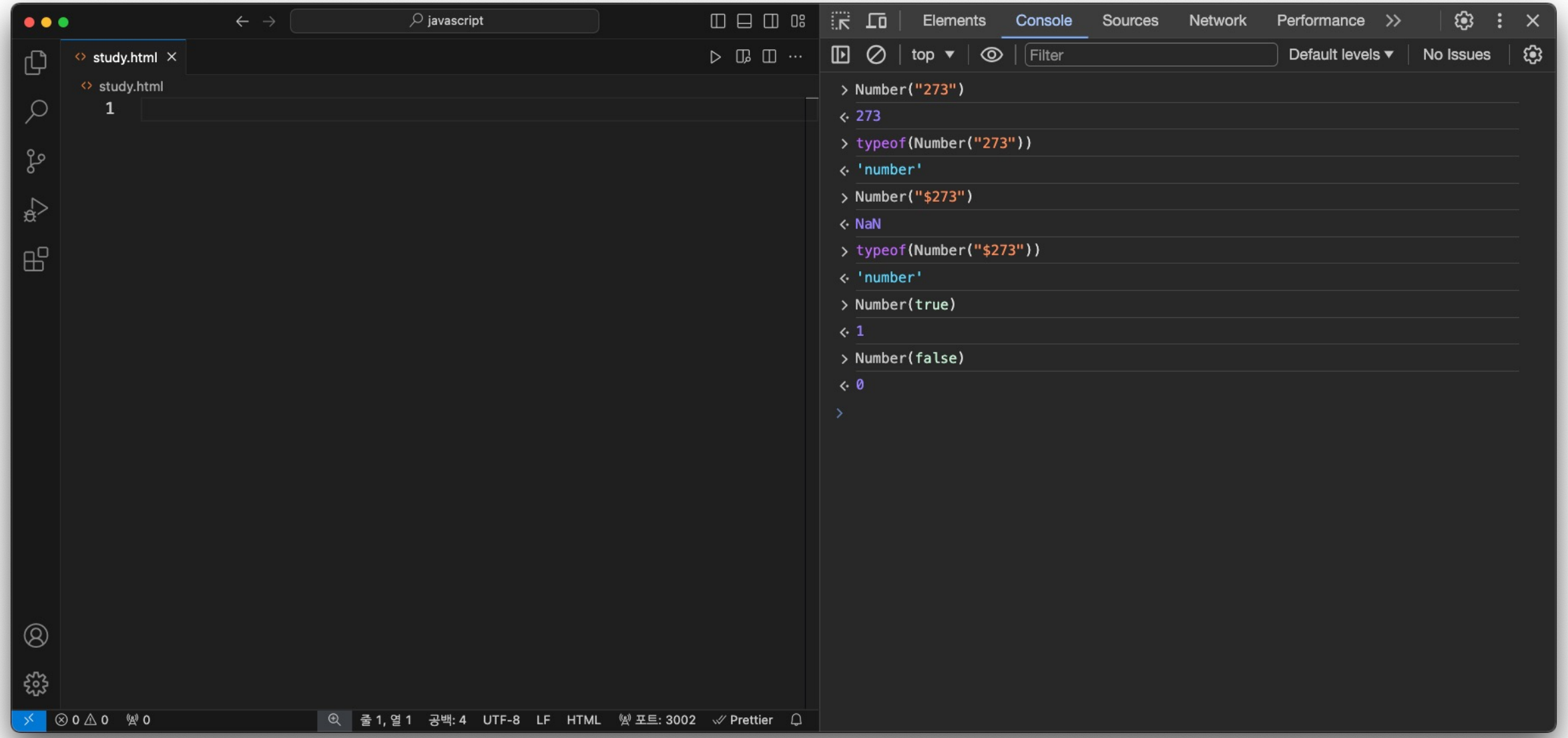
Part 2

불 입력


javascript
study.html
study.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title></title>
5 <script>
6 // 상수를 선언합니다.
7 const input = confirm('수락하시겠습니까?');
8
9 // 출력합니다.
10 alert(input);
11 </script>
12 </head>
13 <body></body>
14 </html>
15
study.html
http://127.0.0.1:3000/study.html
127.0.0.1:3000/study.html
127.0.0.1:3000 내용:
수락하시겠습니까?
취소
확인
127.0.0.1:3000/study.html
127.0.0.1:3000 내용:
true
확인
포트: 3000

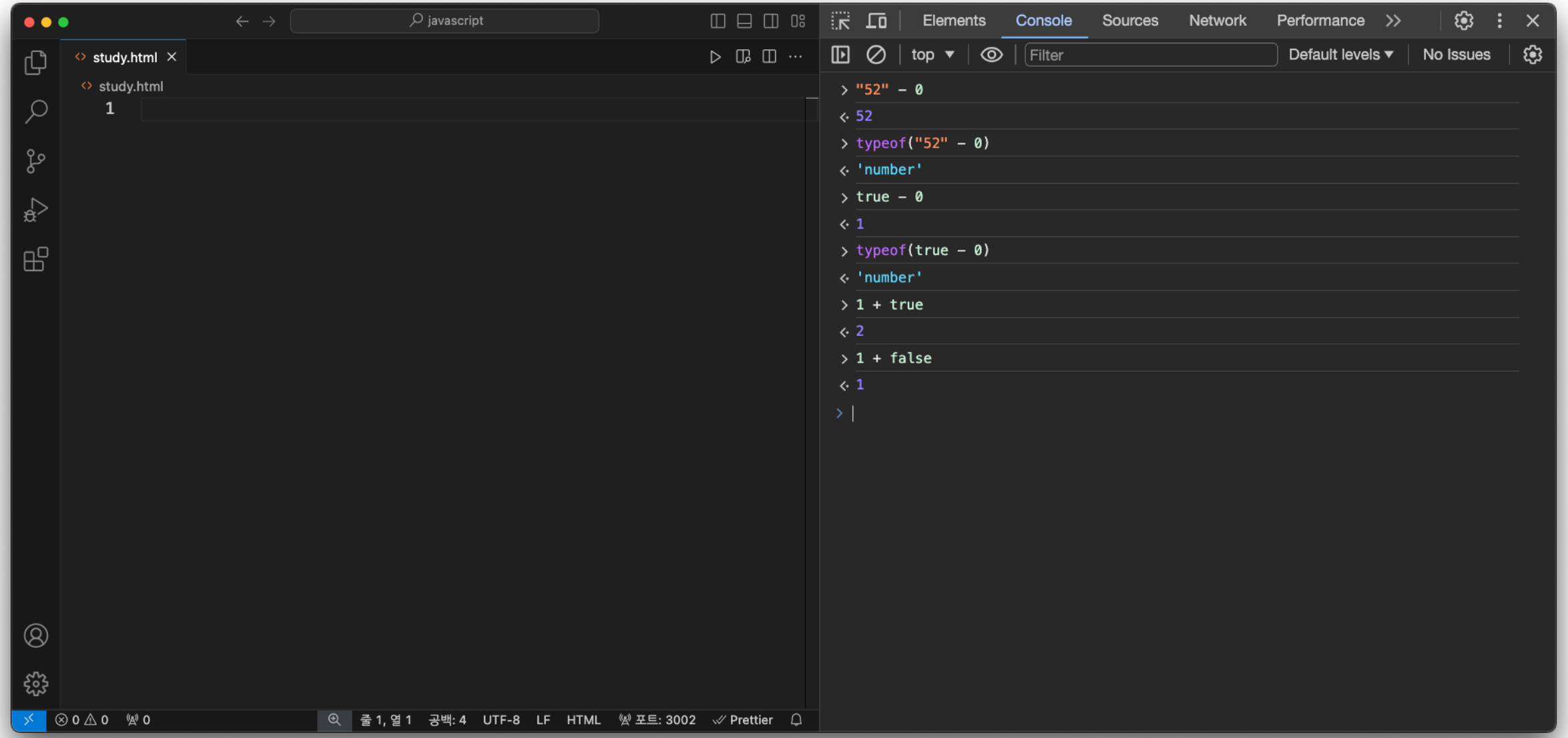
Part 2

숫자 자료형으로 변환하기



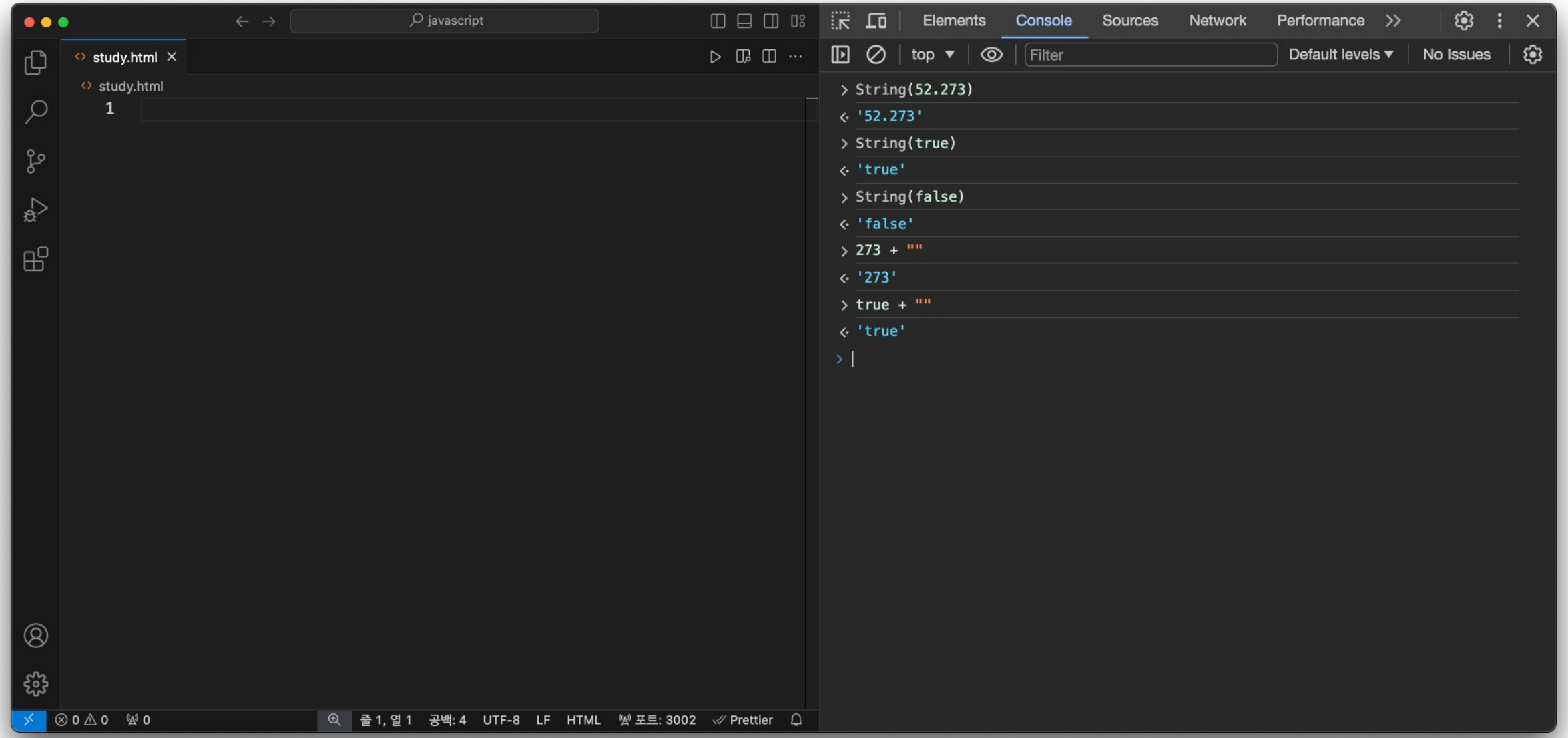
Part 2

숫자 자료형으로 변환하기 > 숫자 연산자 사용



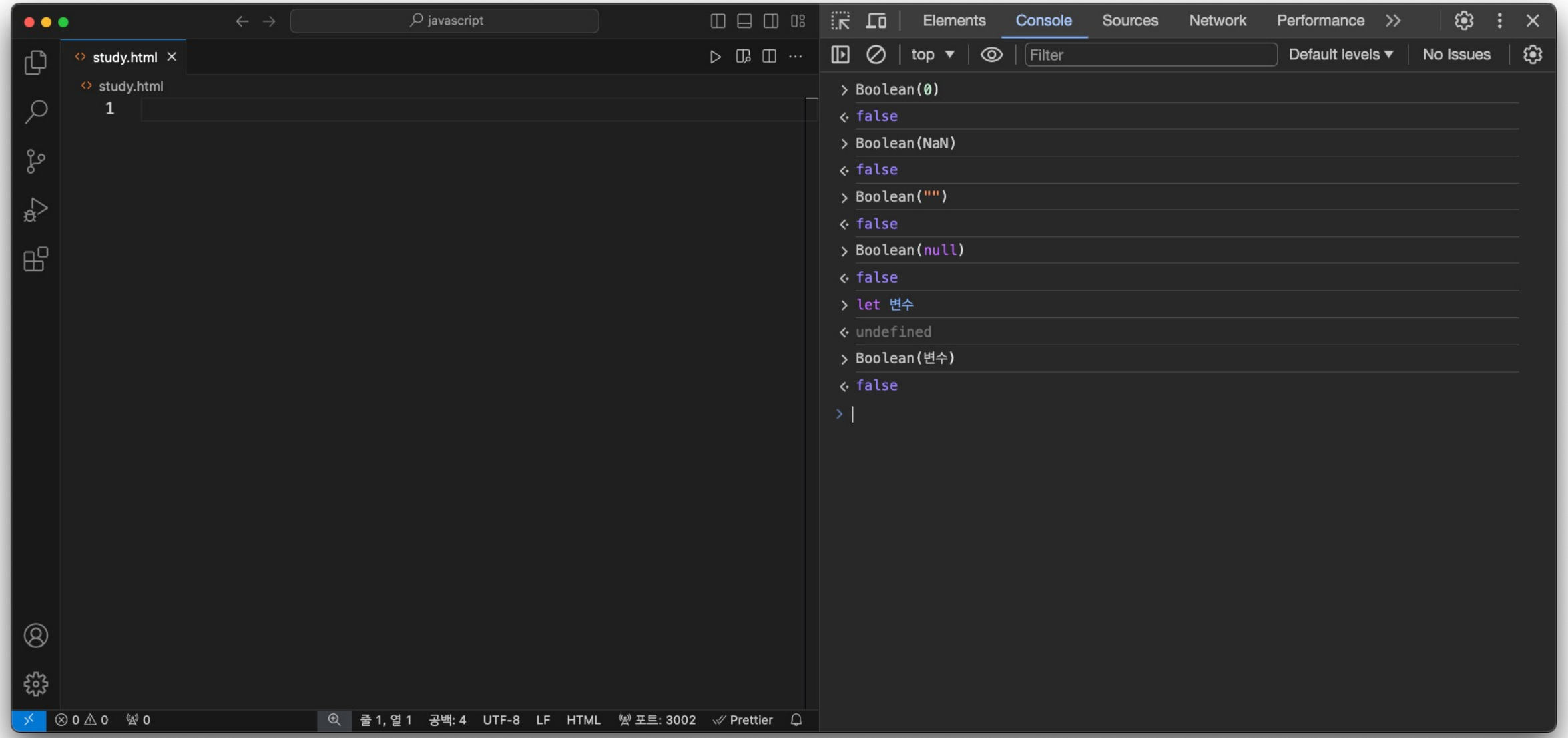
Part 2

문자열 자료형으로 변환하기



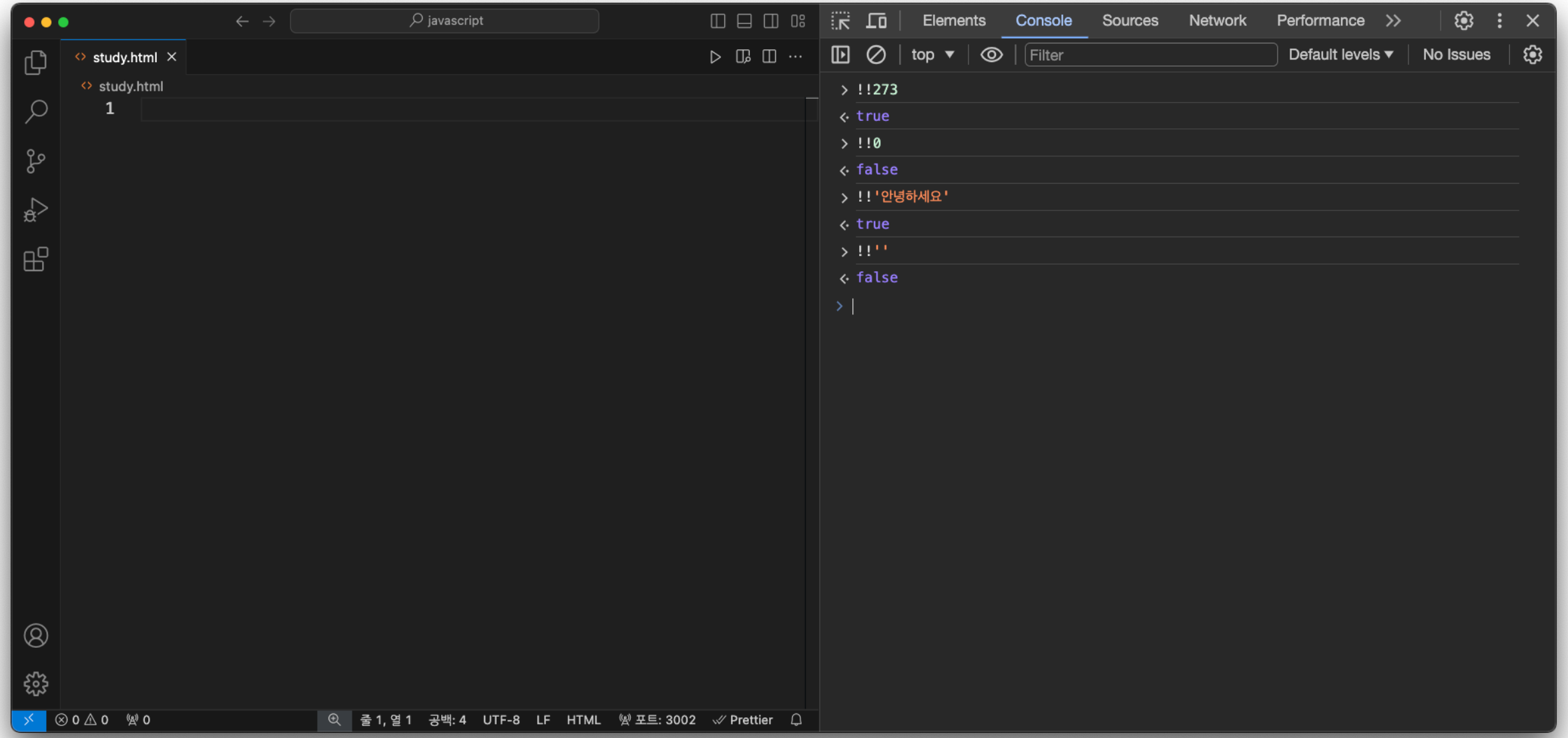
Part 2

불 자료형으로 변환하기



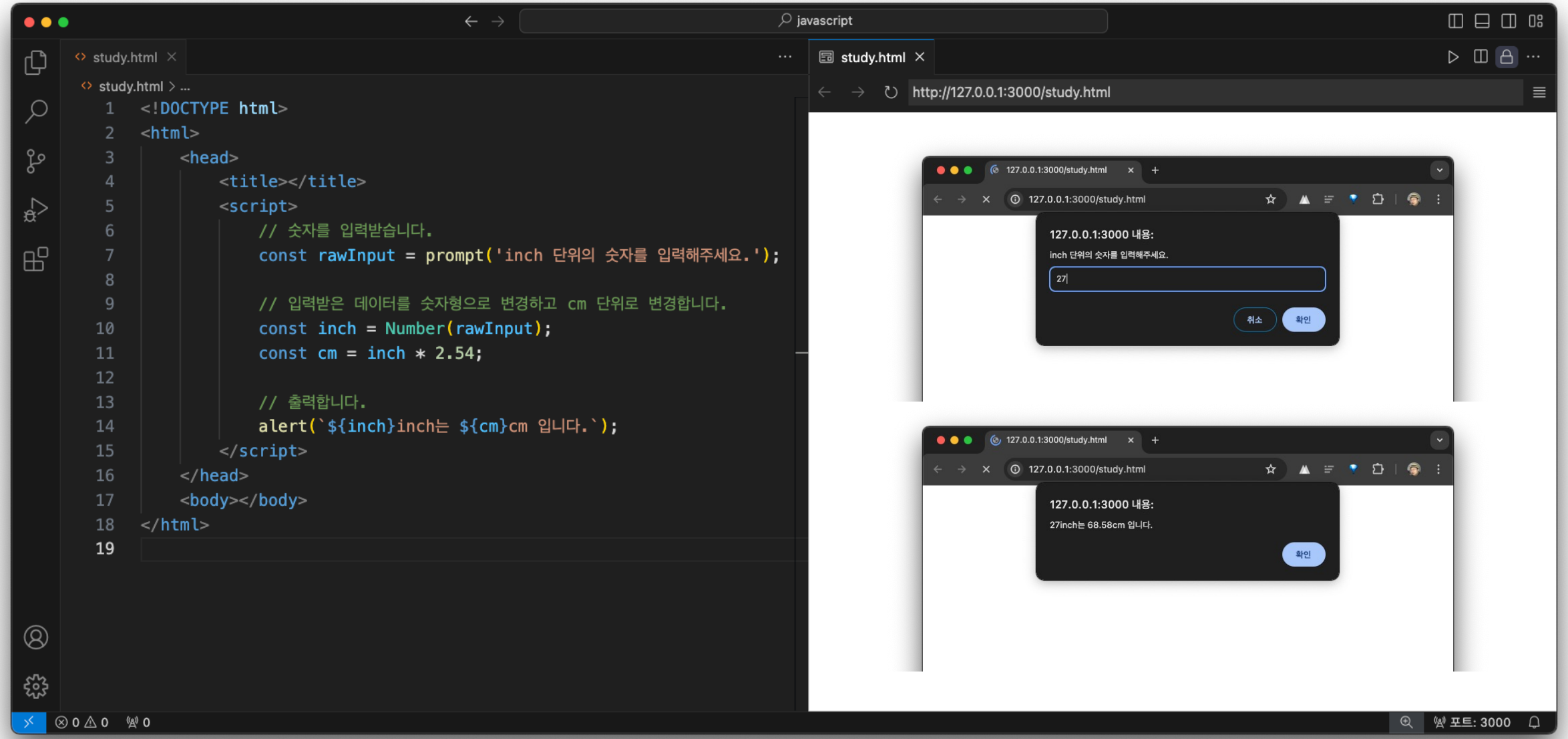
Part 2

불 자료형으로 변환하기 > 논리 부정 연산자 사용

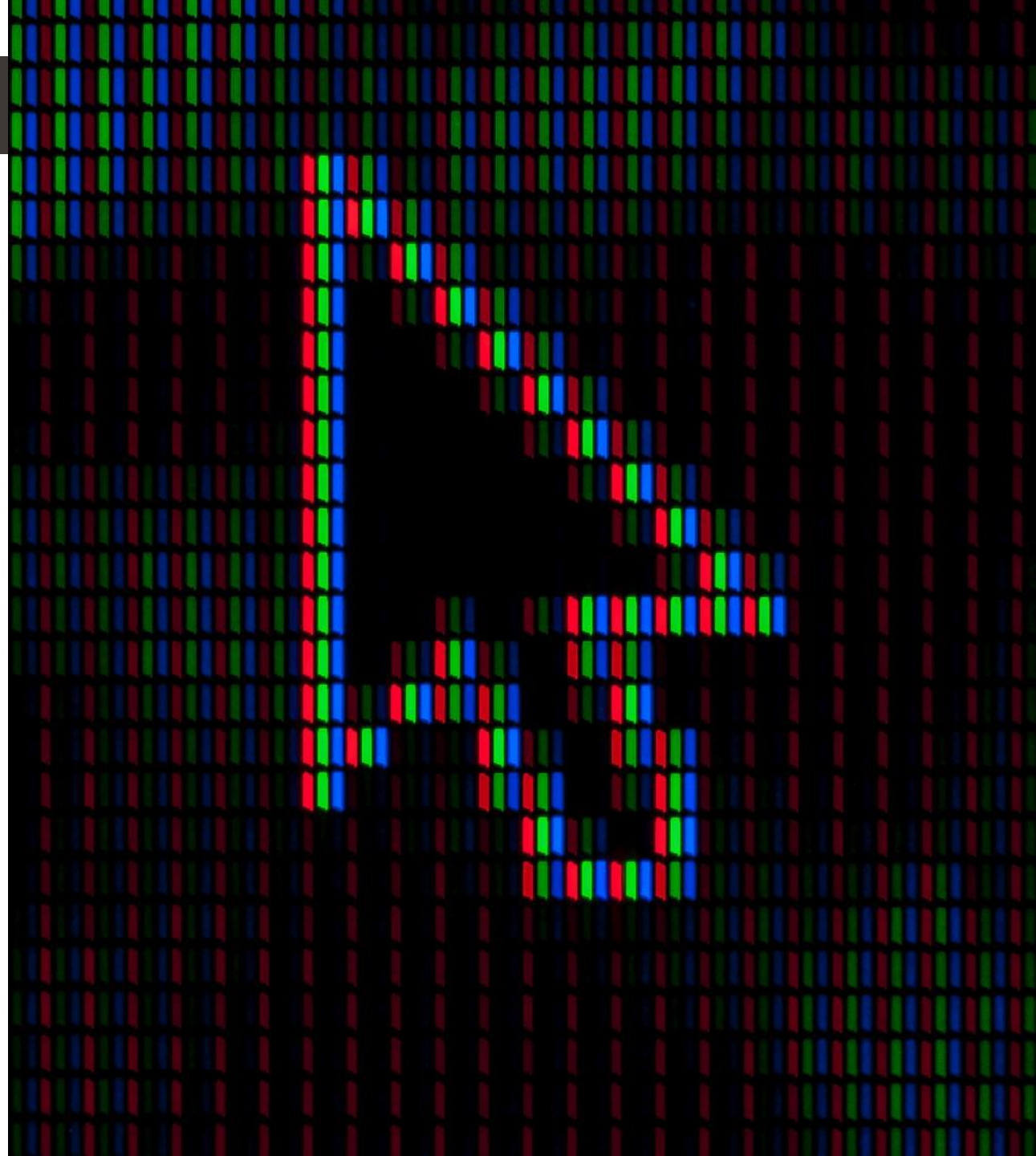


Part 2

inch를 cm 단위로 변경하기



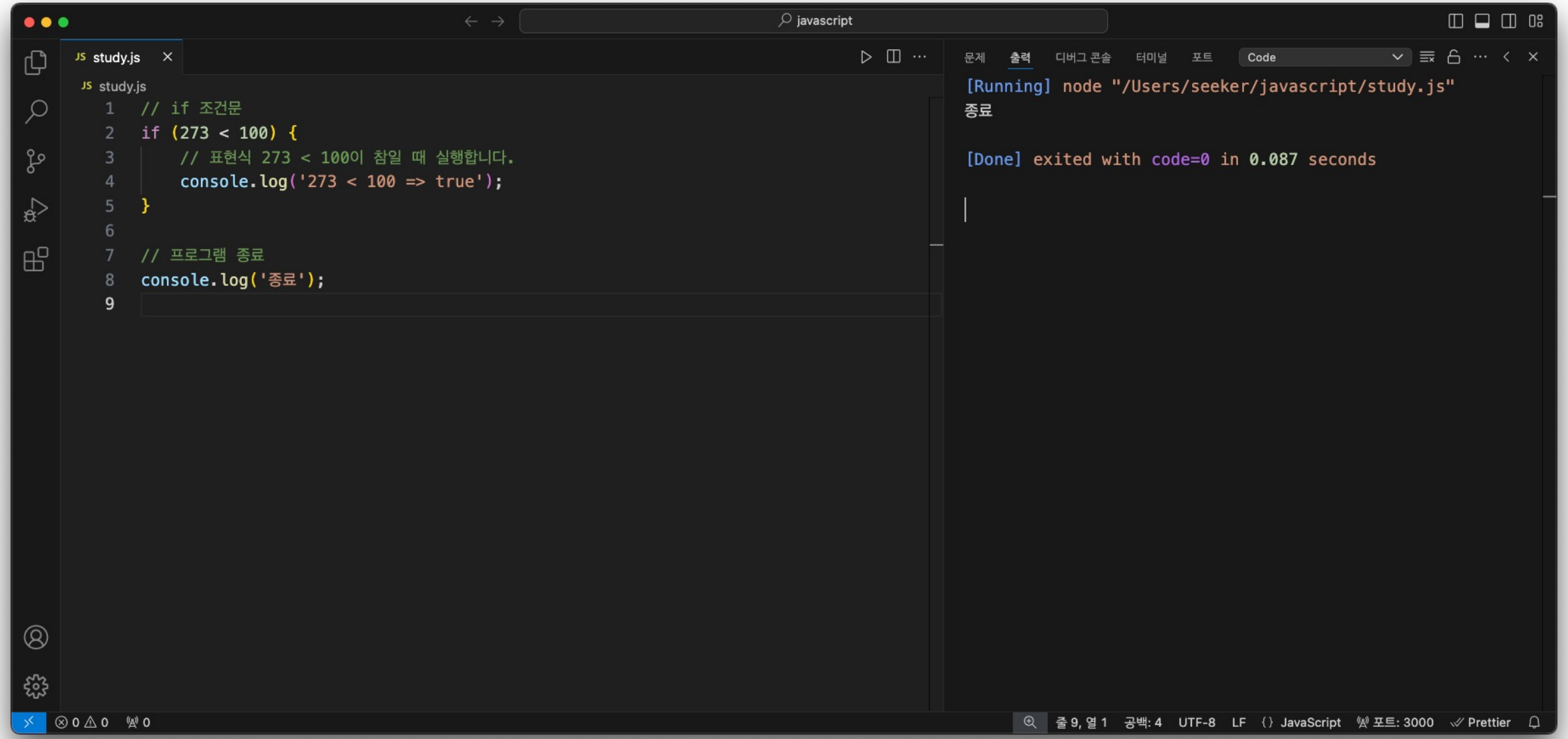
목차 a table of contents



if 조건문

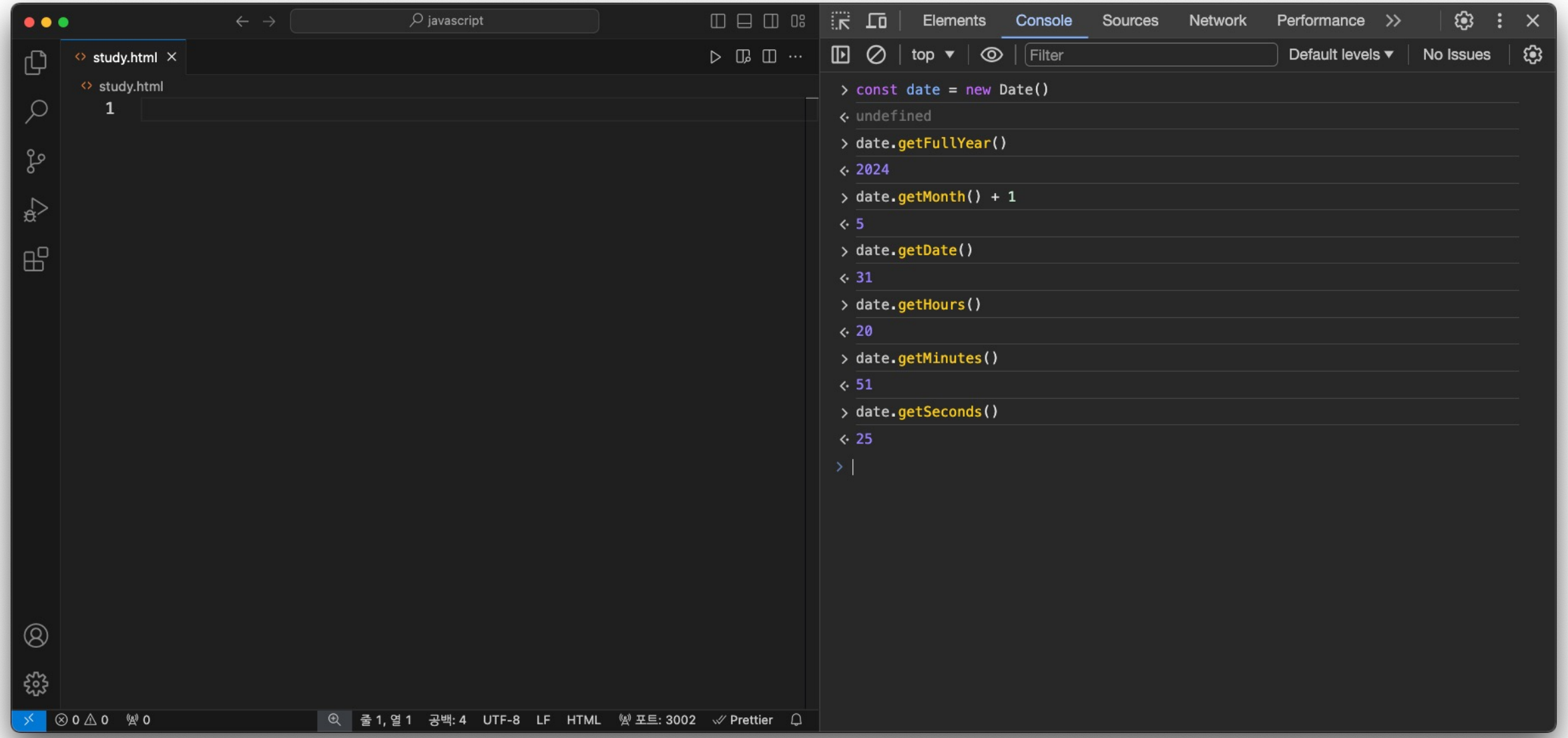
Part 3

if 조건문

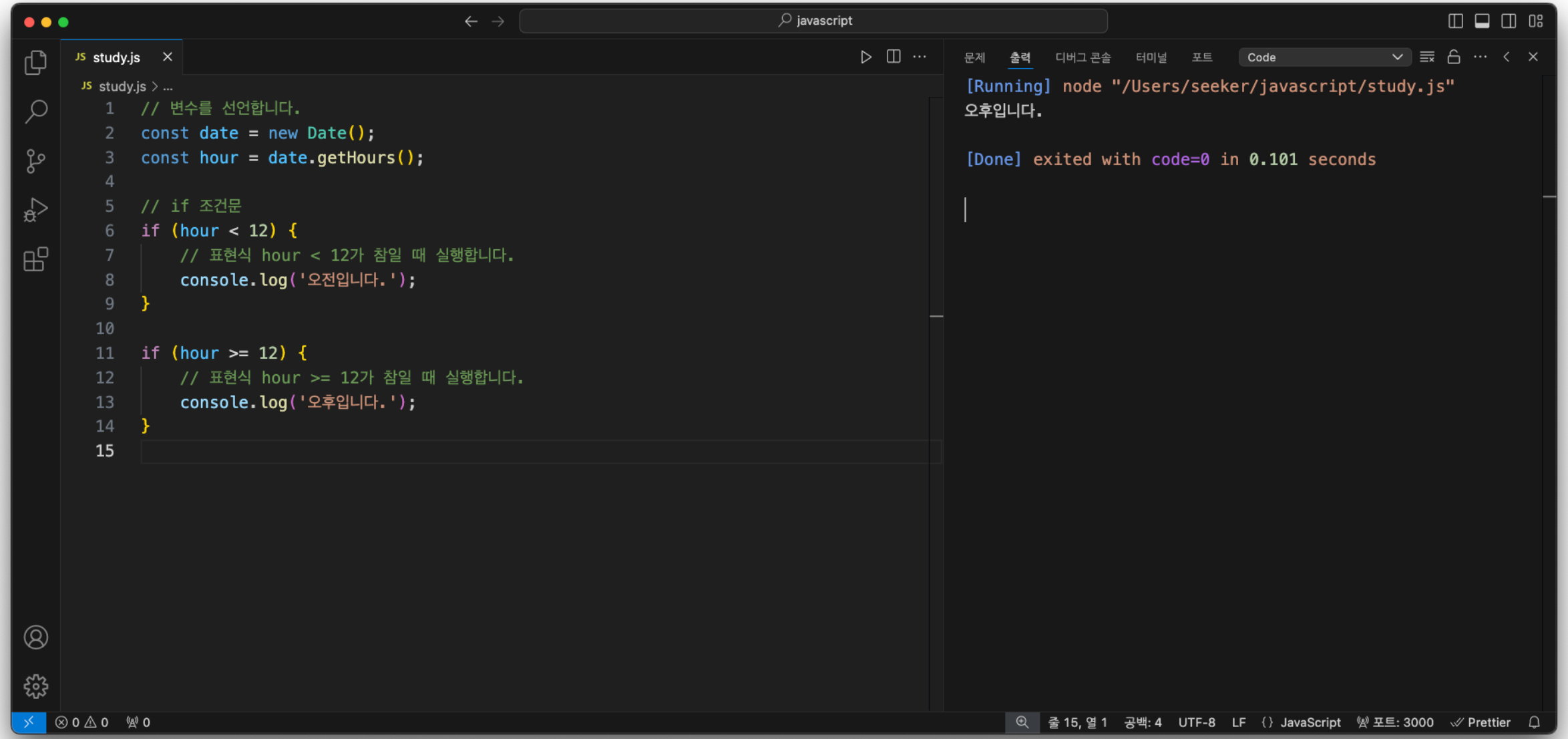

javascript
study.js
study.js
1 // if 조건문
2 if (273 < 100) {
3 // 표현식 273 < 100이 참일 때 실행합니다.
4 console.log('273 < 100 => true');
5 }
6
7 // 프로그램 종료
8 console.log('종료');
9
문제
출력
디버그 콘솔
터미널
포트
Code
[Running] node "/Users/seeker/javascript/study.js"
종료
[Done] exited with code=0 in 0.087 seconds
⊗ 0 ⚠ 0
0
줄 9, 열 1
공백: 4
UTF-8
LF
{} JavaScript
포트: 3000
Prettier

Part 3

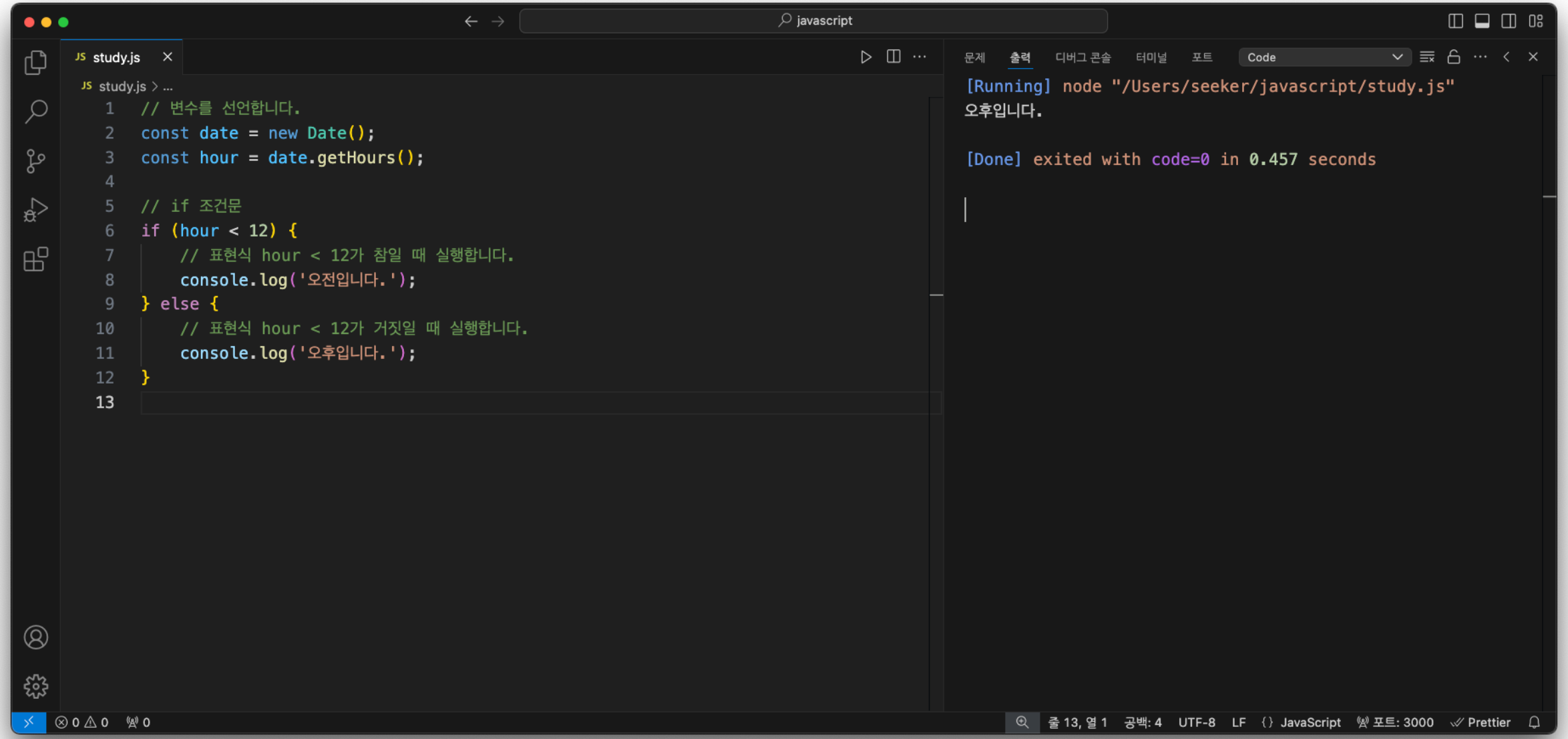
if 조건문


javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
> const date = new Date()
< undefined
> date.getFullYear()
< 2024
> date.getMonth() + 1
< 5
> date.getDate()
< 31
> date.getHours()
< 20
> date.getMinutes()
< 51
> date.getSeconds()
< 25
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

Part 3 if 조건문

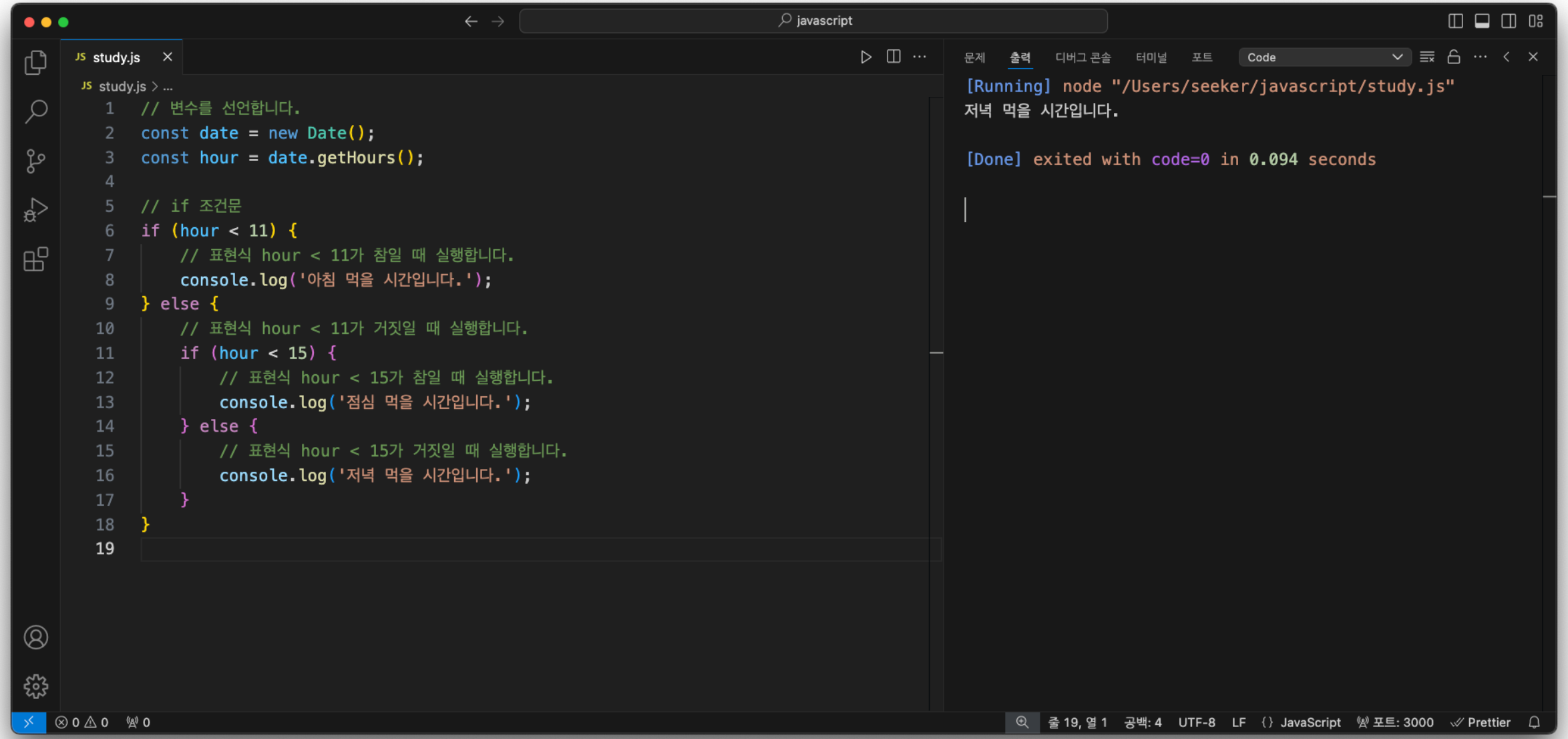

javascript
study.js
study.js > ...
1 // 변수를 선언합니다.
2 const date = new Date();
3 const hour = date.getHours();
4
5 // if 조건문
6 if (hour < 12) {
7 // 표현식 hour < 12가 참일 때 실행합니다.
8 console.log('오전입니다.');
9 }
10
11 if (hour >= 12) {
12 // 표현식 hour >= 12가 참일 때 실행합니다.
13 console.log('오후입니다.');
14 }
15
문제 출력 디버그 콘솔 터미널 포트
Code
[Running] node "/Users/seeker/javascript/study.js"
오후입니다.
[Done] exited with code=0 in 0.101 seconds
0 0 0
줄 15, 열 1 공백: 4 UTF-8 LF JavaScript 포트: 3000 Prettier

Part 3 if else 조건문

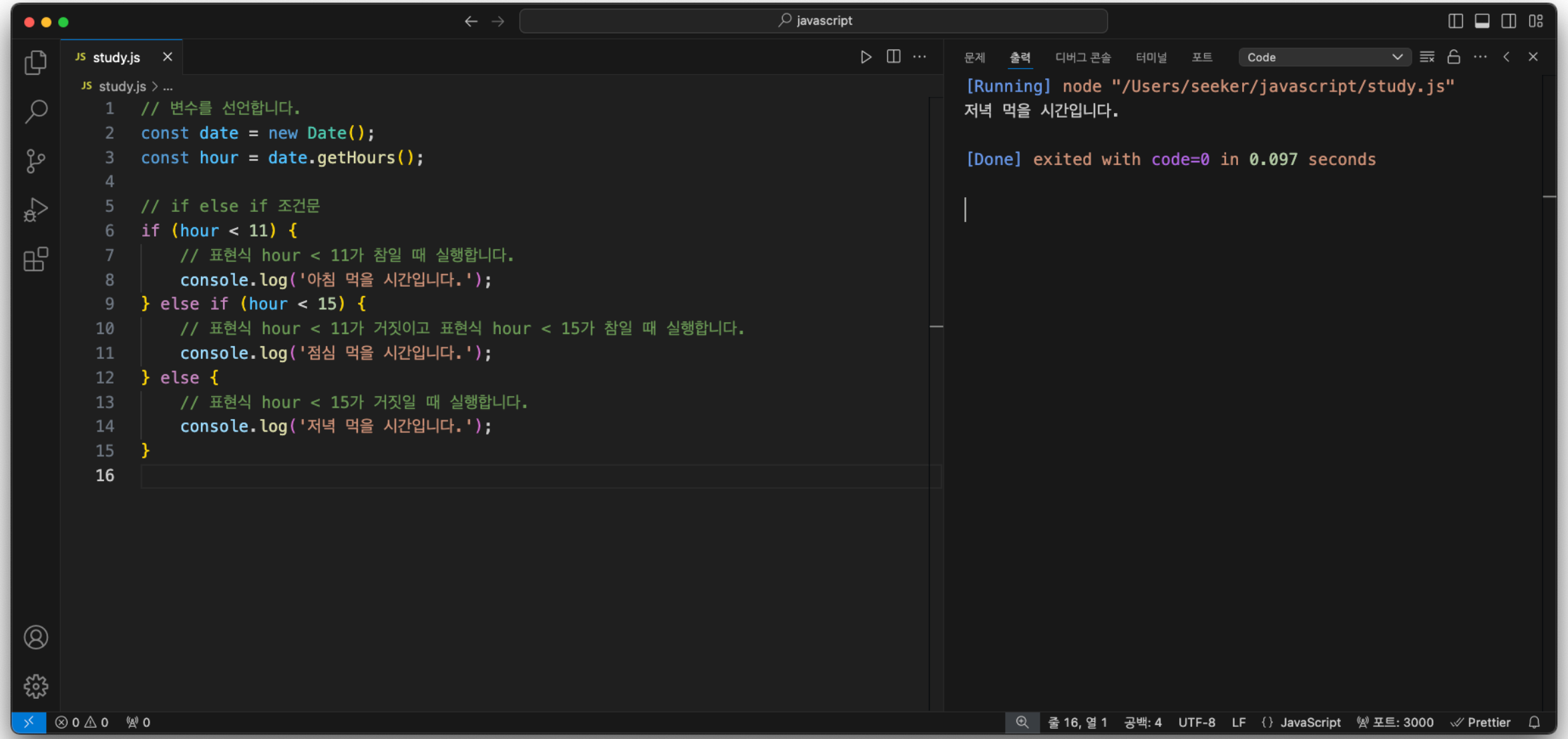


Part 3

중첩 조건문

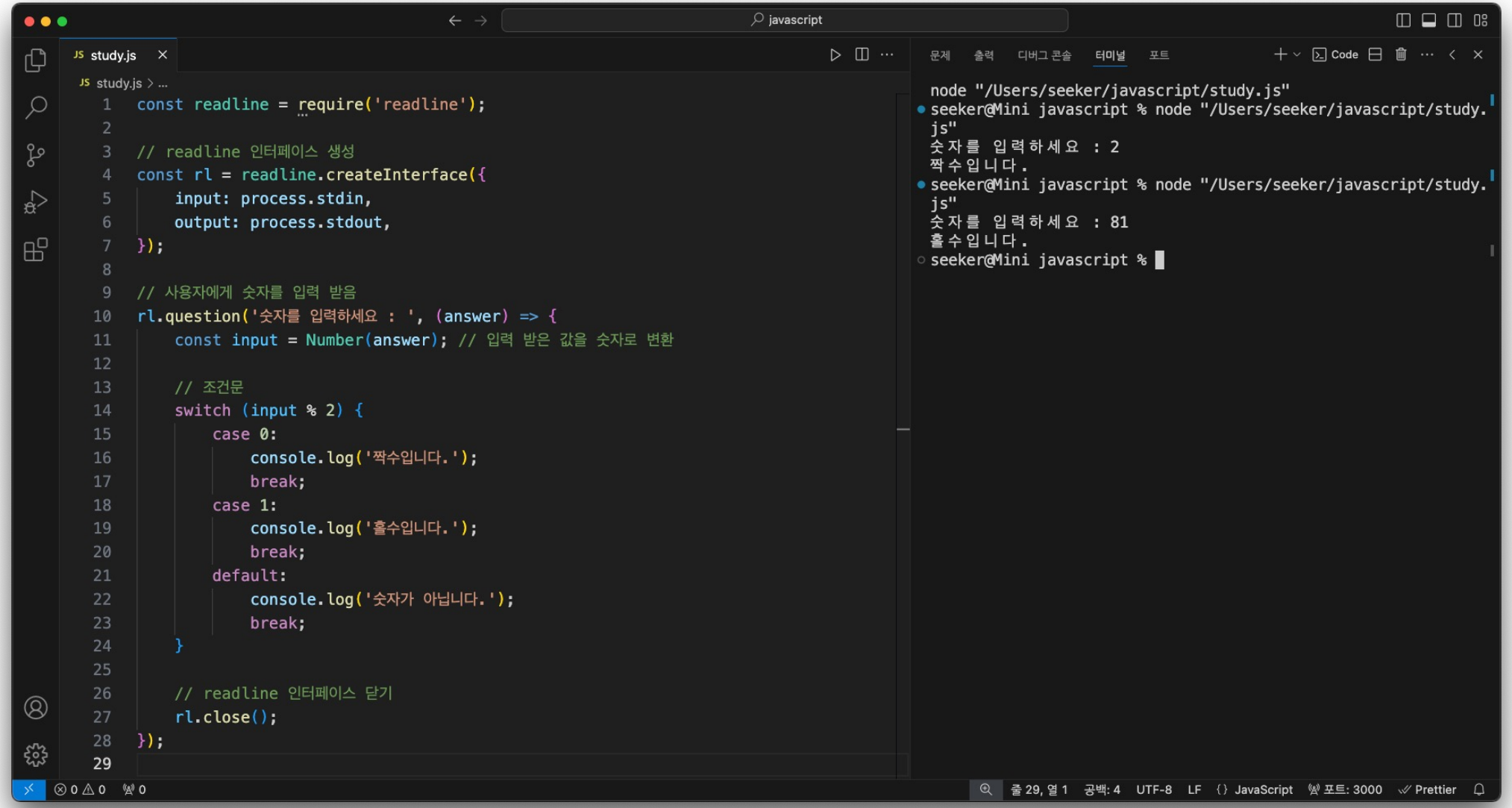

javascript
study.js
study.js > ...
1 // 변수를 선언합니다.
2 const date = new Date();
3 const hour = date.getHours();
4
5 // if 조건문
6 if (hour < 11) {
7 // 표현식 hour < 11가 참일 때 실행합니다.
8 console.log('아침 먹을 시간입니다.');
9 } else {
10 // 표현식 hour < 11가 거짓일 때 실행합니다.
11 if (hour < 15) {
12 // 표현식 hour < 15가 참일 때 실행합니다.
13 console.log('점심 먹을 시간입니다.');
14 } else {
15 // 표현식 hour < 15가 거짓일 때 실행합니다.
16 console.log('저녁 먹을 시간입니다.');
17 }
18 }
19
문제 출력 디버그 콘솔 터미널 포트 Code
[Running] node "/Users/seeker/javascript/study.js"
저녁 먹을 시간입니다.
[Done] exited with code=0 in 0.094 seconds
⊗ 0 ⚠ 0 0
줄 19, 열 1 공백: 4 UTF-8 LF { } JavaScript 포트: 3000 Prettier

Part 3 if else if 조건문

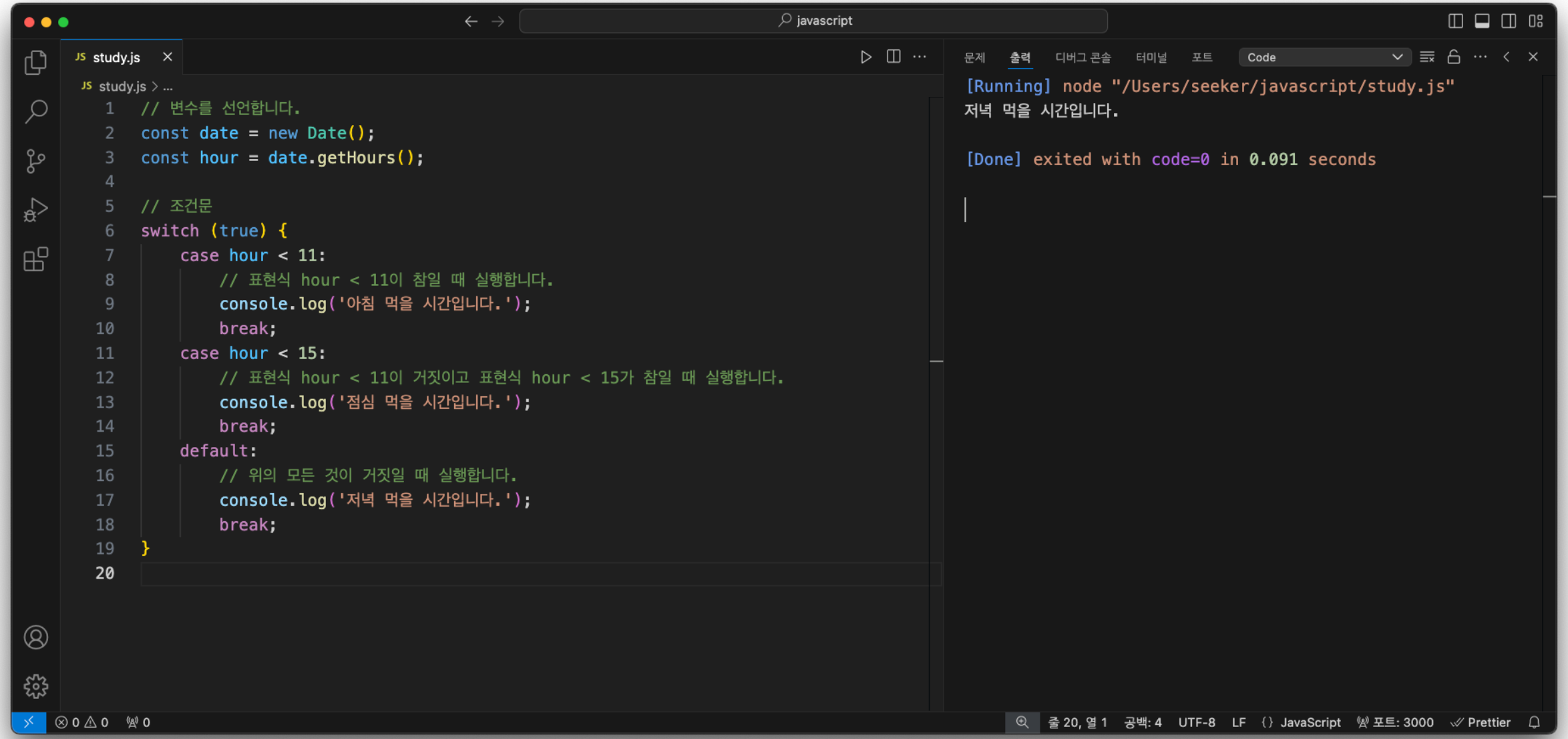


switch 조건문

Part 3 switch 조건문



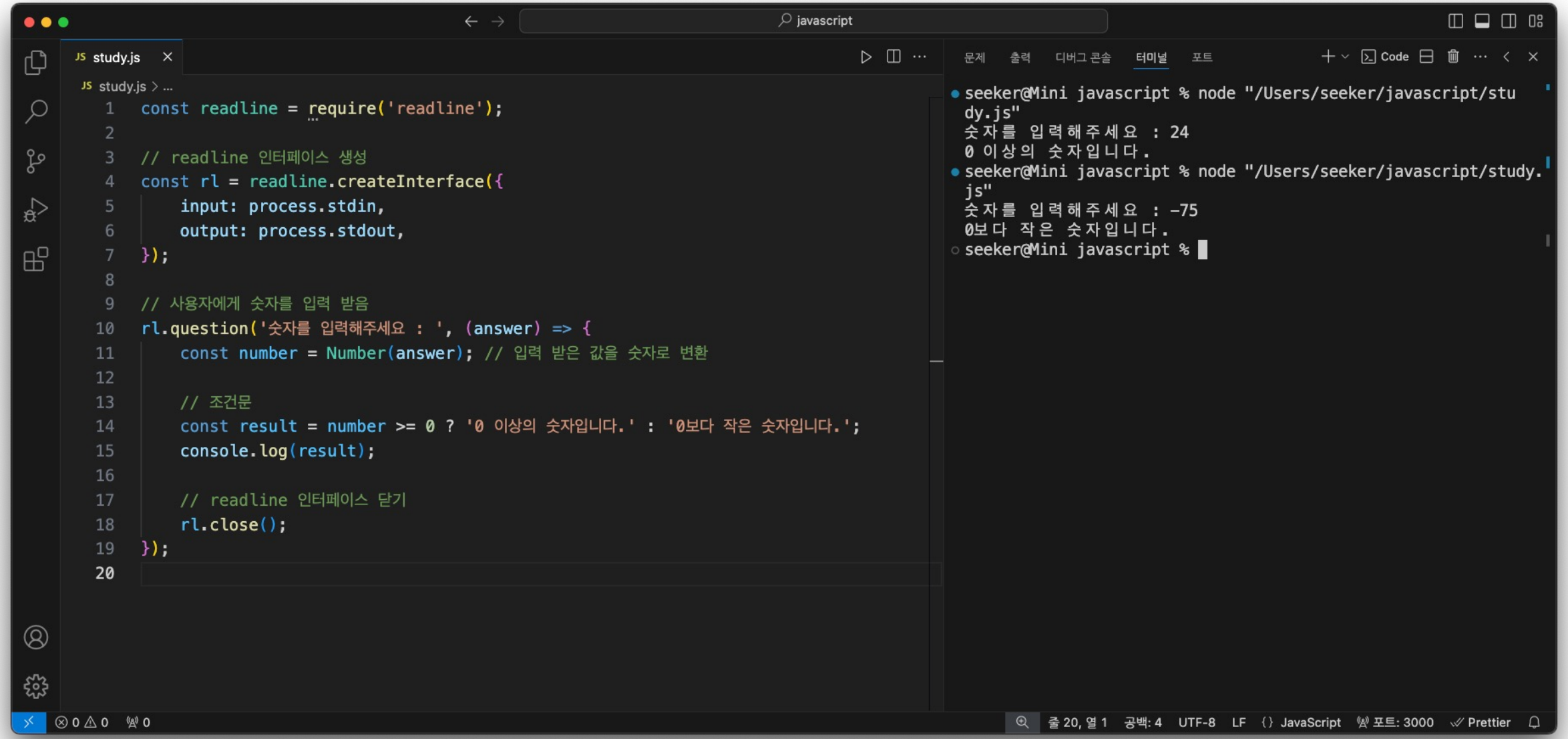
Part 3 switch 조건문 > if 조건문으로 변환



기타 조건문

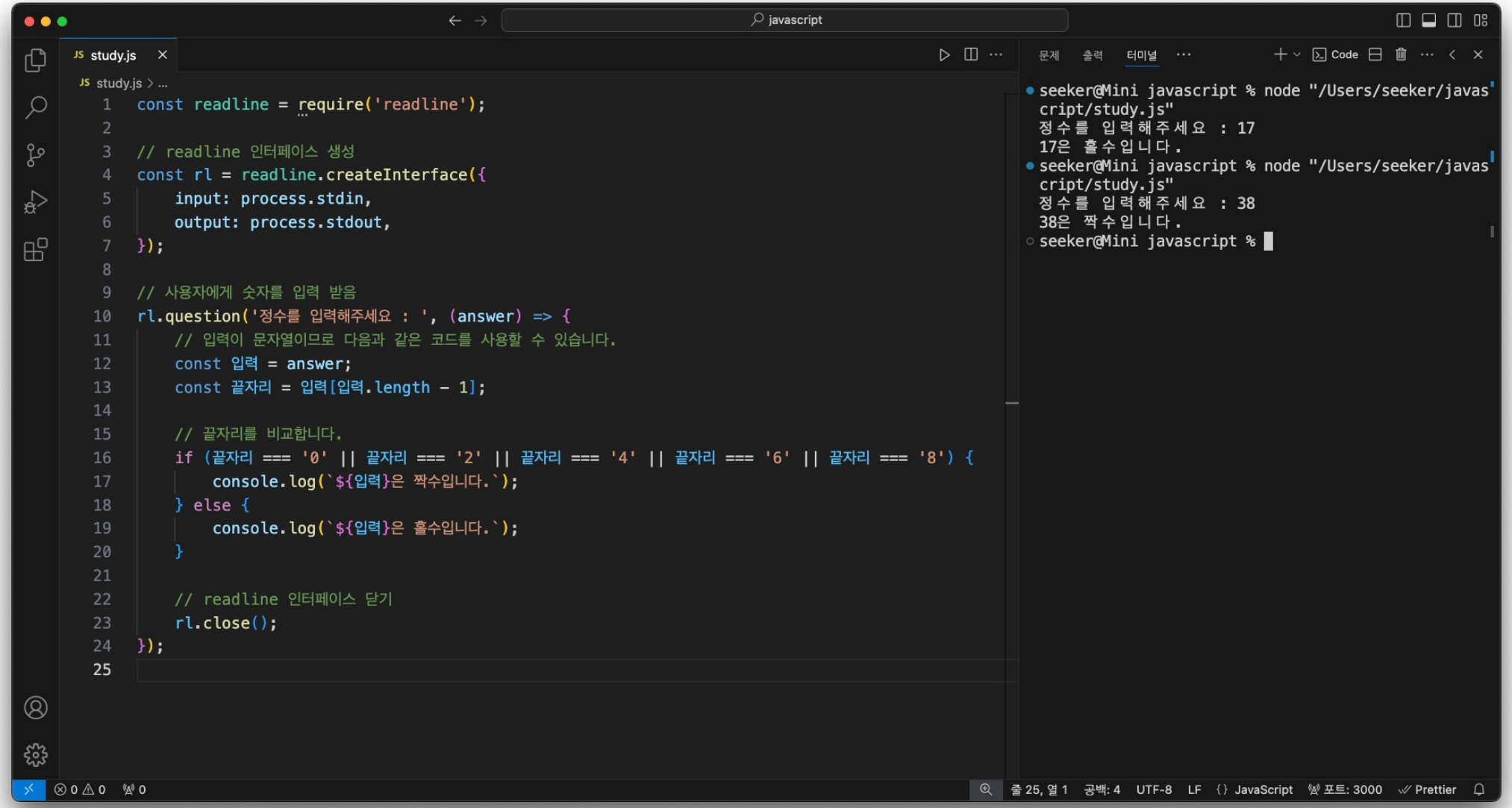
Part 3

조건부 연산자



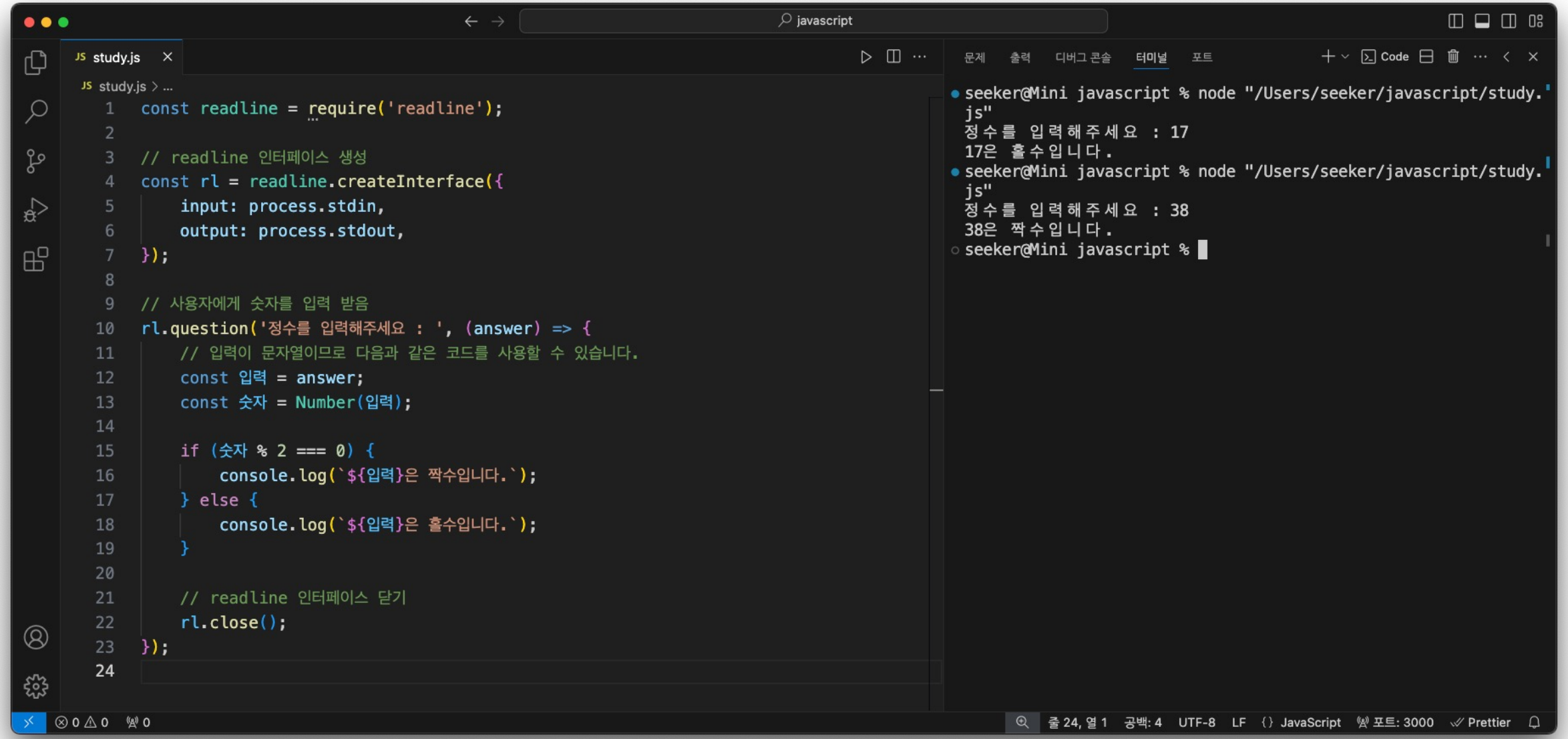
Part 3

짝수와 홀수 구분하기



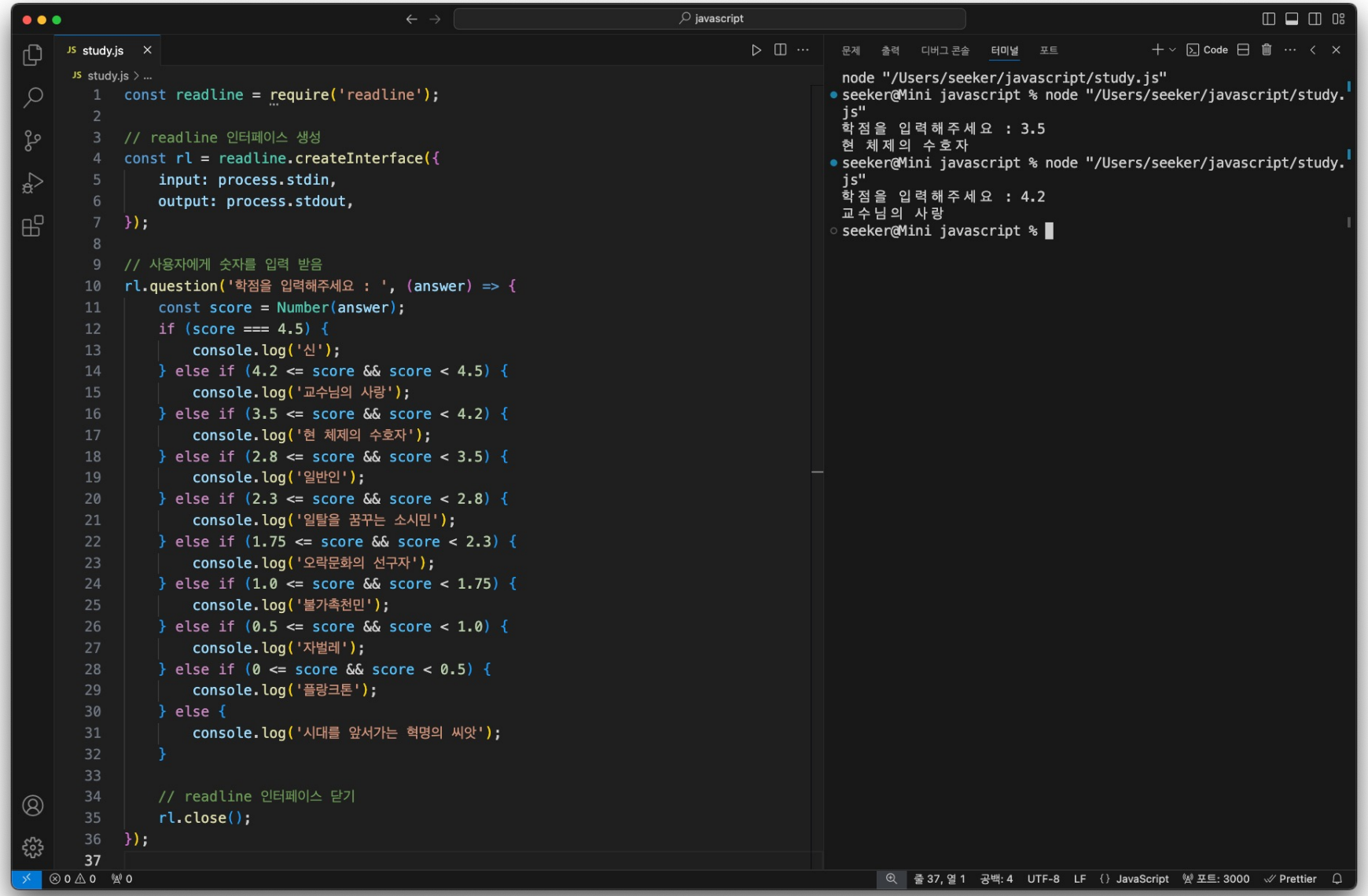
Part 3

짝수와 홀수 구분하기



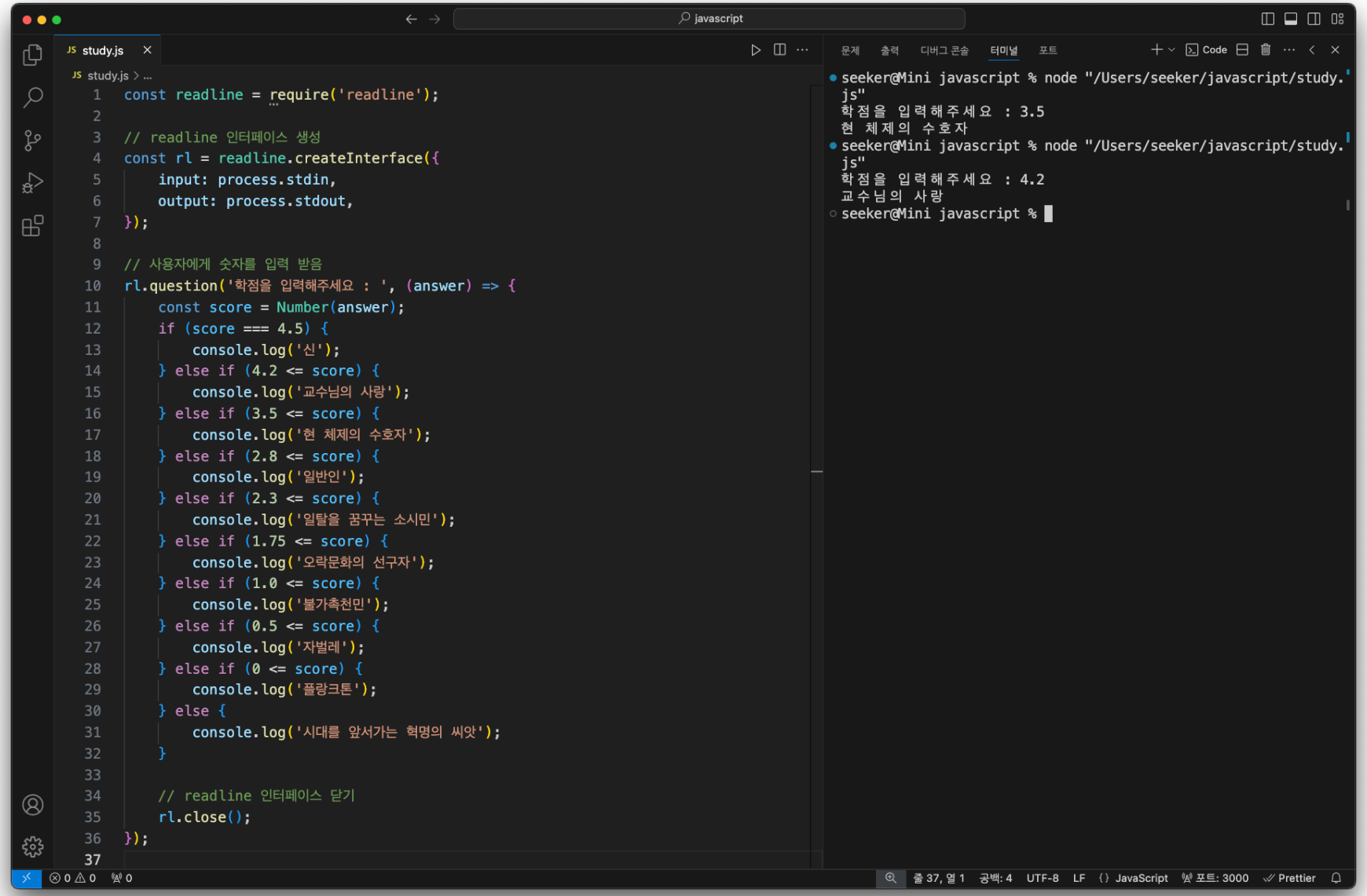
Part 3

학점을 기반으로 별명 붙여주기



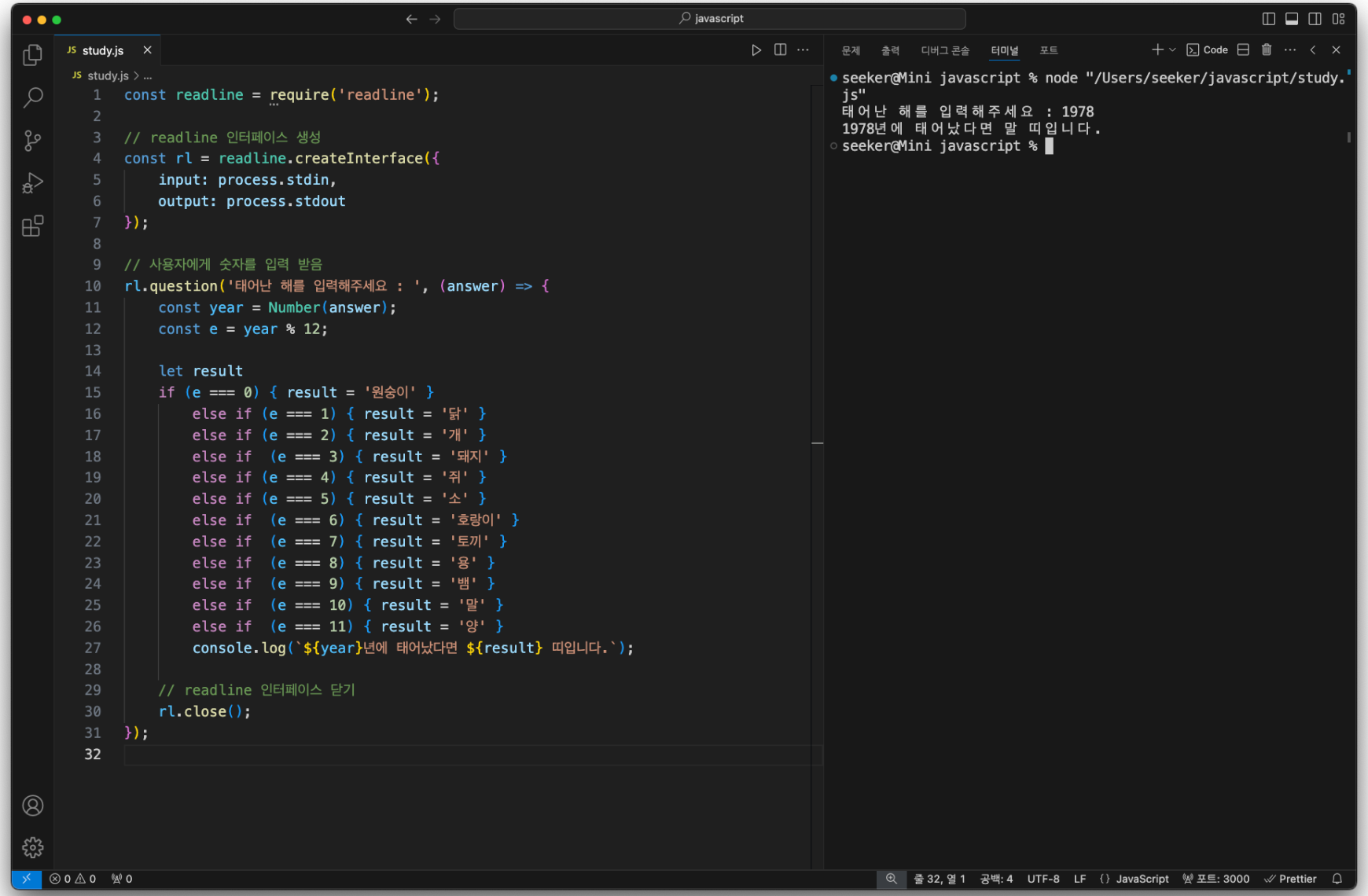
Part 3

학점을 기반으로 별명 붙여주기



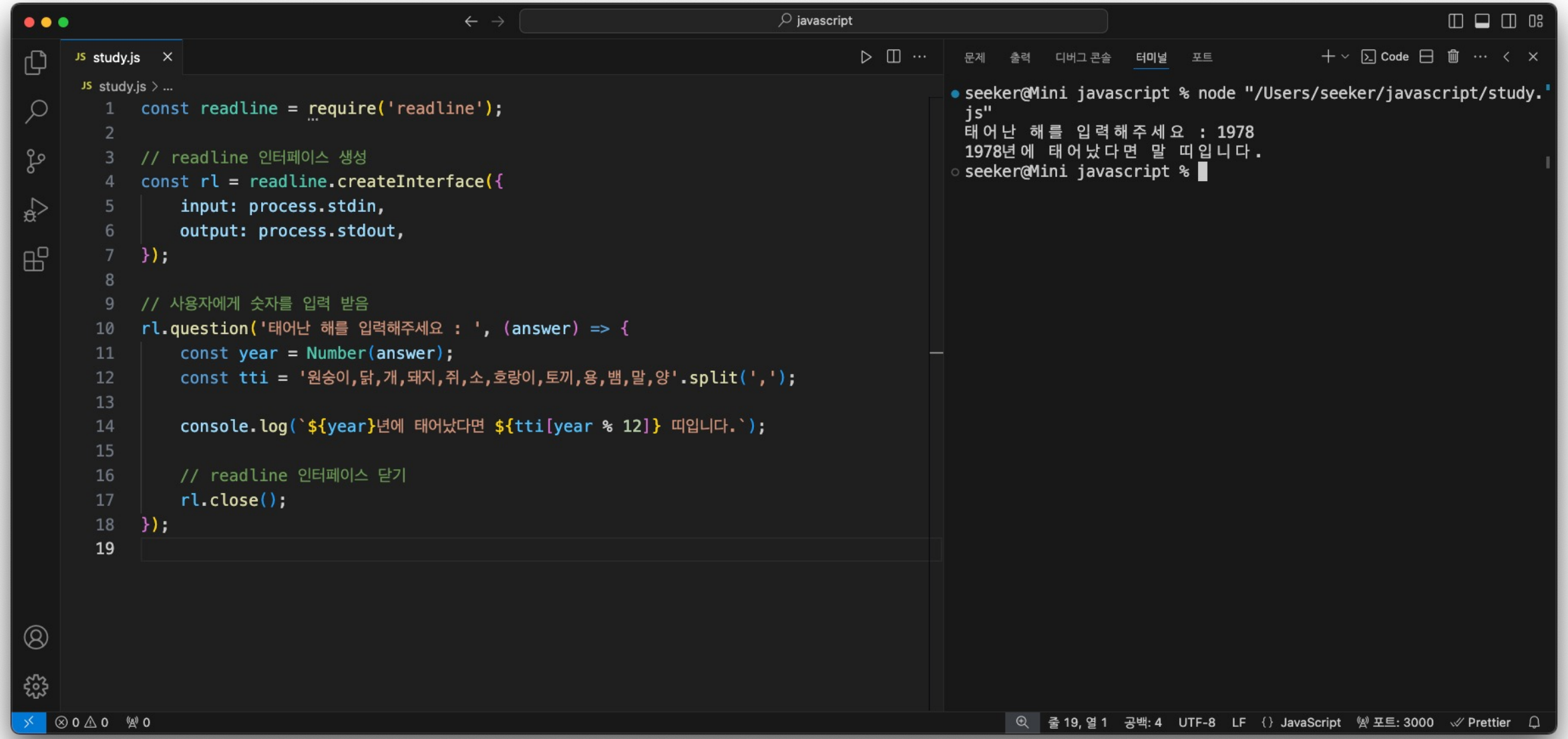
Part 3

태어난 연도를 입력 받아서 띠 출력하기

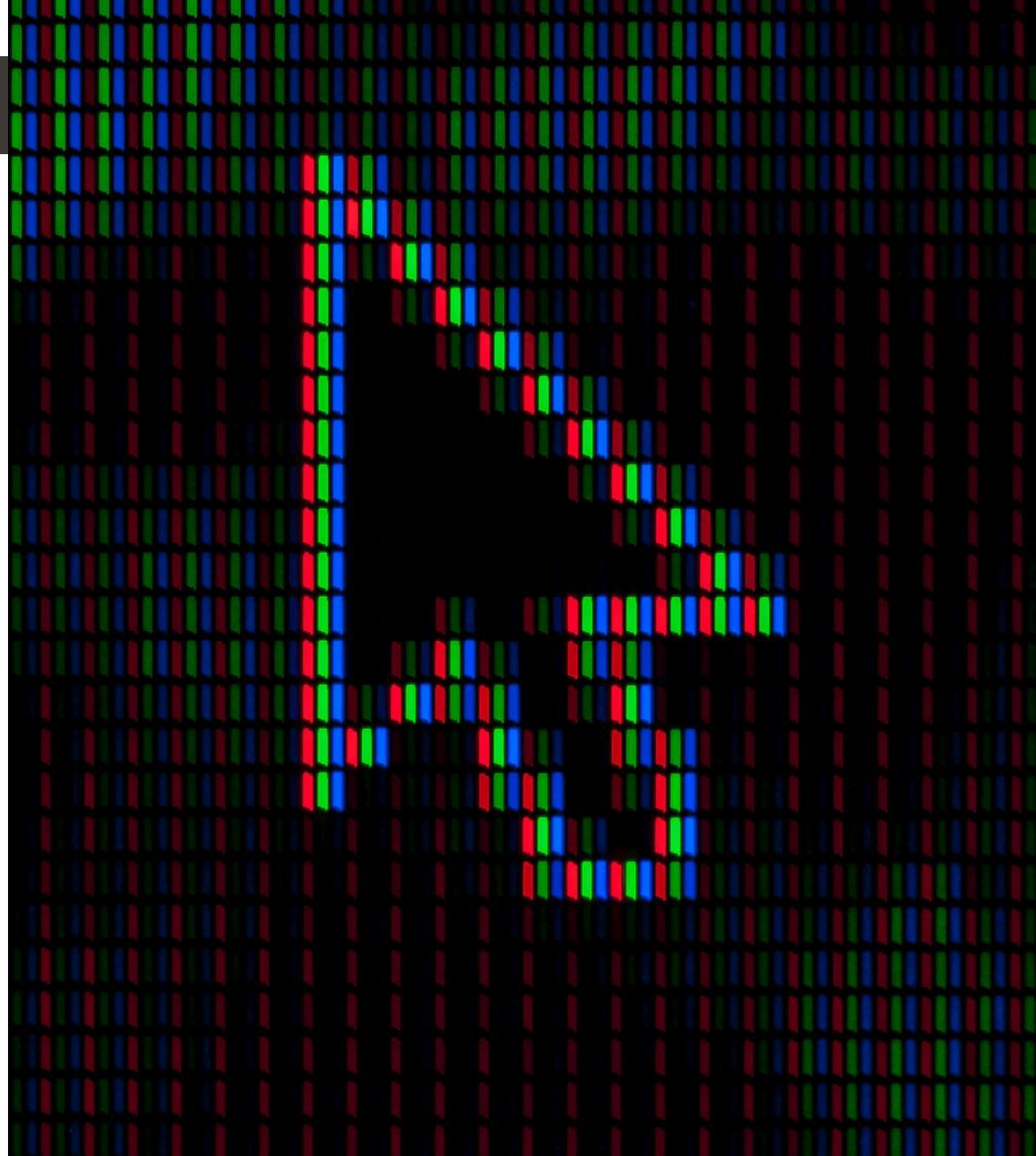


Part 3

태어난 연도를 입력 받아서 띠 출력하기



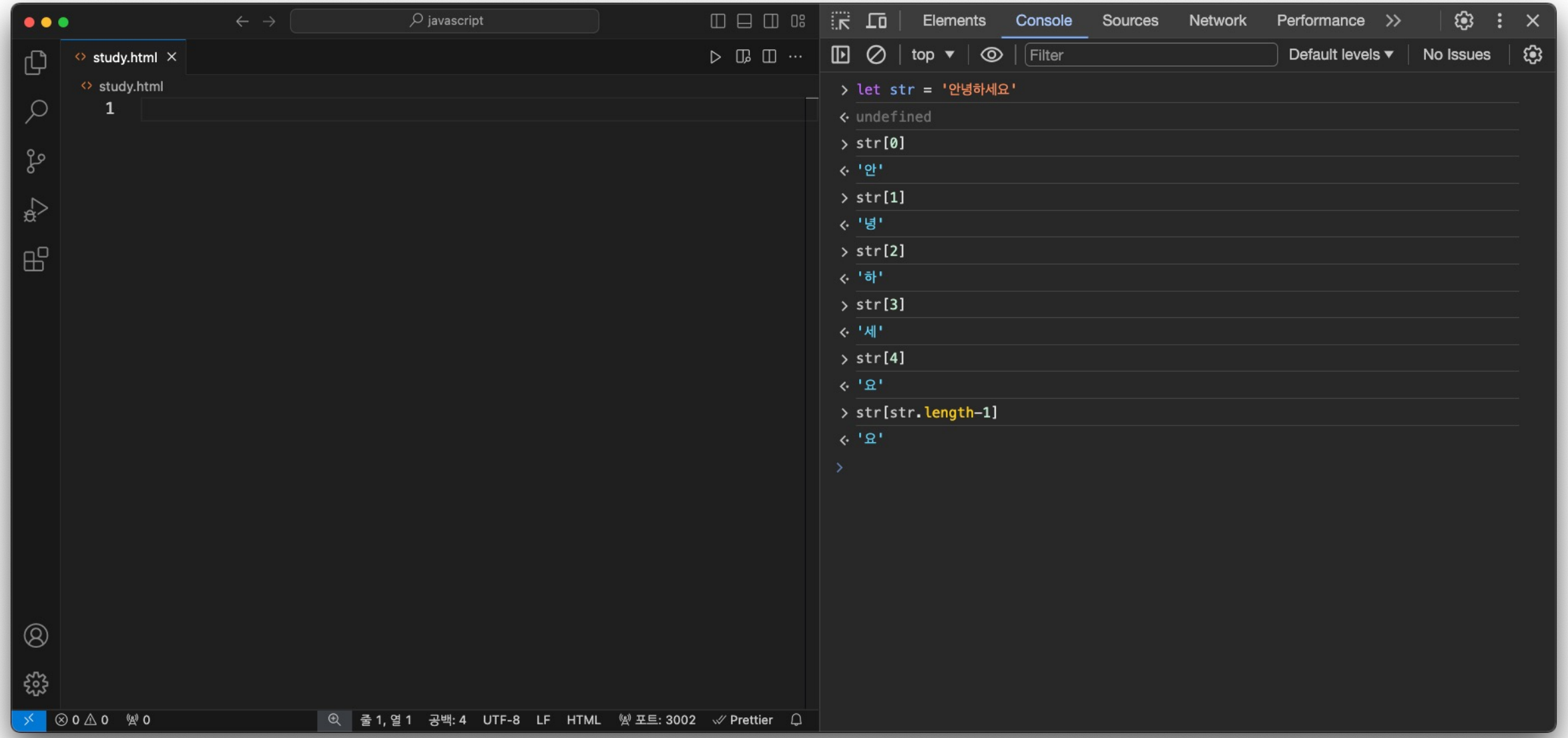
목차 a table of contents



배 열

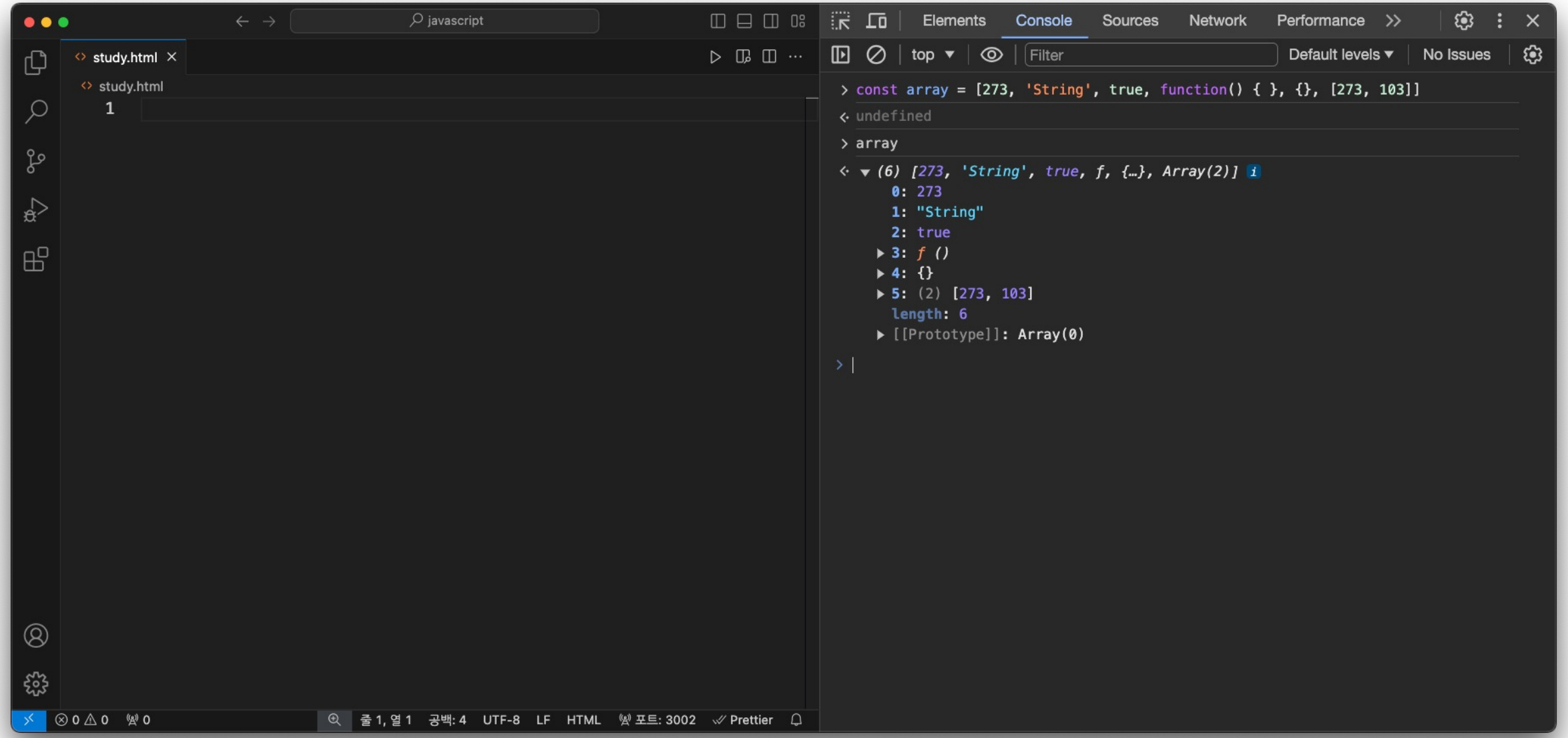
Part 4

배열


javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
> let str = '안녕하세요'
< undefined
> str[0]
< '안'
> str[1]
< '녕'
> str[2]
< '하'
> str[3]
< '세'
> str[4]
< '요'
> str[str.length-1]
< '요'
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

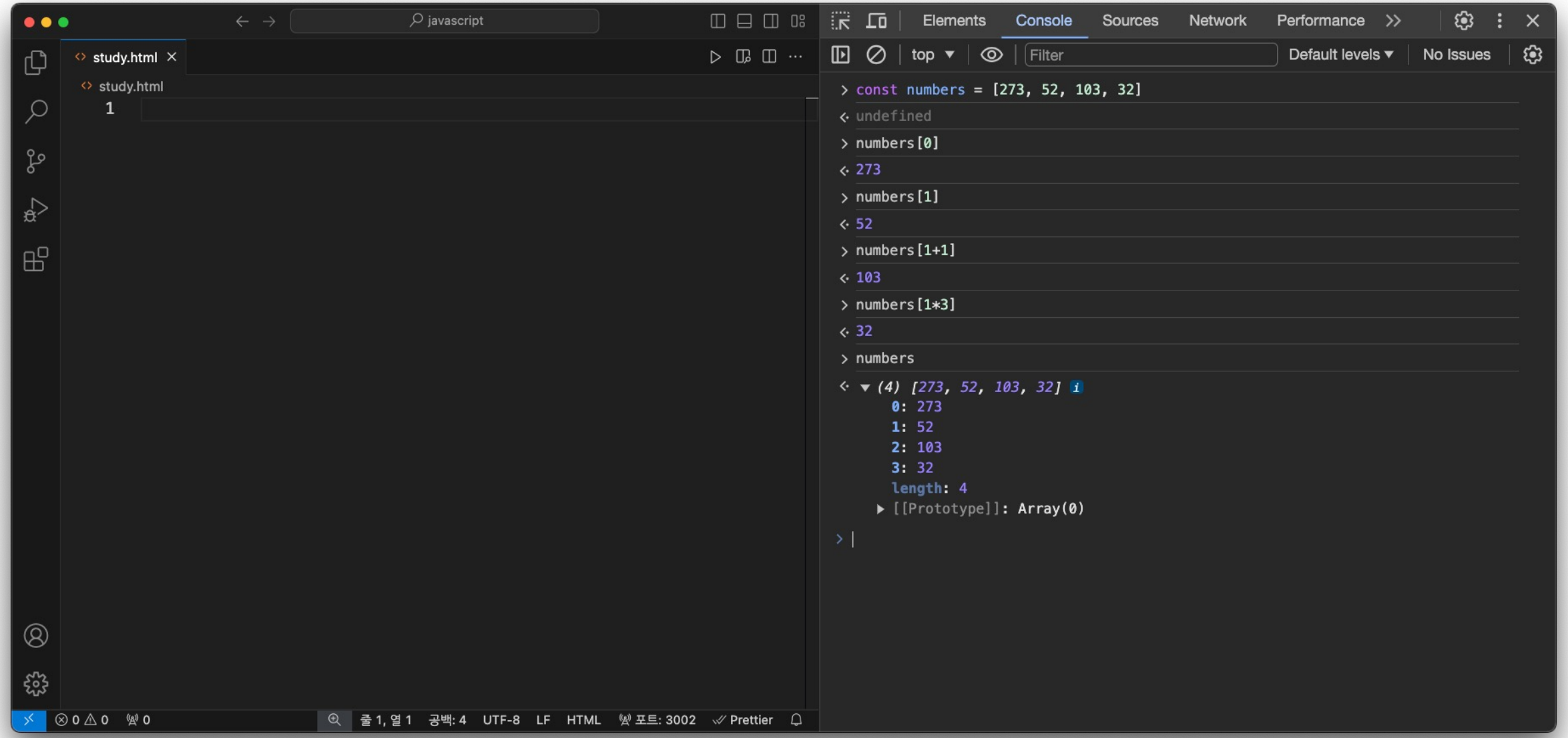
Part 4

배열 만들기


javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
> const array = [273, 'String', true, function() { }, {}, [273, 103]]
undefined
> array
(6) [273, 'String', true, f, {…}, Array(2)]
0: 273
1: "String"
2: true
3: f ()
4: {}
5: (2) [273, 103]
length: 6
[[Prototype]]: Array(0)
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

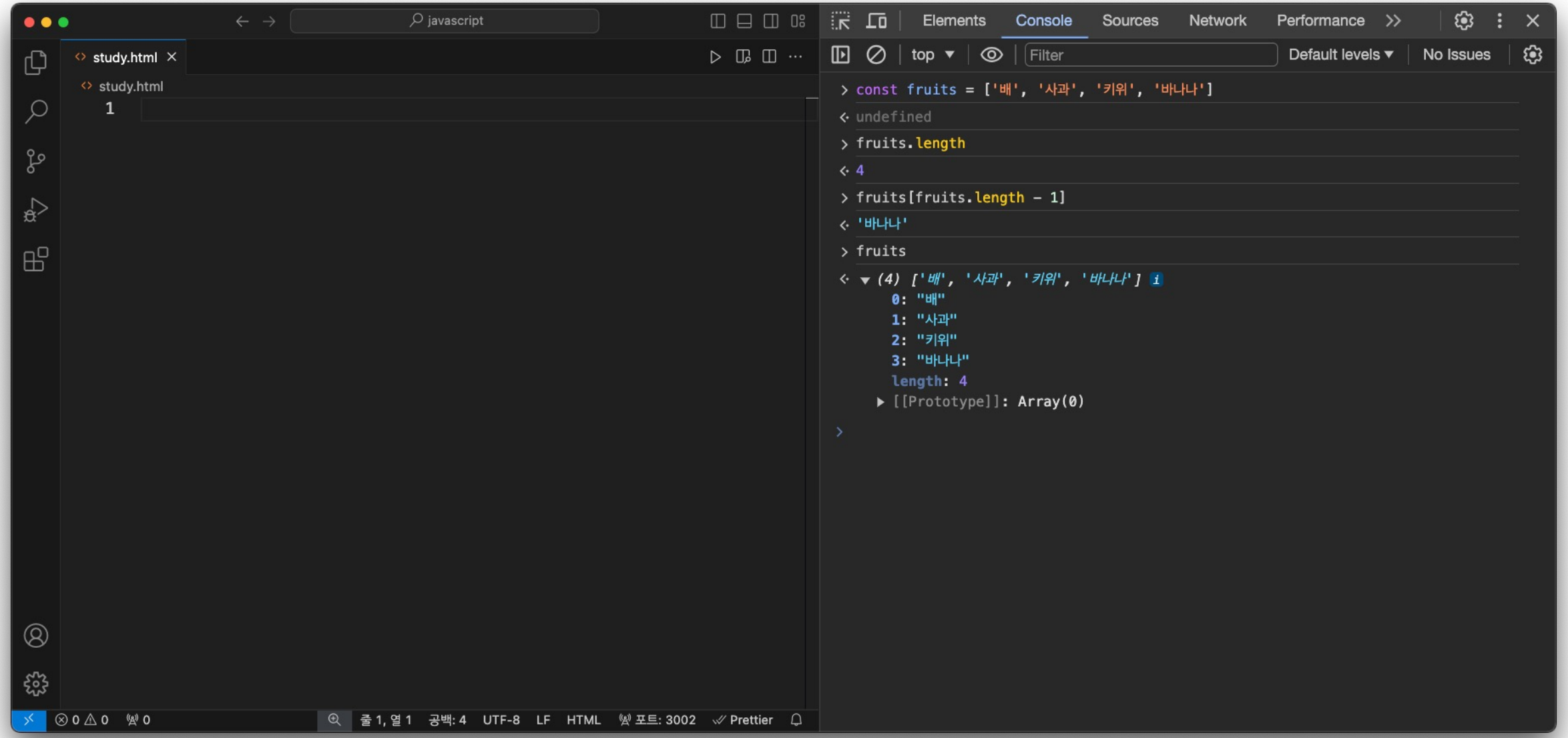
Part 4

배열 요소에 접근하기

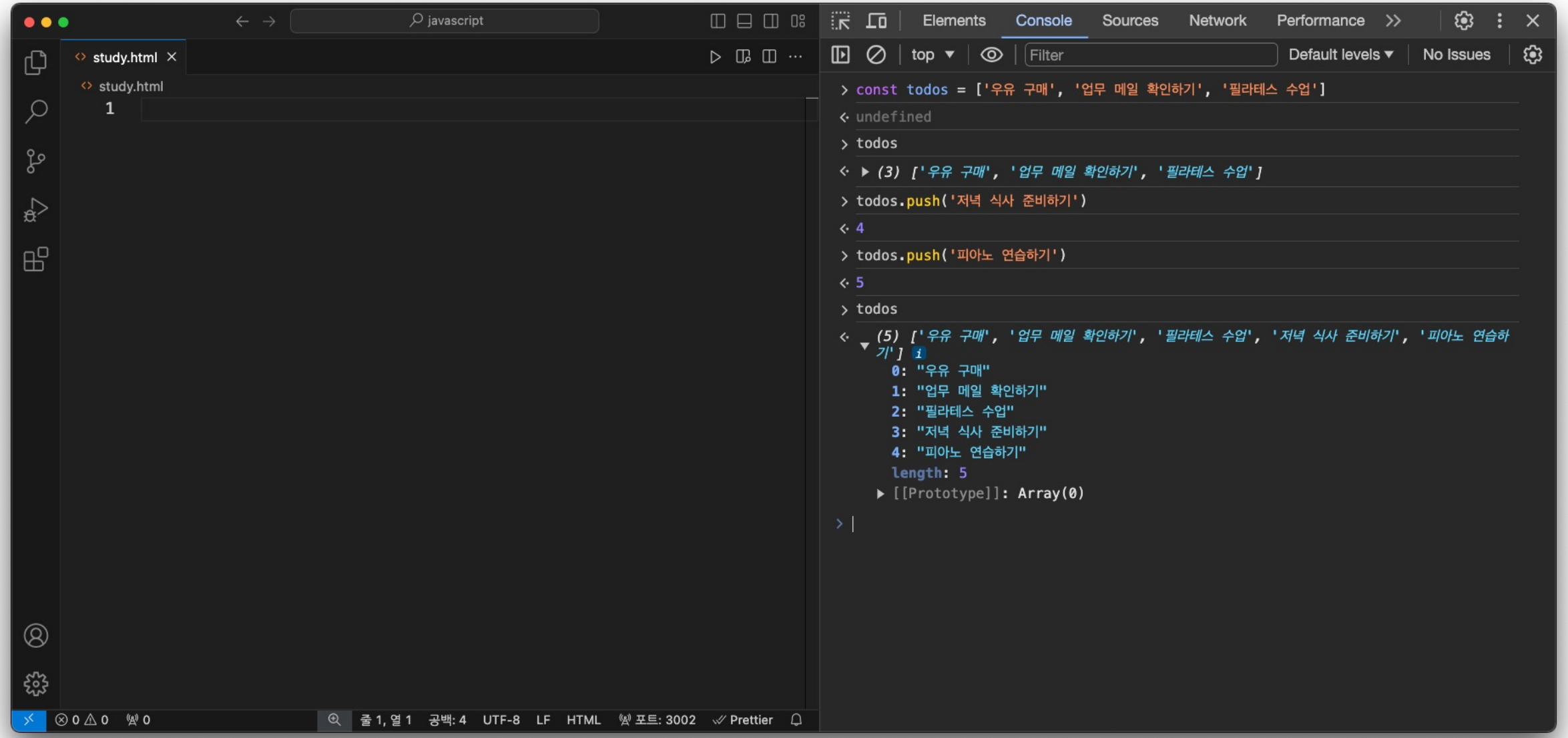


Part 4

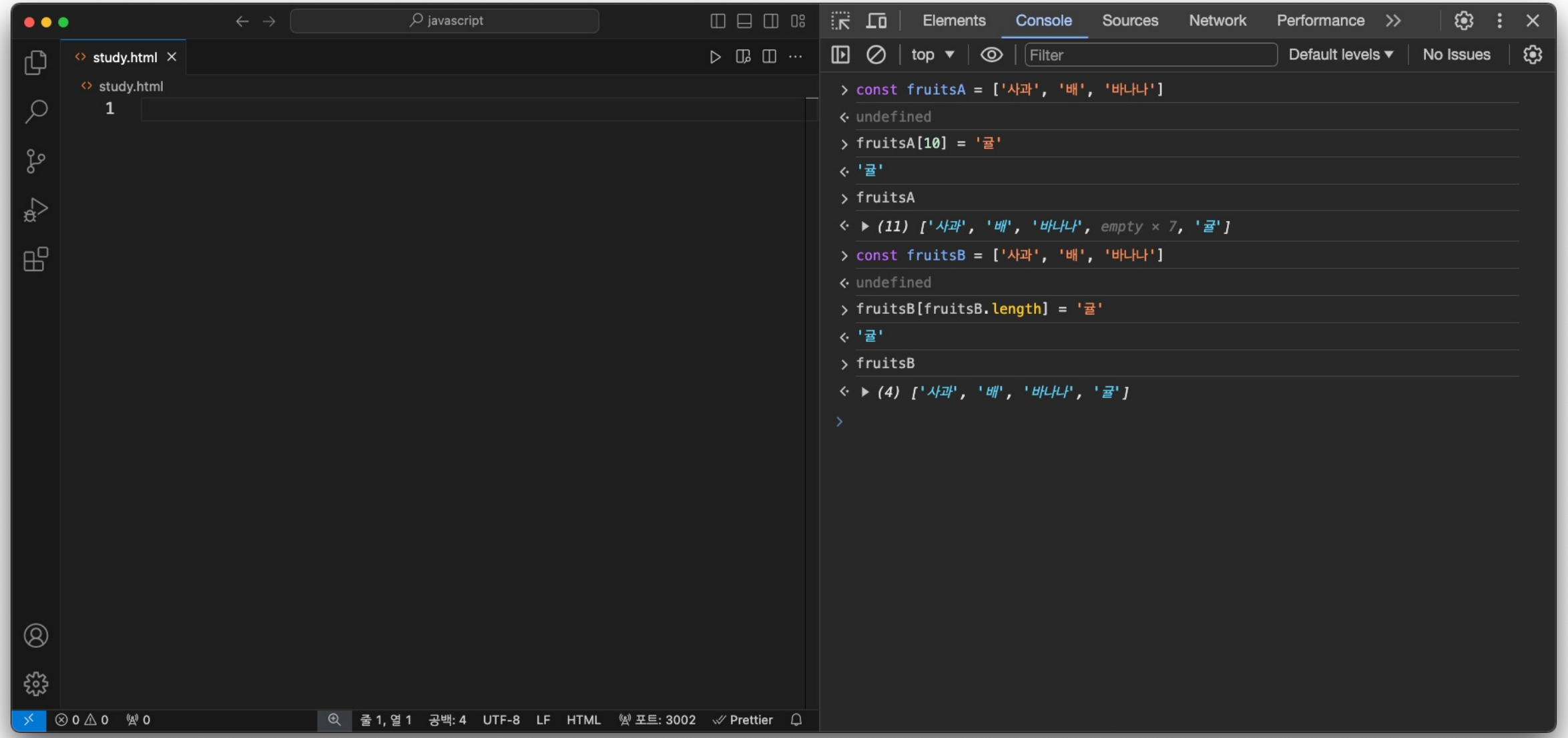
배열 요소 개수 확인하기



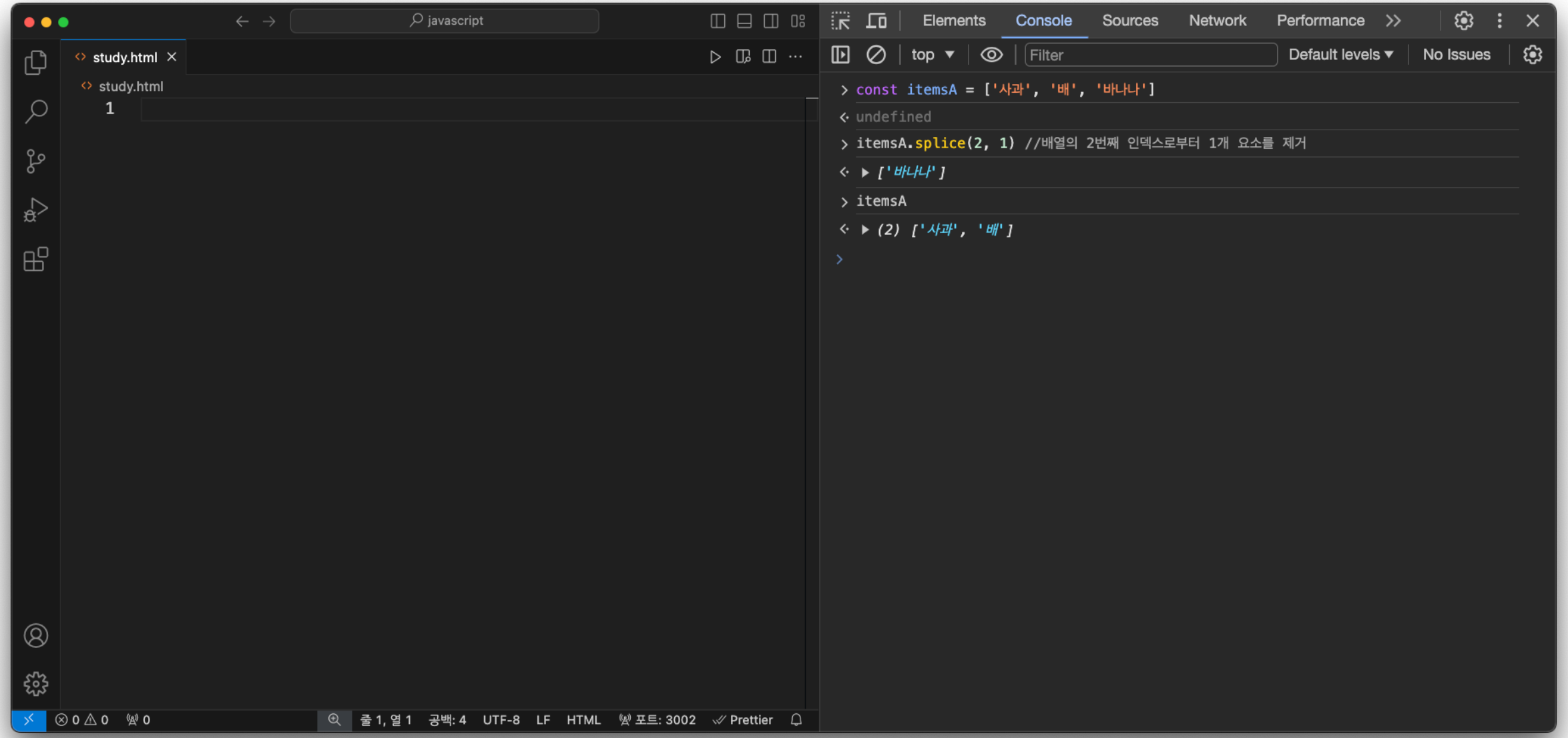
Part 4 배열 뒷부분에 요소 추가하기 > push 메소드 사용



Part 4 배열 뒷부분에 요소 추가하기 > 인덱스 사용

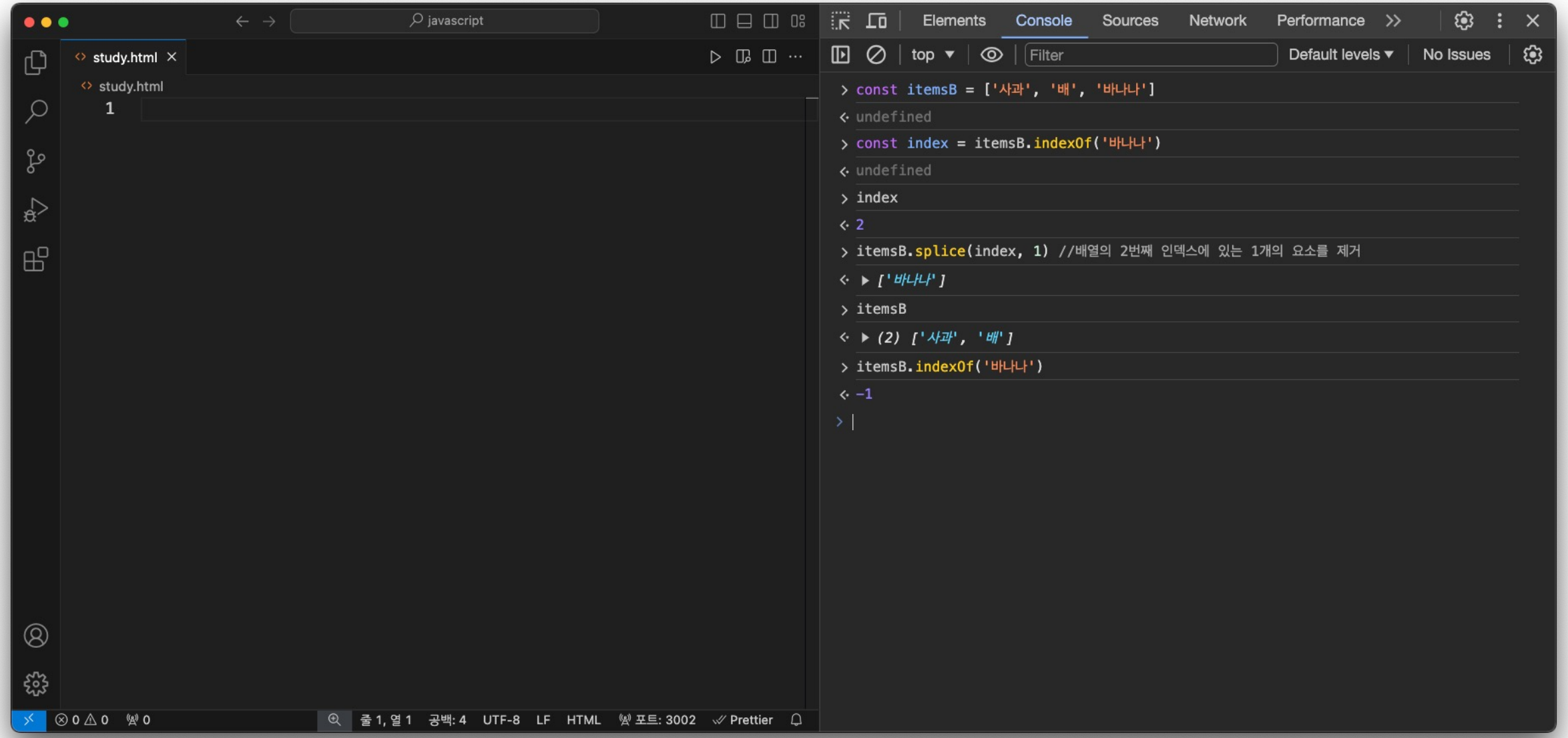


Part 4 배열 요소 제거하기 > 인덱스 사용



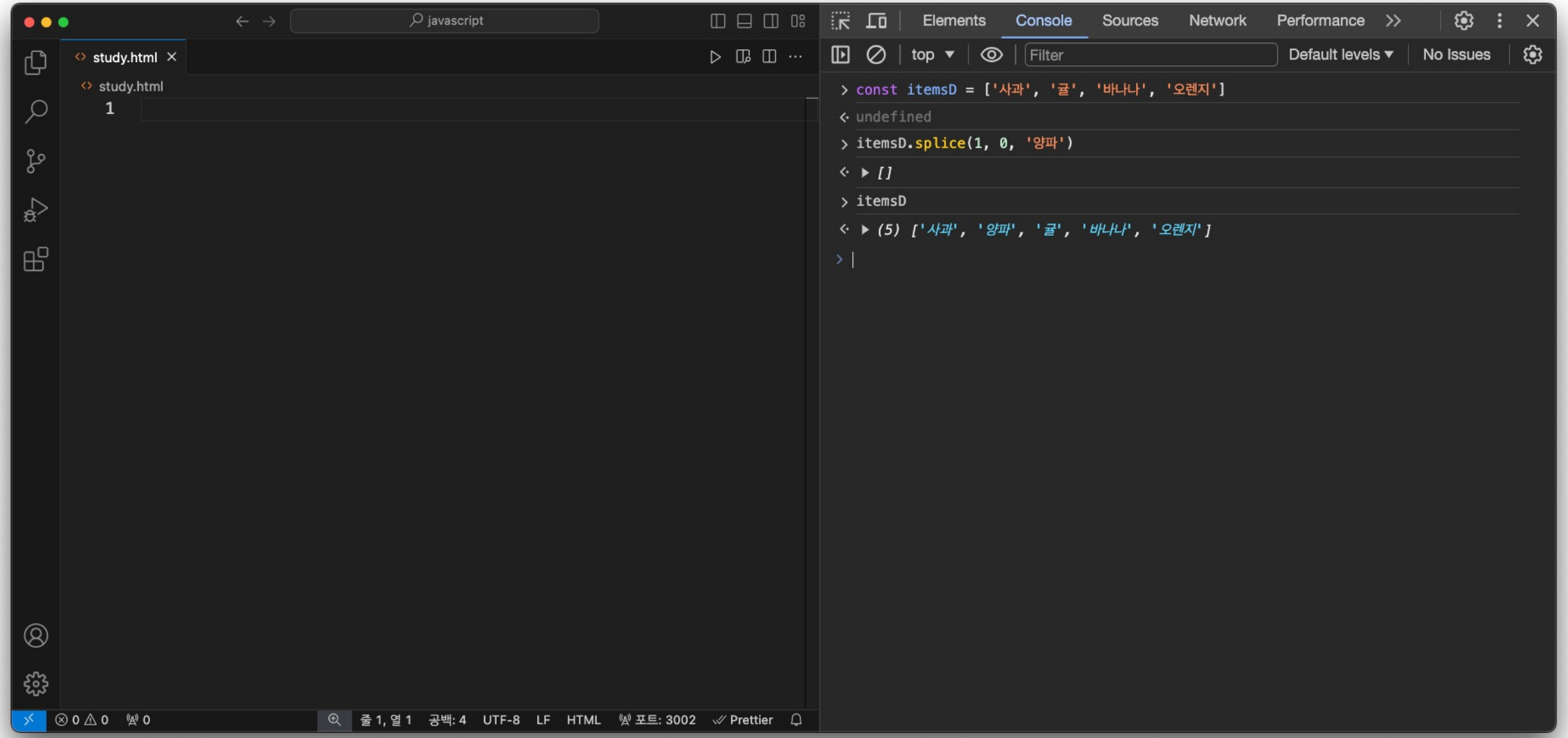
Part 4

배열 요소 제거하기 > 값 사용



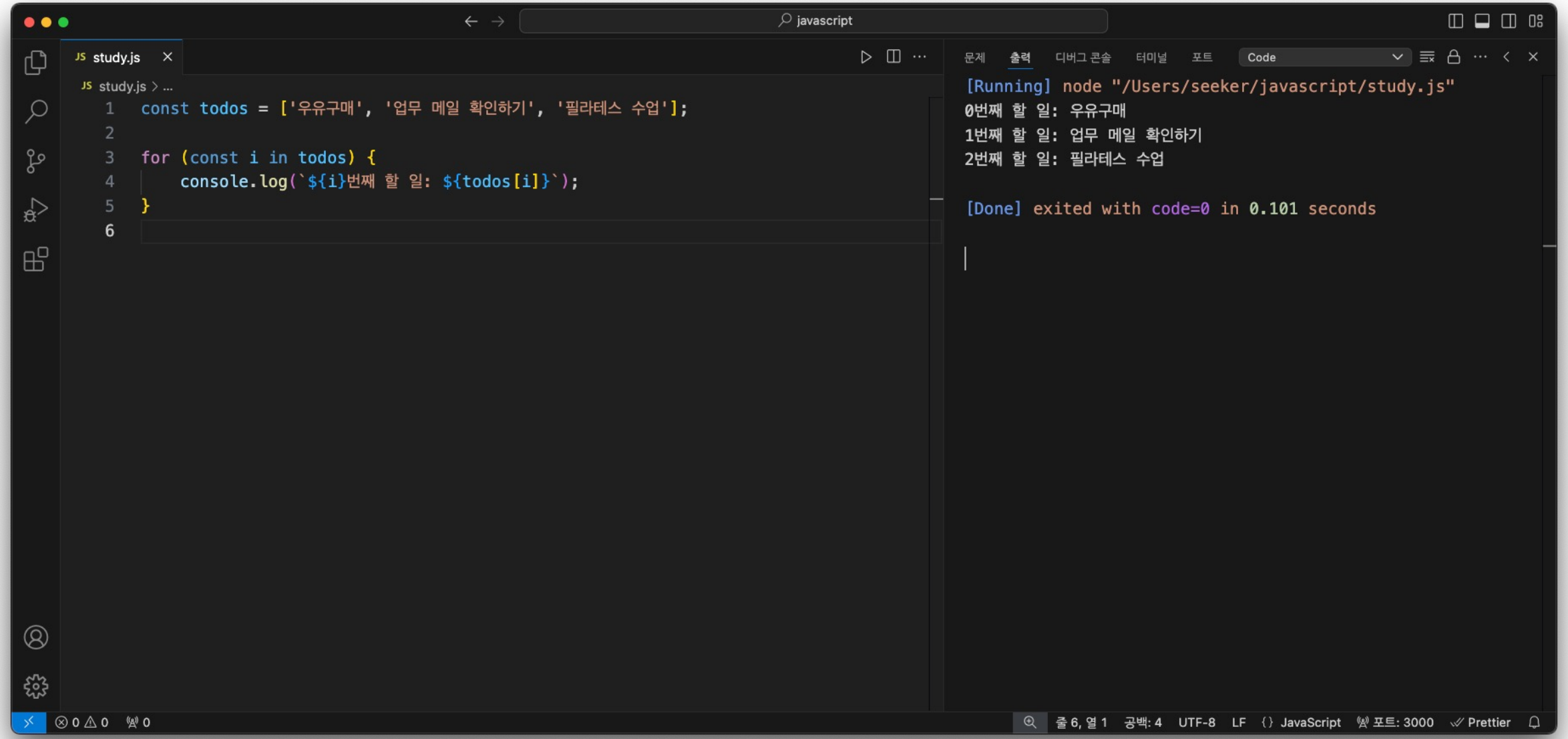
Part 4

배열의 특정 위치에 요소 추가하기

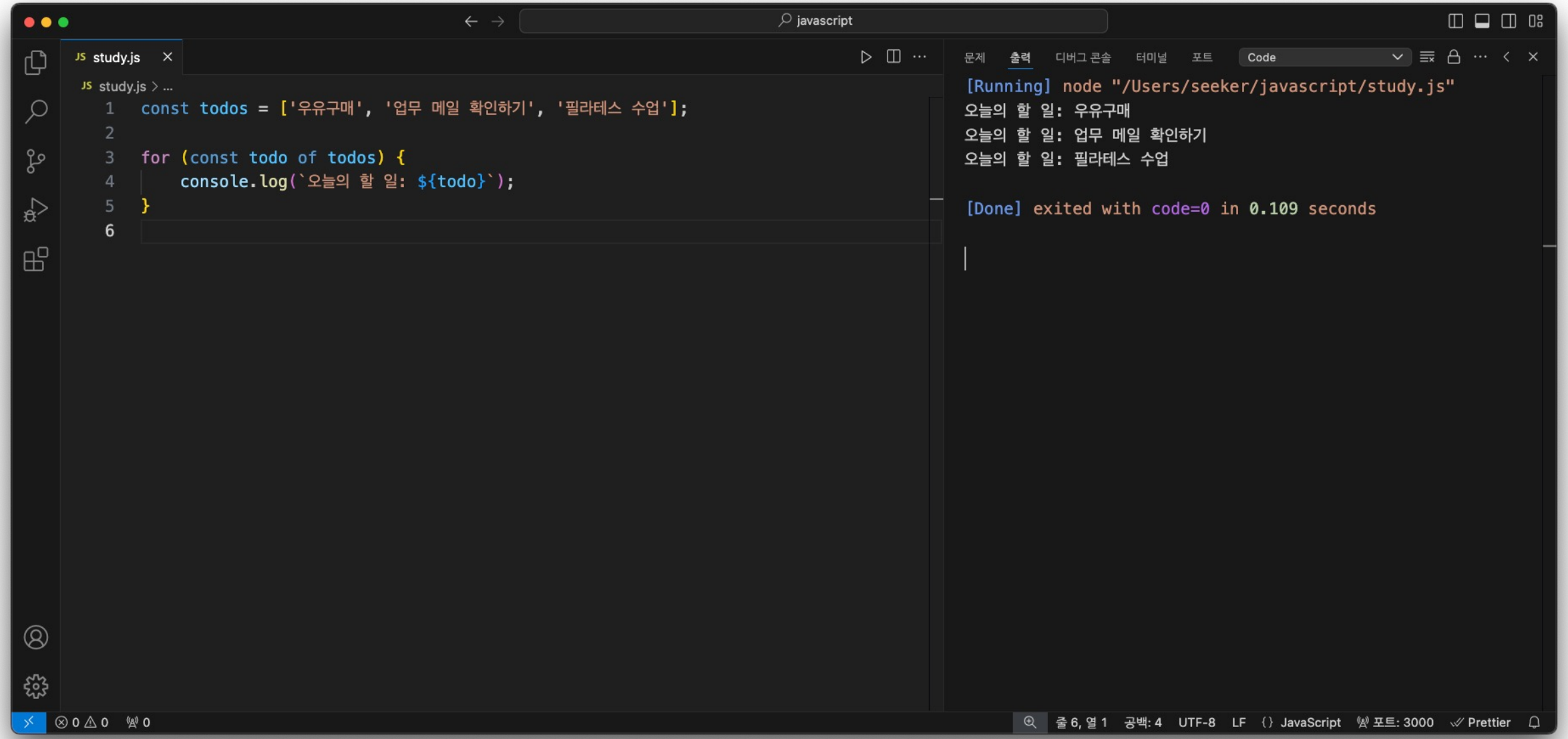


반 복 문

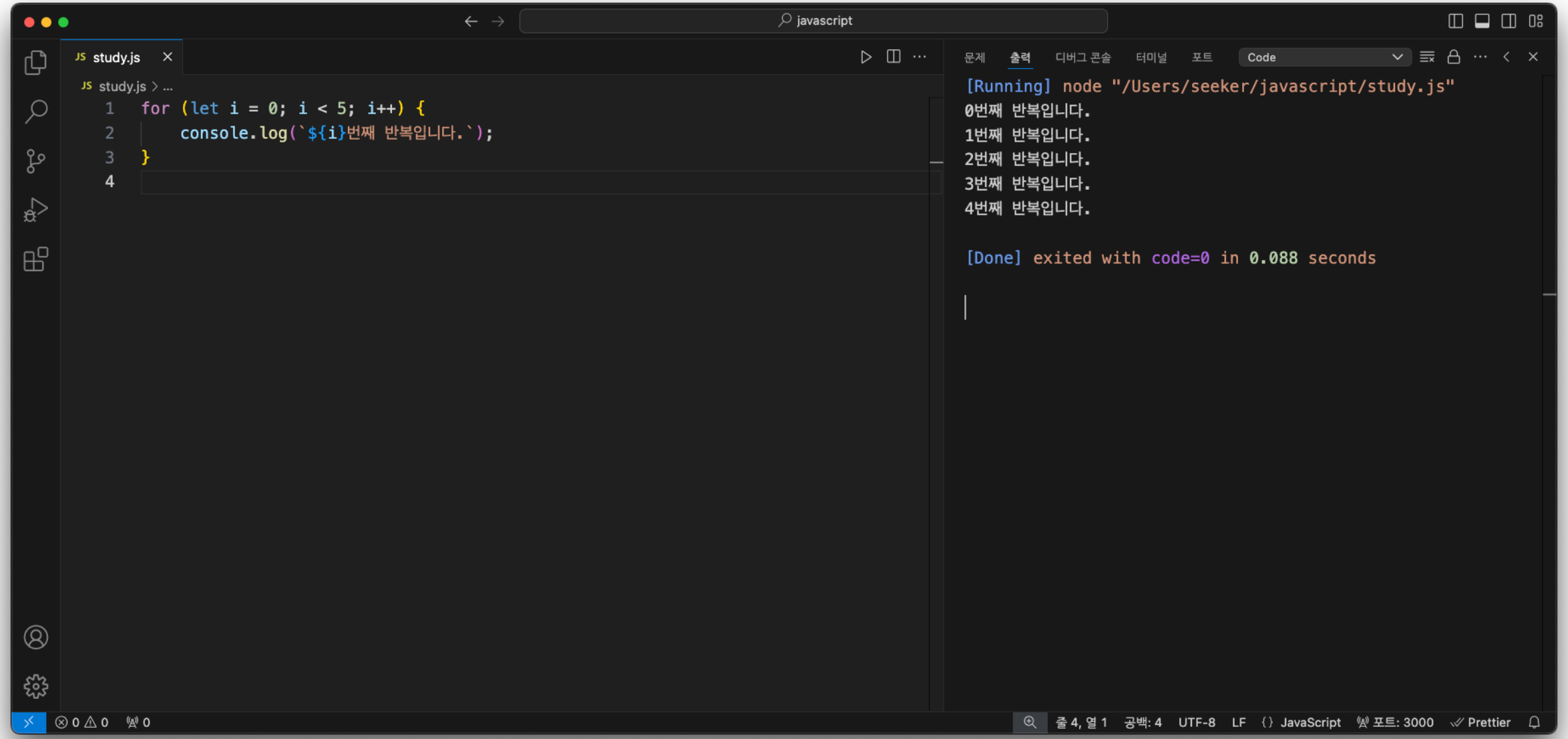
Part 4 for in 반복문



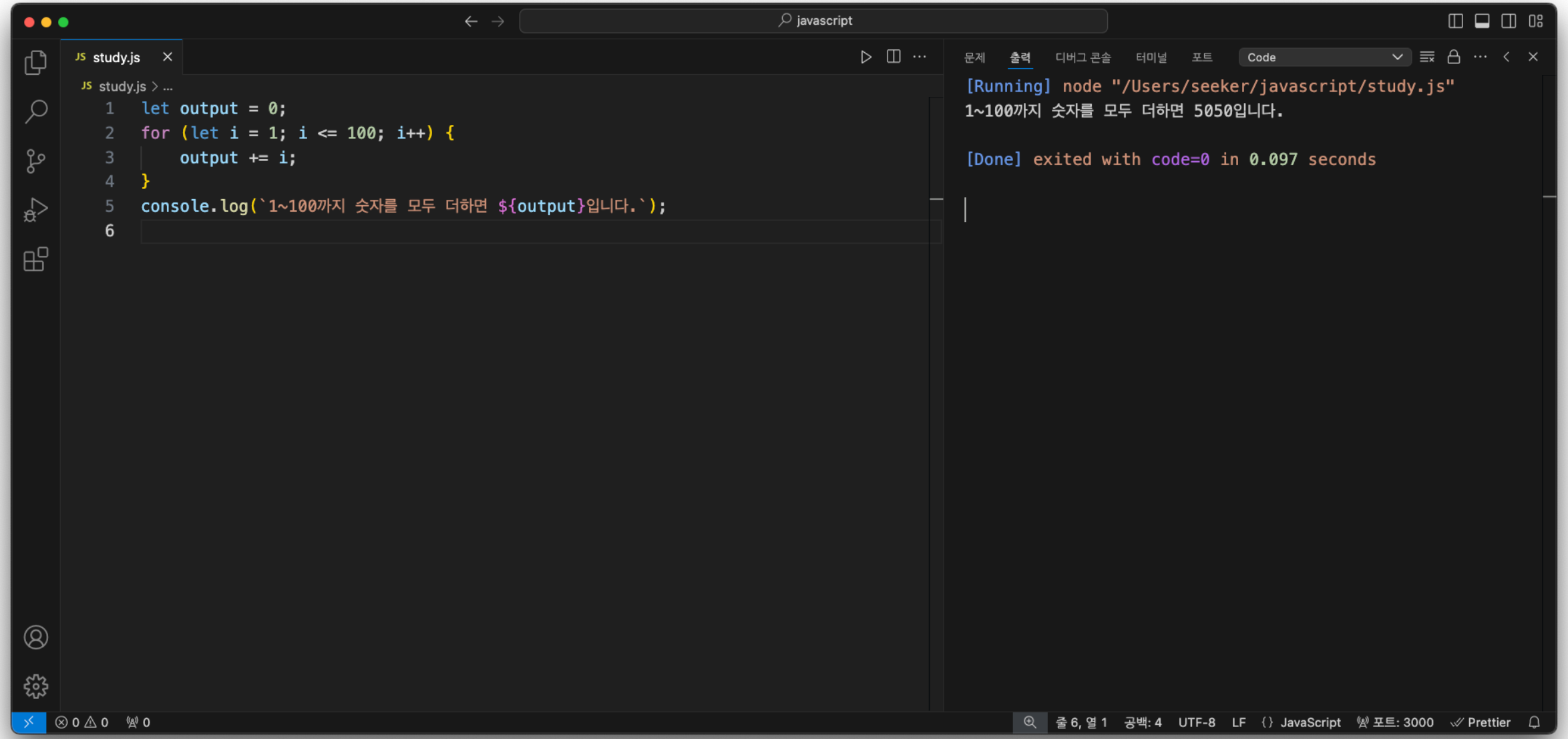
Part 4 for of 반복문



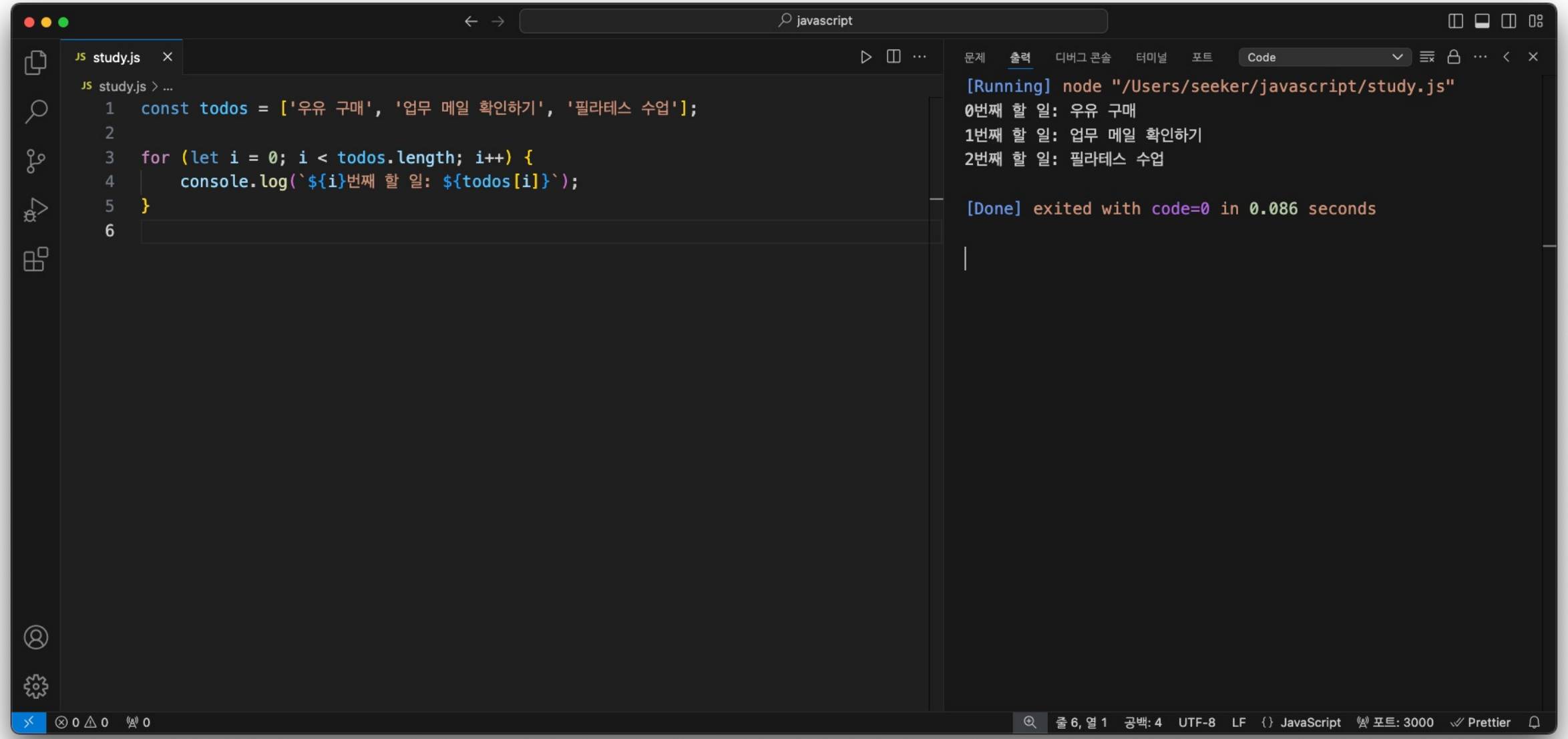
Part 4 for 반복문



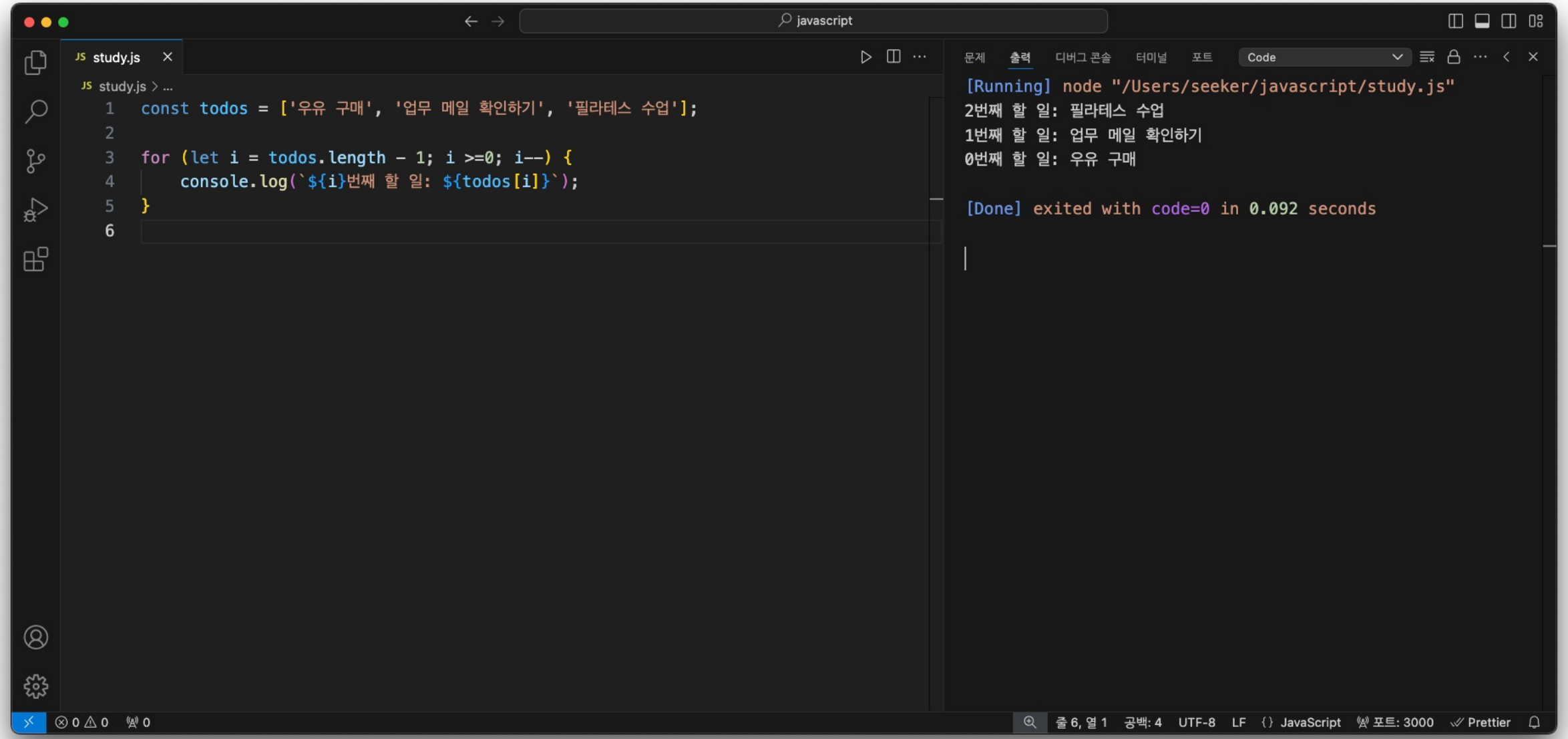
Part 4 for 반복문



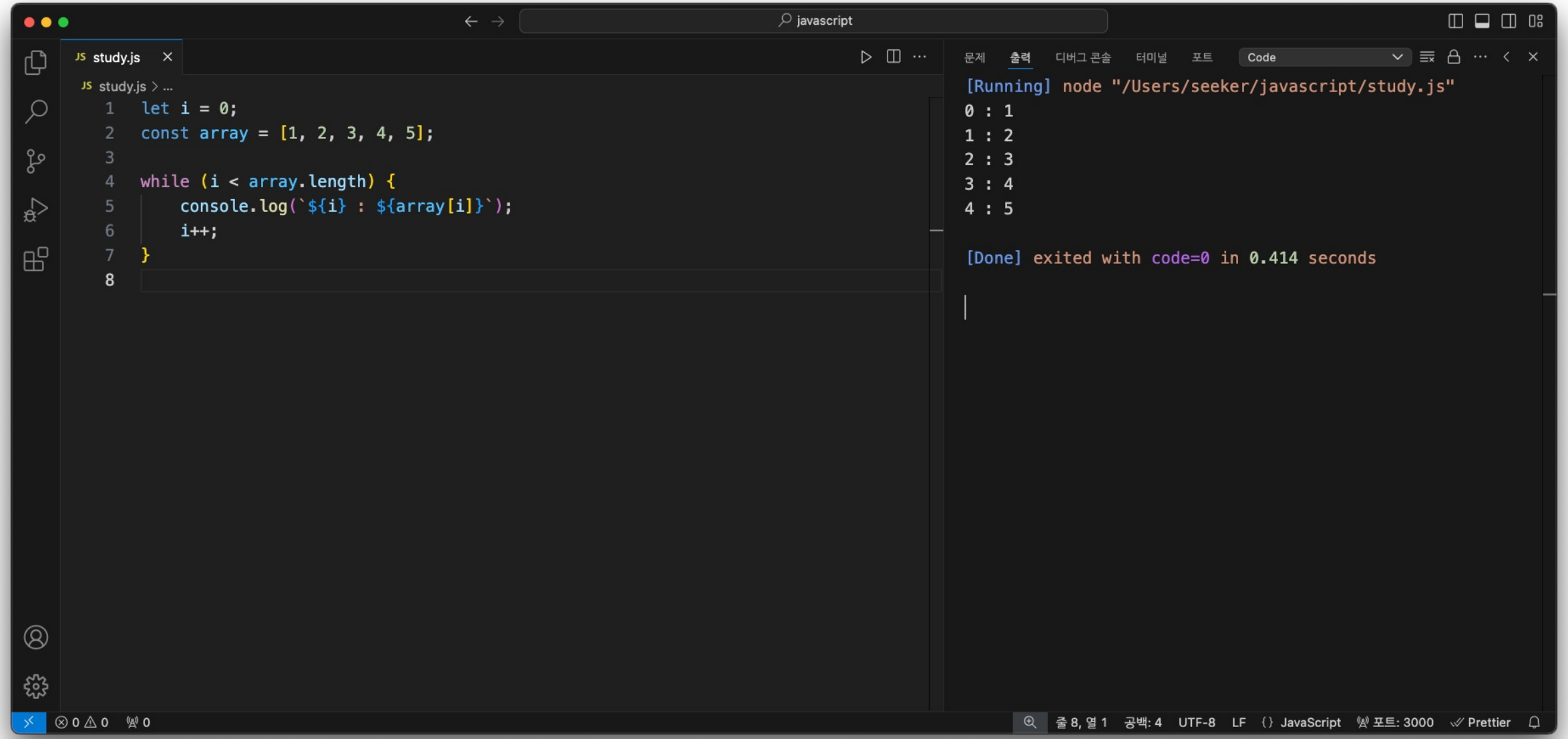
Part 4 for 반복문



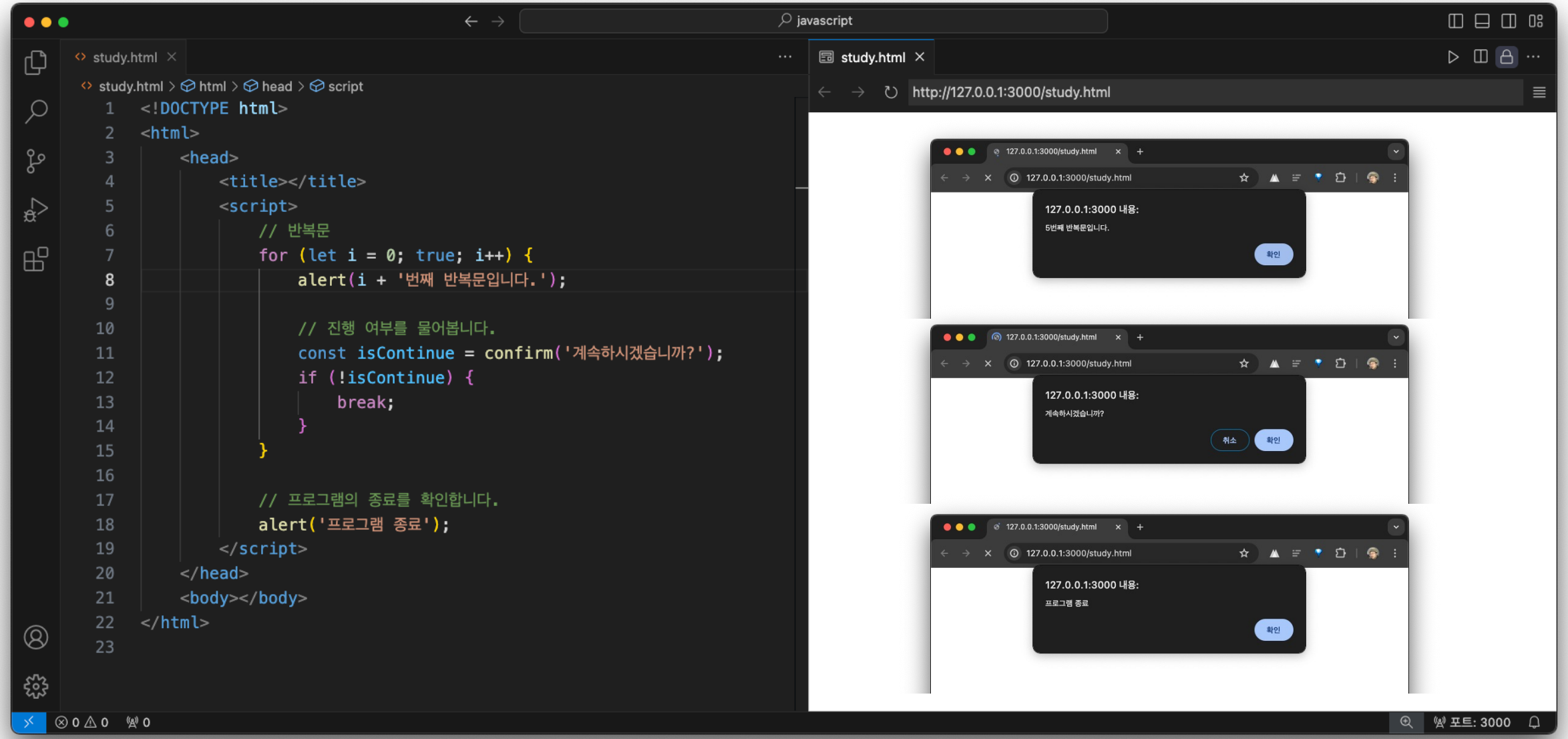
Part 4 for 반복문



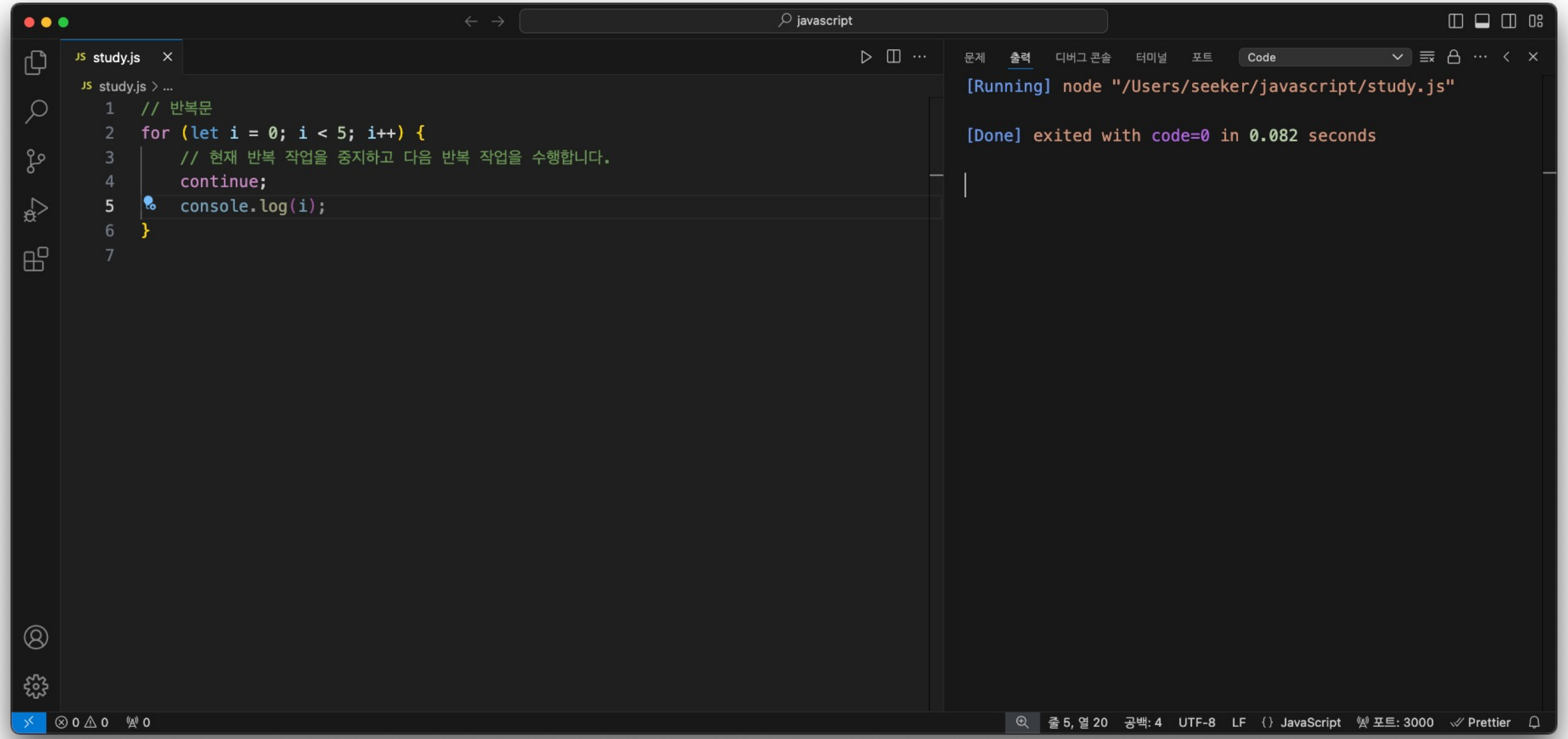
Part 4 while 반복문



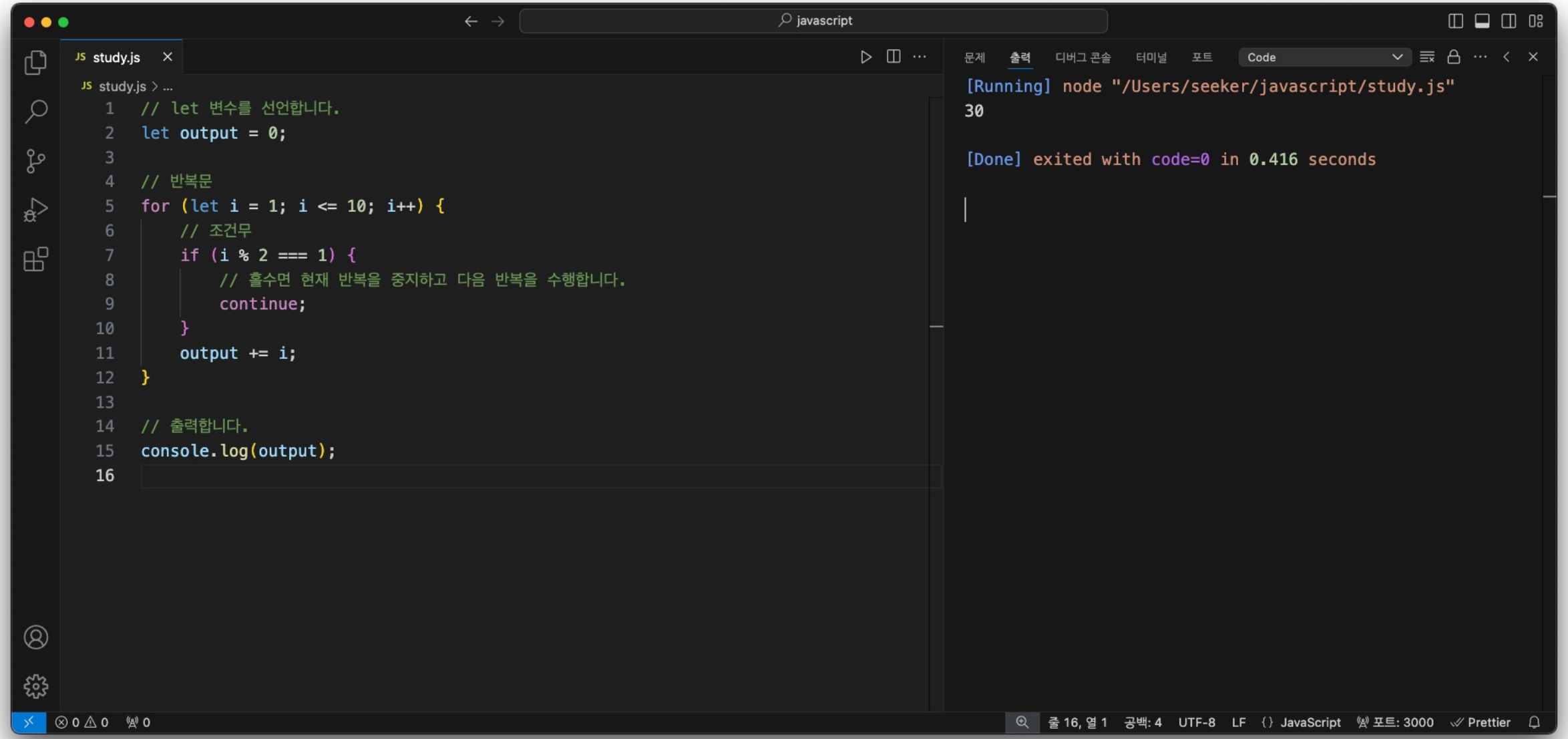
Part 4 break 키워드



Part 4 continue 키워드

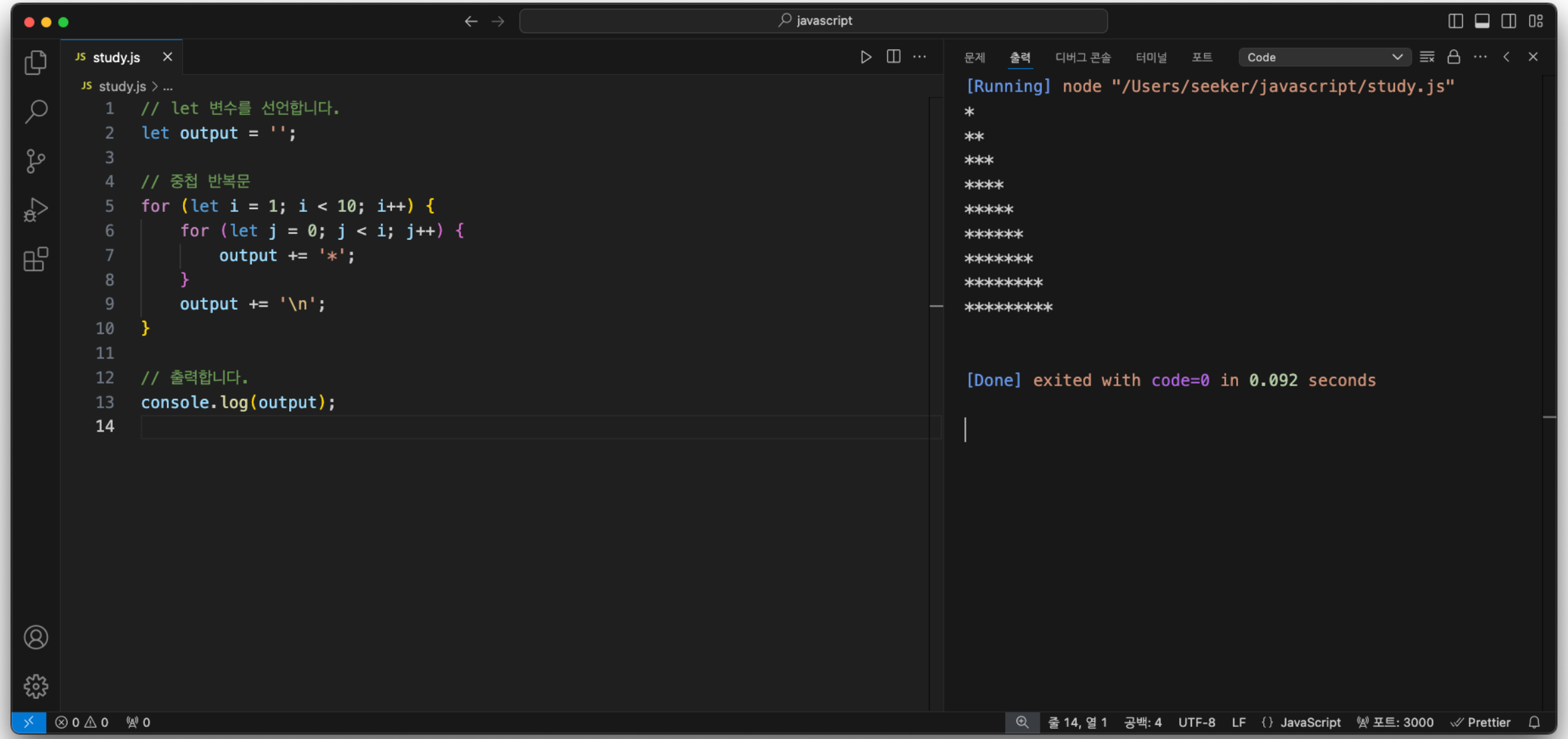


Part 4 continue 키워드



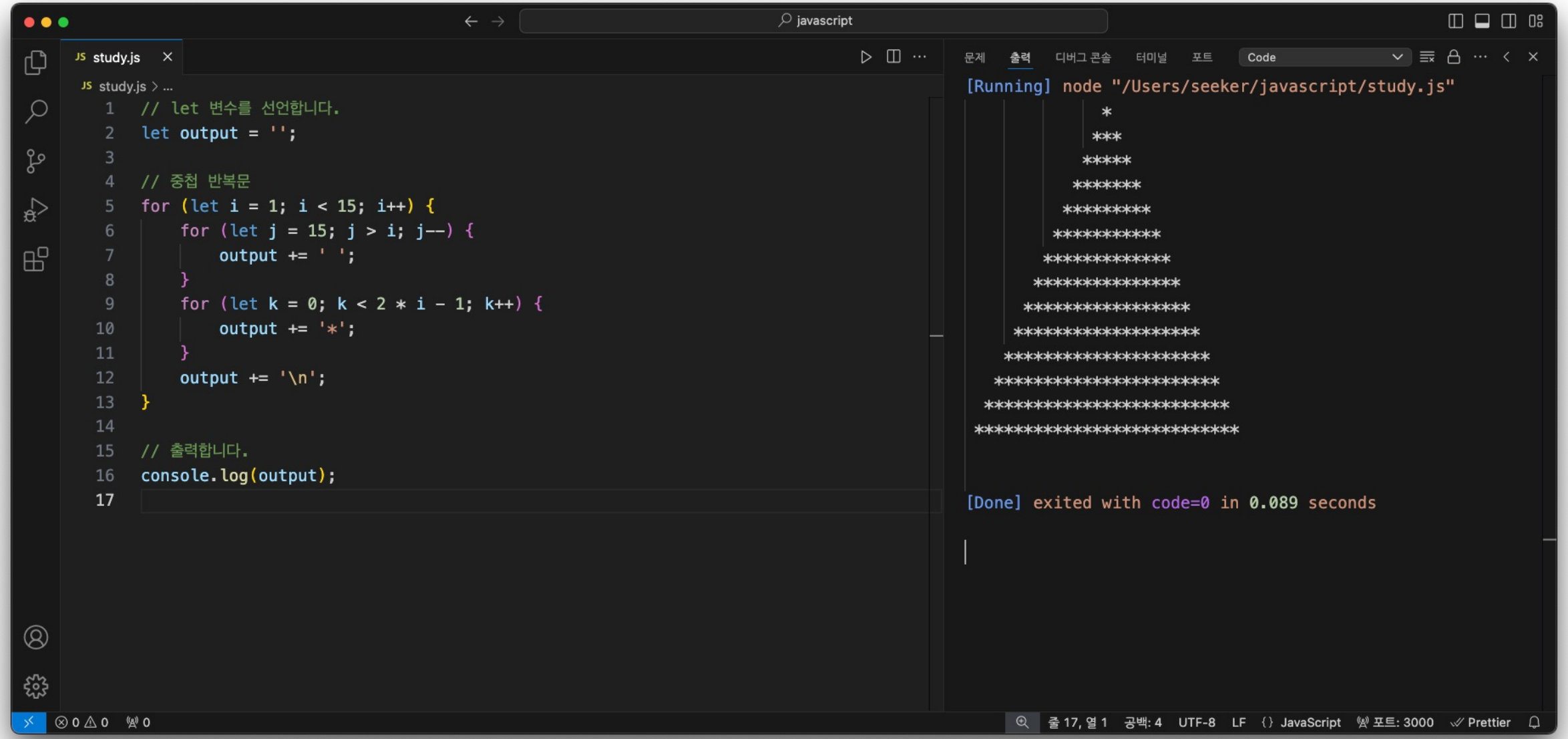
Part 4

중첩 반복문을 사용하는 피라미드

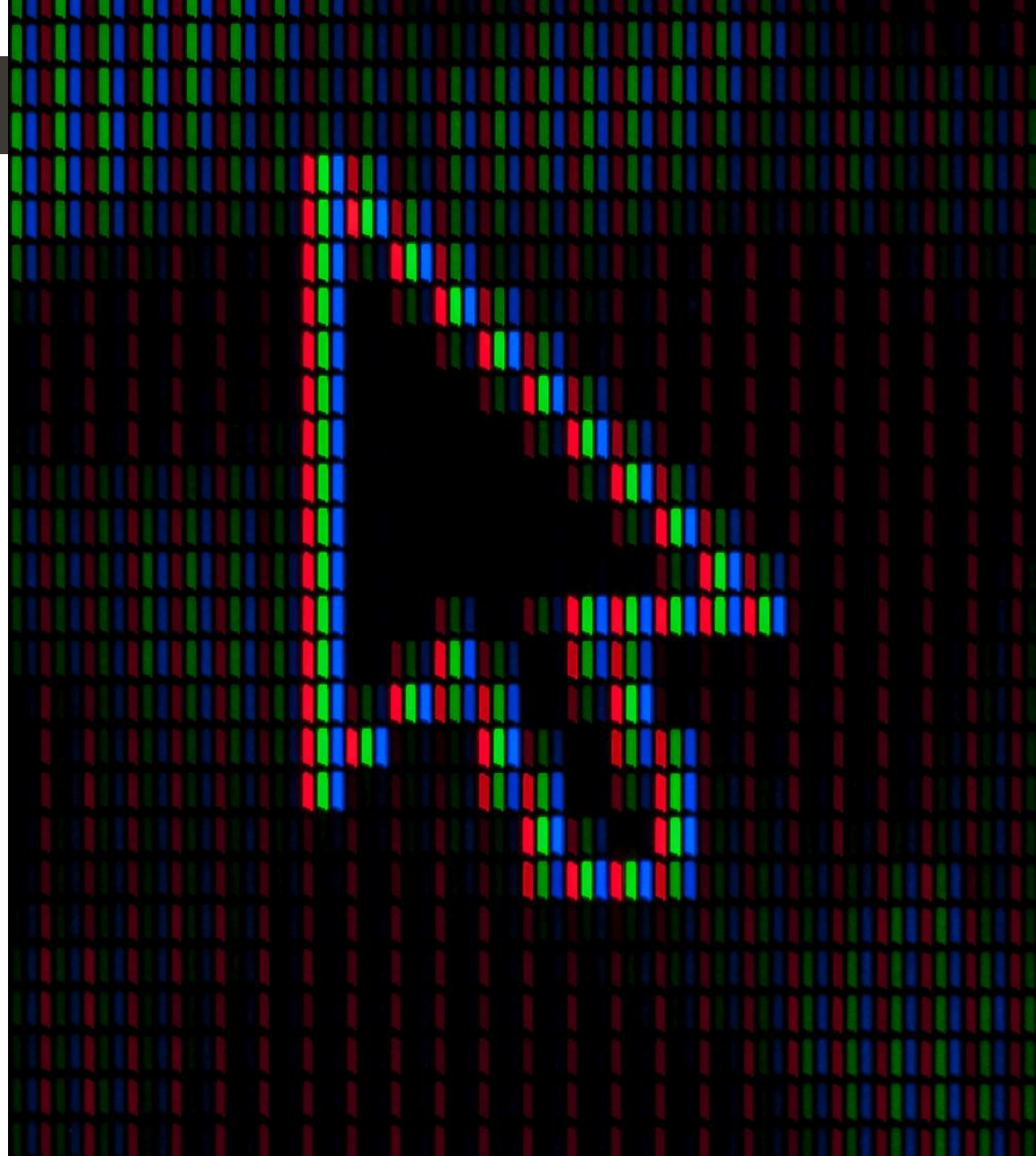


Part 4

중첩 반복문을 사용하는 피라미드



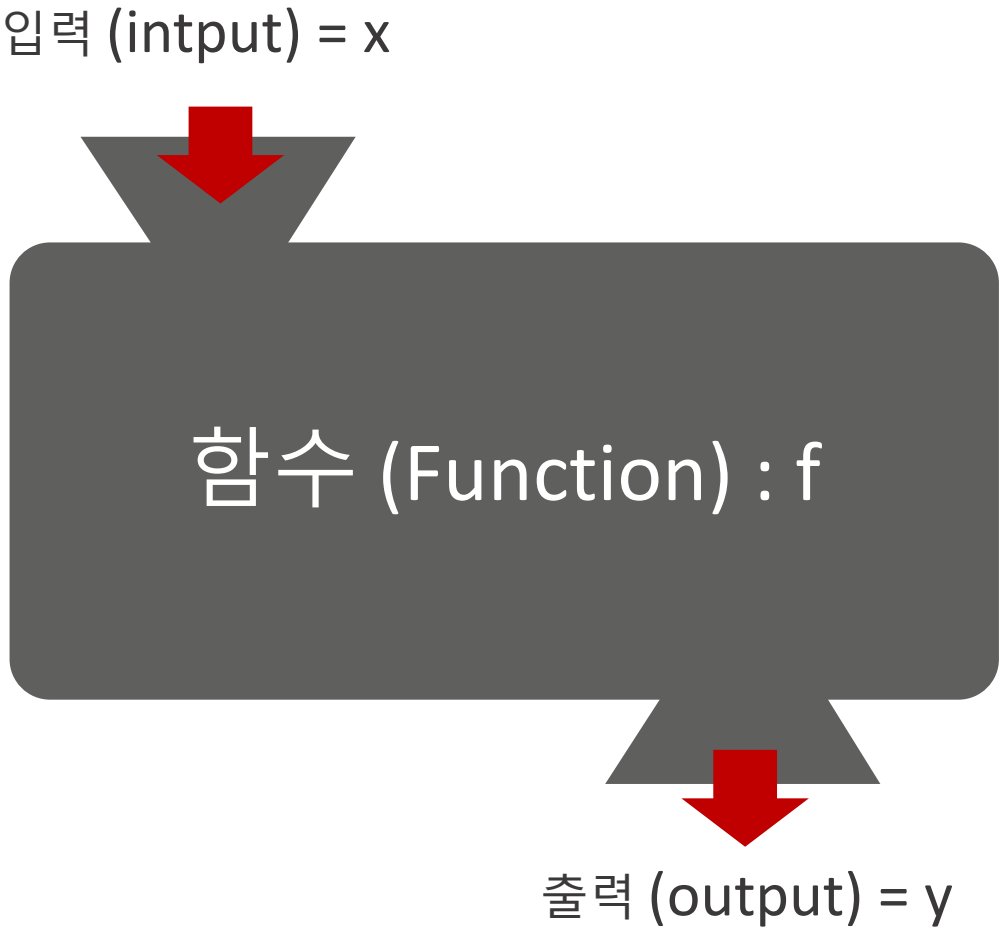
목차 a table of contents



함수 기본

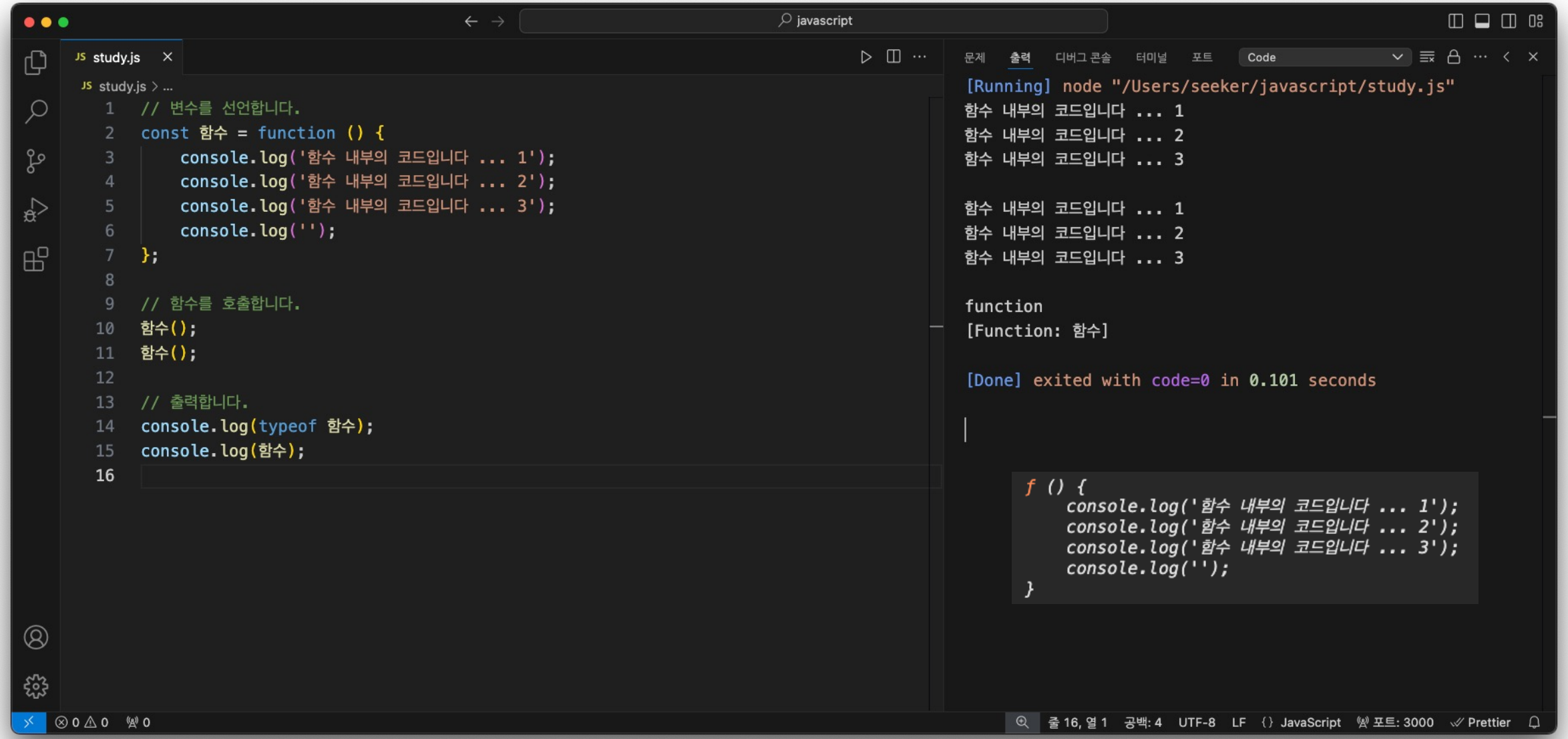
Part 5 함수

$$f(x) = y$$

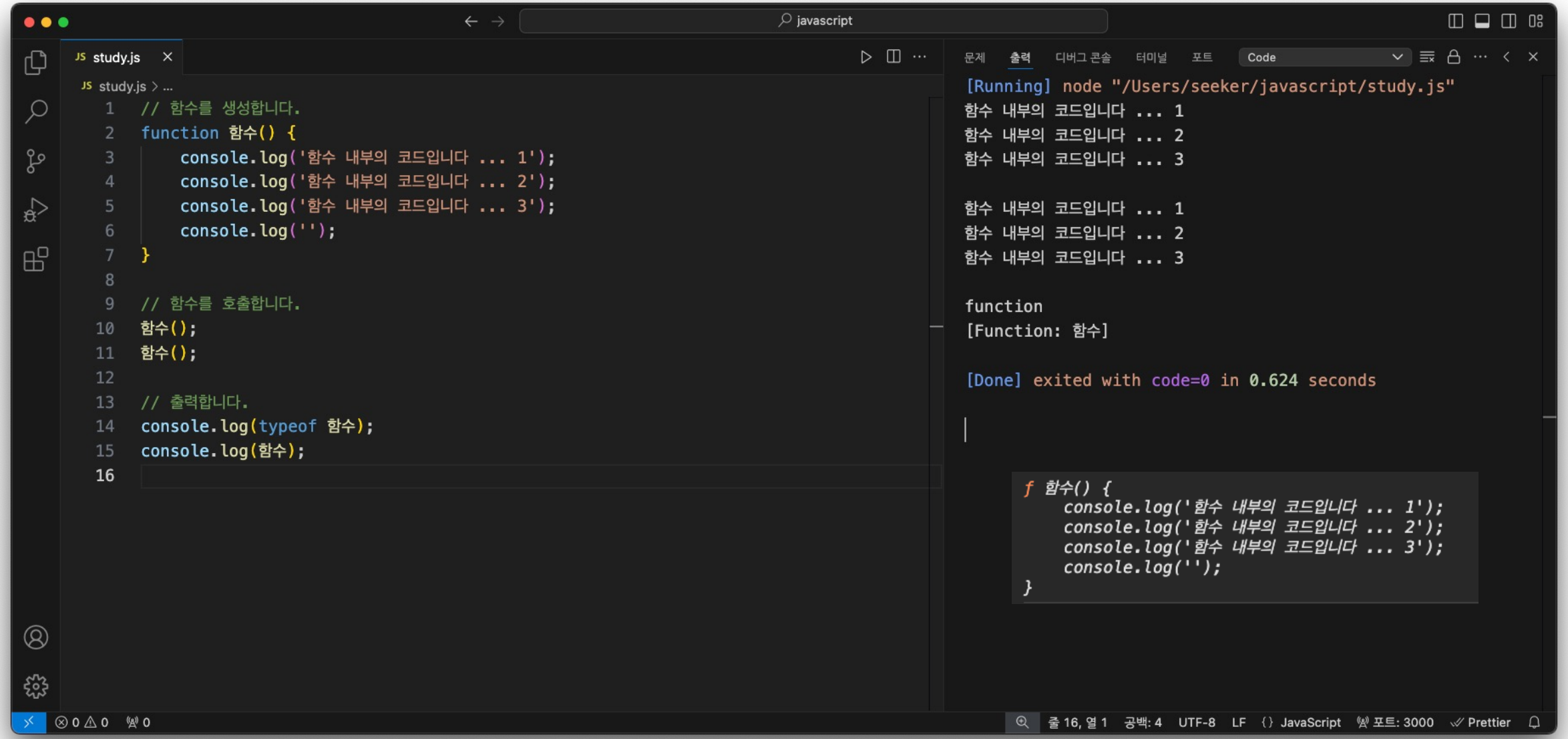


Part 5

익명 함수

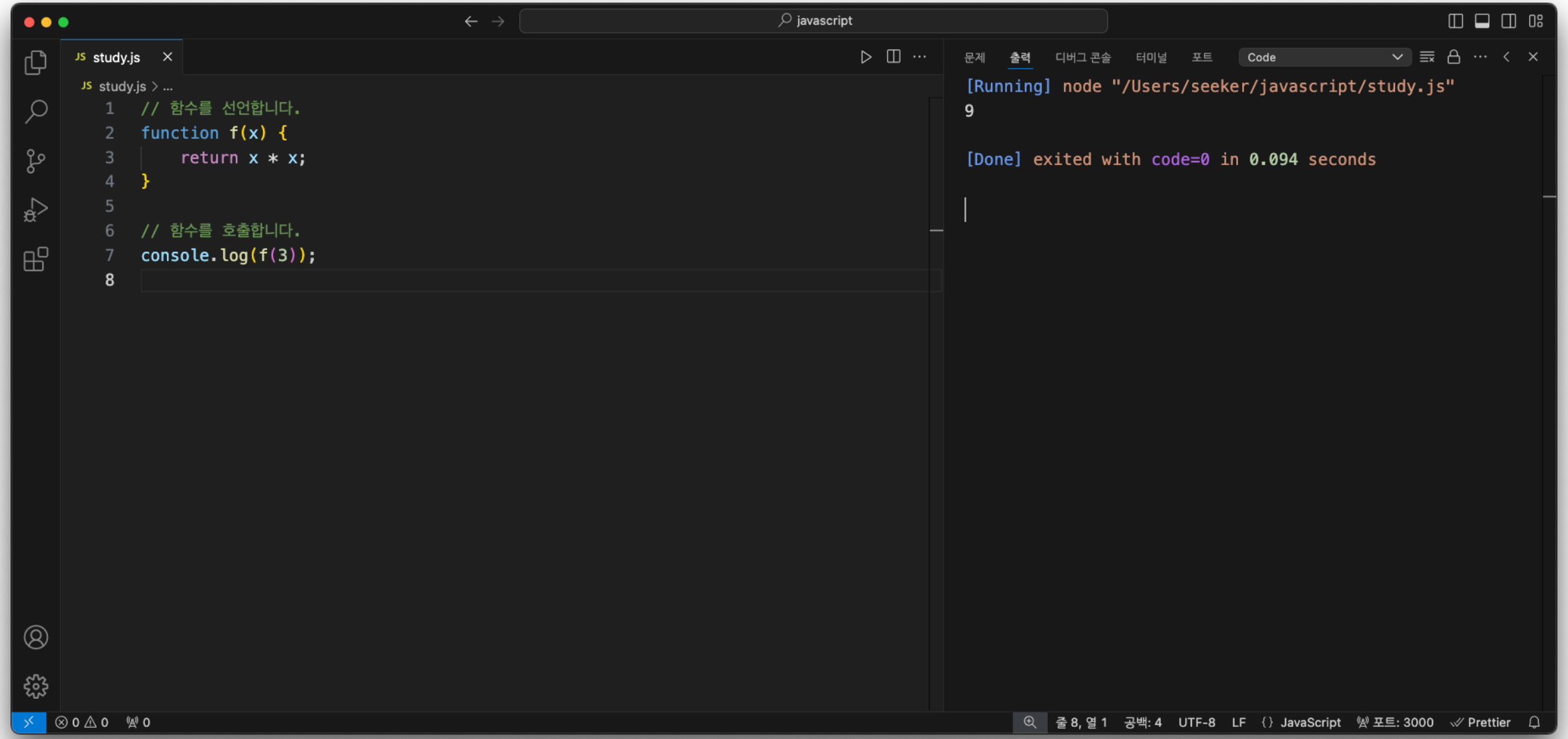

javascript
study.js
study.js > ...
1 // 변수를 선언합니다.
2 const 함수 = function () {
3 console.log('함수 내부의 코드입니다 ... 1');
4 console.log('함수 내부의 코드입니다 ... 2');
5 console.log('함수 내부의 코드입니다 ... 3');
6 console.log('');
7 };
8
9 // 함수를 호출합니다.
10 함수();
11 함수();
12
13 // 출력합니다.
14 console.log(typeof 함수);
15 console.log(함수);
16
문제 출력 디버그 콘솔 터미널 포트 Code
[Running] node "/Users/seeker/javascript/study.js"
함수 내부의 코드입니다 ... 1
함수 내부의 코드입니다 ... 2
함수 내부의 코드입니다 ... 3
함수 내부의 코드입니다 ... 1
함수 내부의 코드입니다 ... 2
함수 내부의 코드입니다 ... 3
function
[Function: 함수]
[Done] exited with code=0 in 0.101 seconds
f () {
console.log('함수 내부의 코드입니다 ... 1');
console.log('함수 내부의 코드입니다 ... 2');
console.log('함수 내부의 코드입니다 ... 3');
console.log('');
}
0 0 0
줄 16, 열 1 공백: 4 UTF-8 LF JavaScript 포트: 3000 Prettier

Part 5 선언적 함수


javascript
study.js
study.js > ...
1 // 함수를 생성합니다.
2 function 함수() {
3 console.log('함수 내부의 코드입니다 ... 1');
4 console.log('함수 내부의 코드입니다 ... 2');
5 console.log('함수 내부의 코드입니다 ... 3');
6 console.log('');
7 }
8
9 // 함수를 호출합니다.
10 함수();
11 함수();
12
13 // 출력합니다.
14 console.log(typeof 함수);
15 console.log(함수);
16
문제 출력 디버그 콘솔 터미널 포트 Code
[Running] node "/Users/seeker/javascript/study.js"
함수 내부의 코드입니다 ... 1
함수 내부의 코드입니다 ... 2
함수 내부의 코드입니다 ... 3
함수 내부의 코드입니다 ... 1
함수 내부의 코드입니다 ... 2
함수 내부의 코드입니다 ... 3
function
[Function: 함수]
[Done] exited with code=0 in 0.624 seconds
f 함수() {
console.log('함수 내부의 코드입니다 ... 1');
console.log('함수 내부의 코드입니다 ... 2');
console.log('함수 내부의 코드입니다 ... 3');
console.log('');
}
⊗ 0 ⚠ 0 0
줄 16, 열 1 공백: 4 UTF-8 LF { } JavaScript 포트: 3000 Prettier

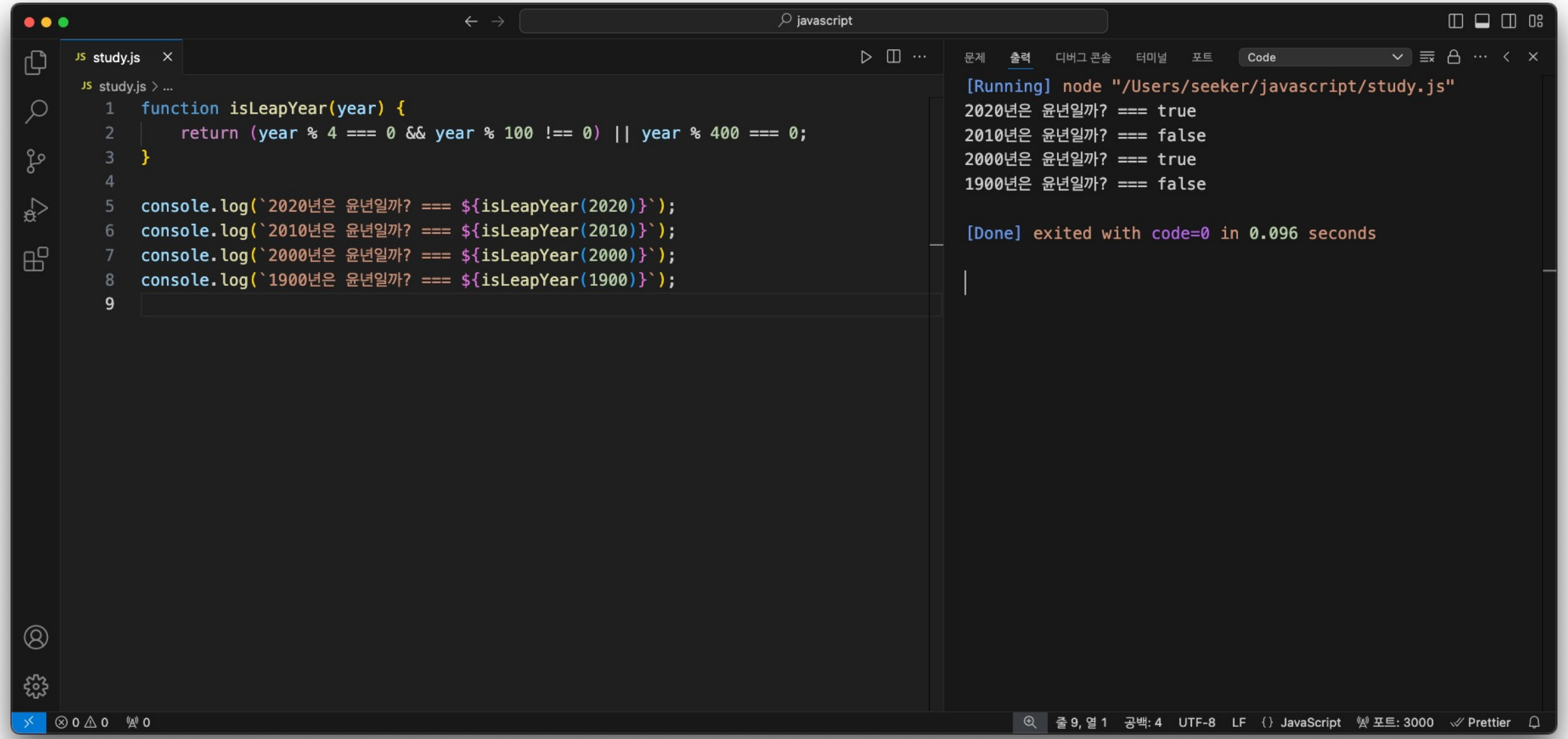
Part 5

매개변수와 리턴 값



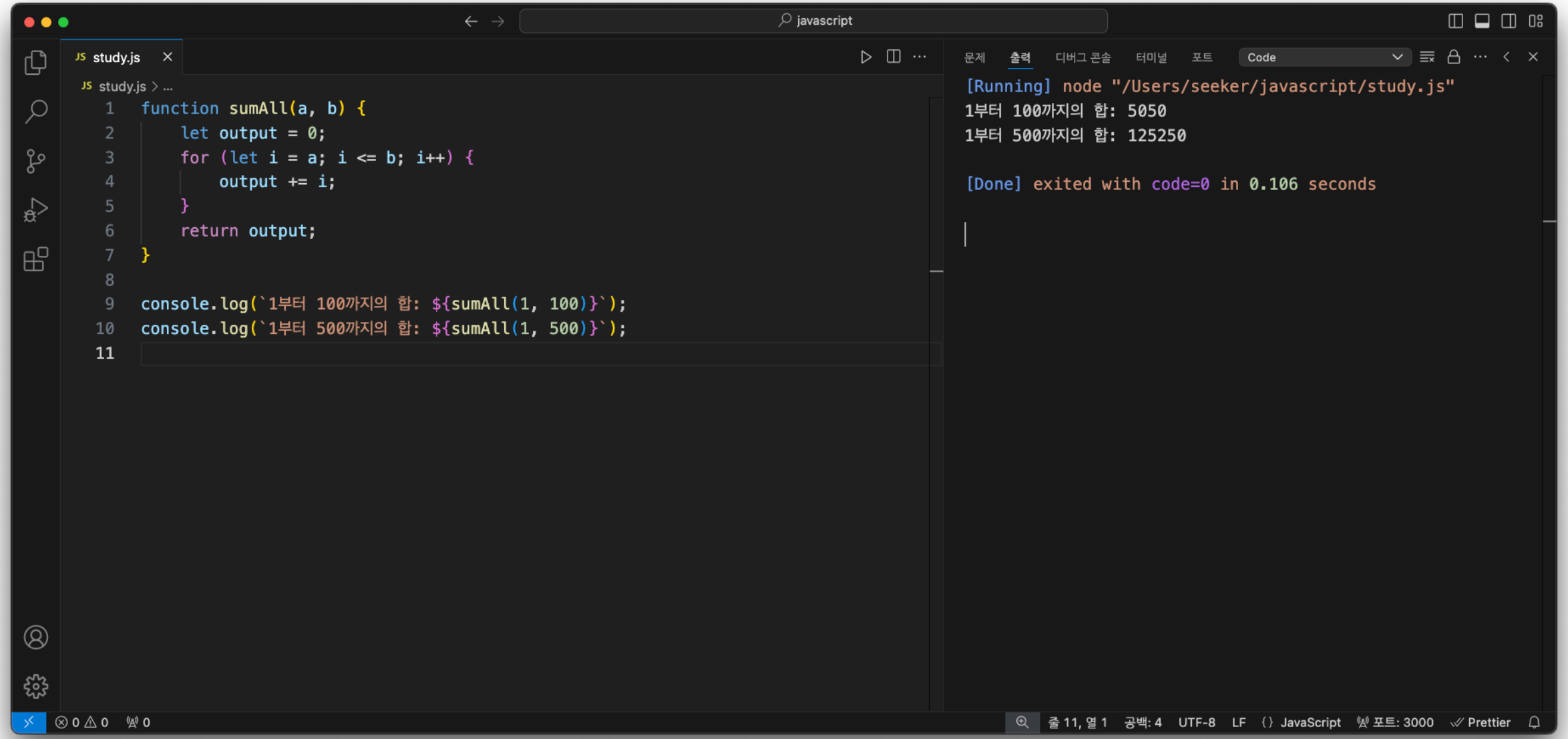
Part 5

기본적인 함수 예제 > 윤년 확인



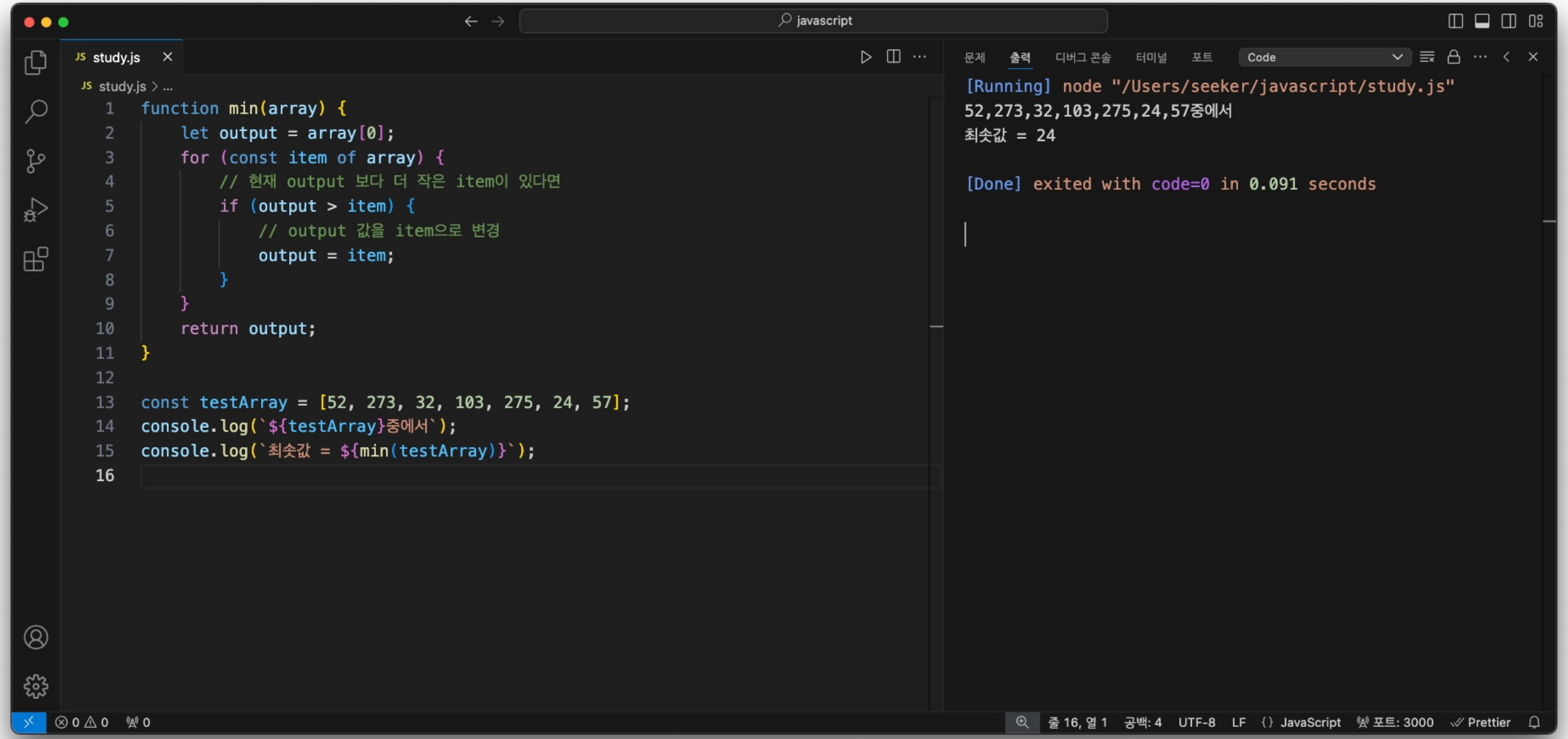
Part 5

기본적인 함수 예제 > A부터 B까지 더하기



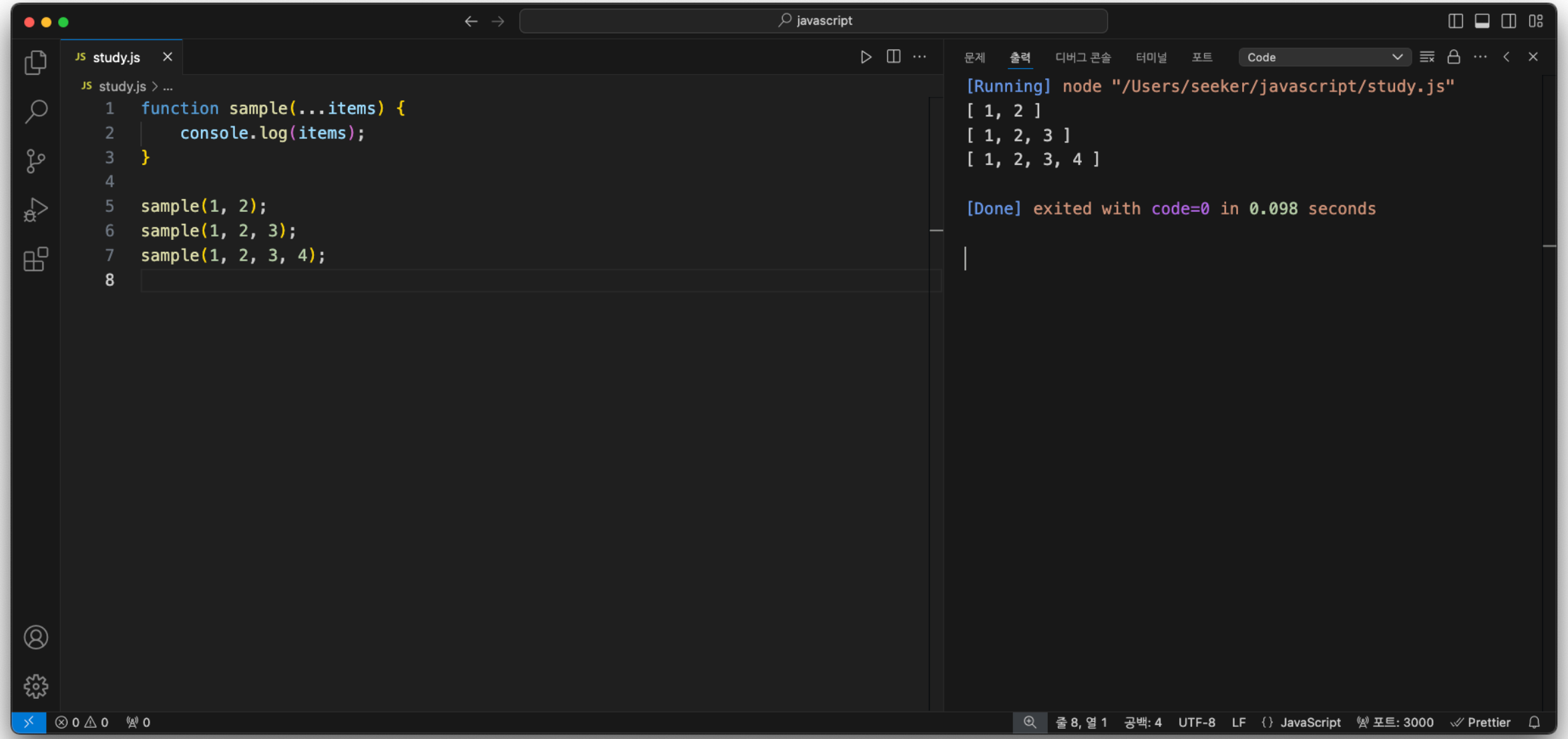
Part 5

기본적인 함수 예제 > 최소값 구하기



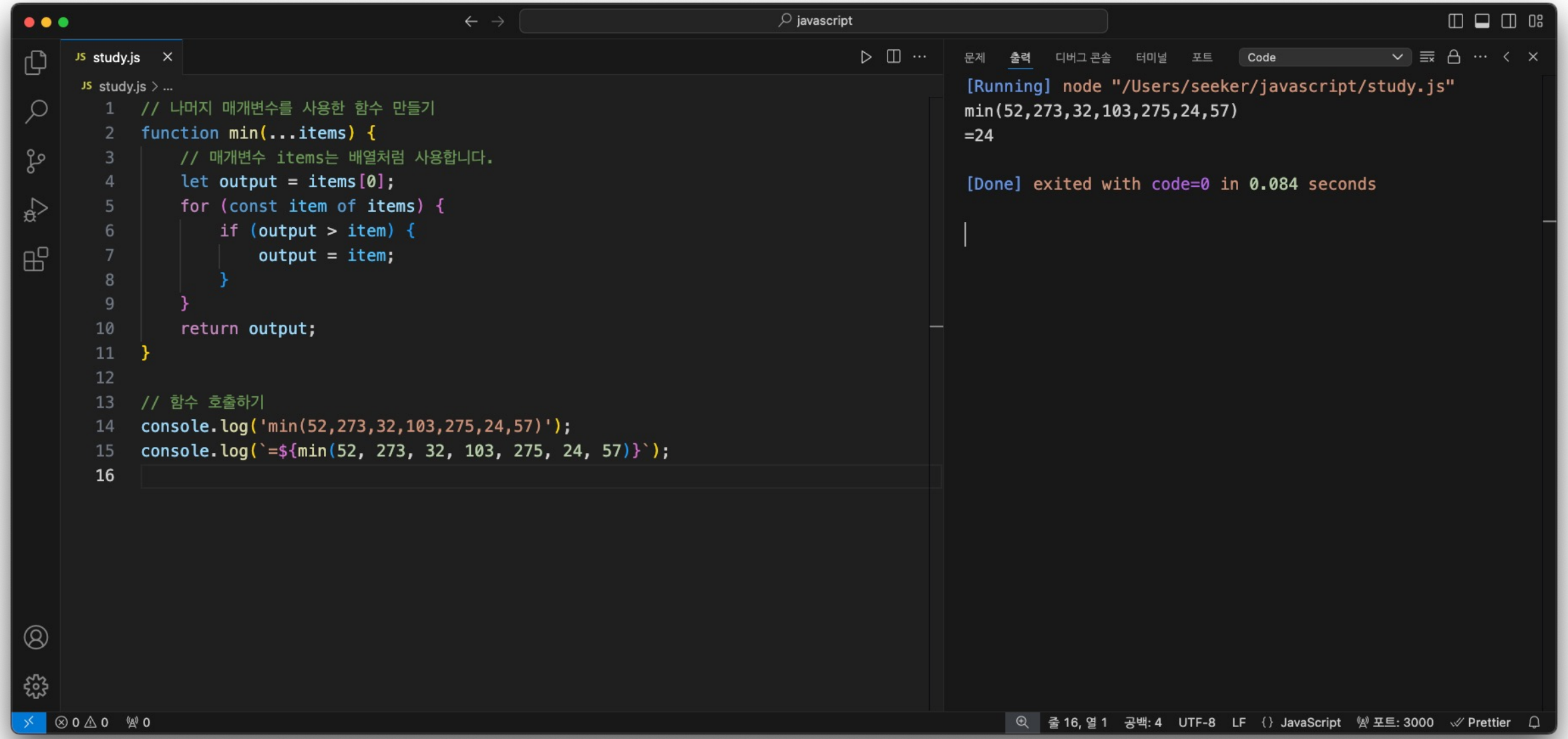
Part 5

나머지 매개변수



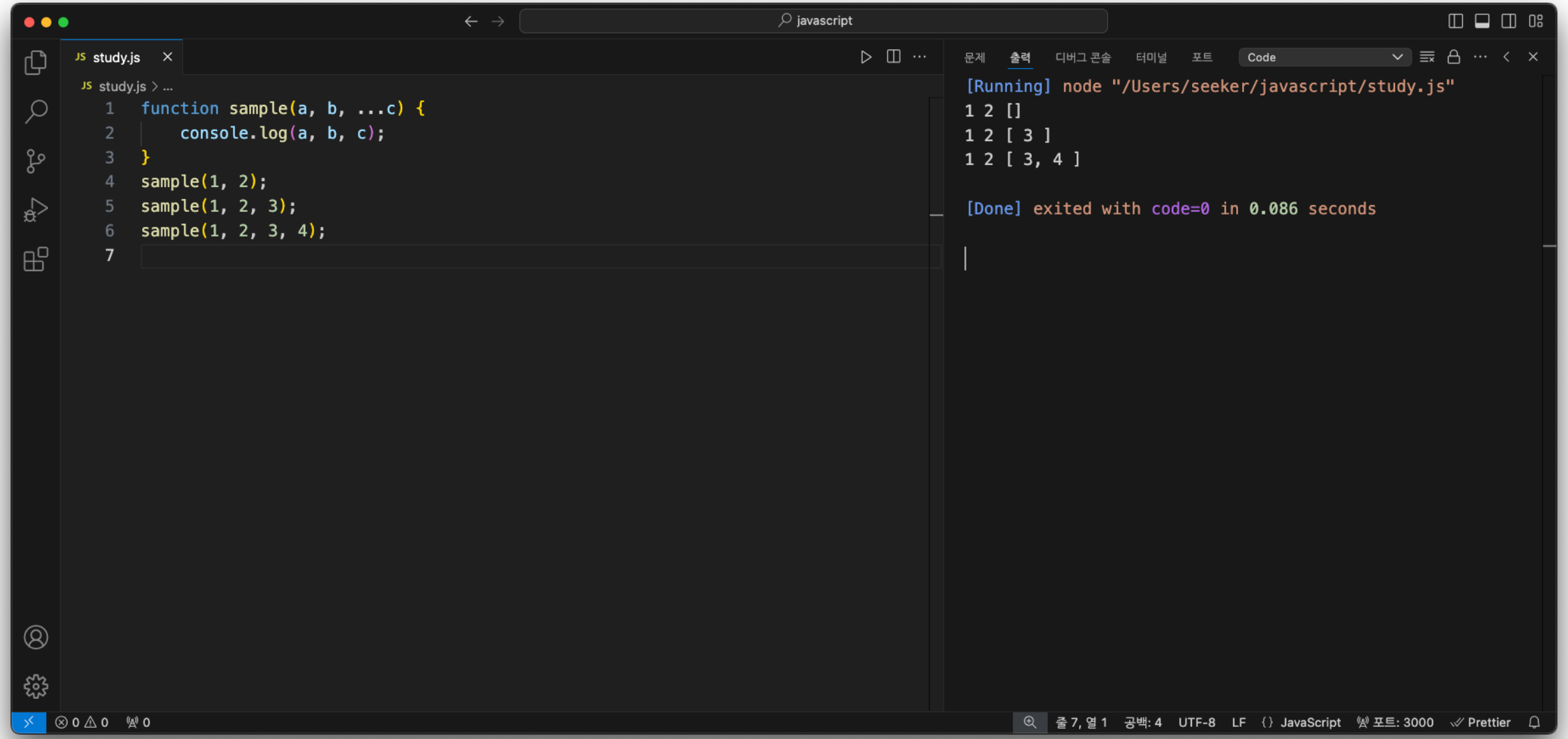
Part 5

나머지 매개변수



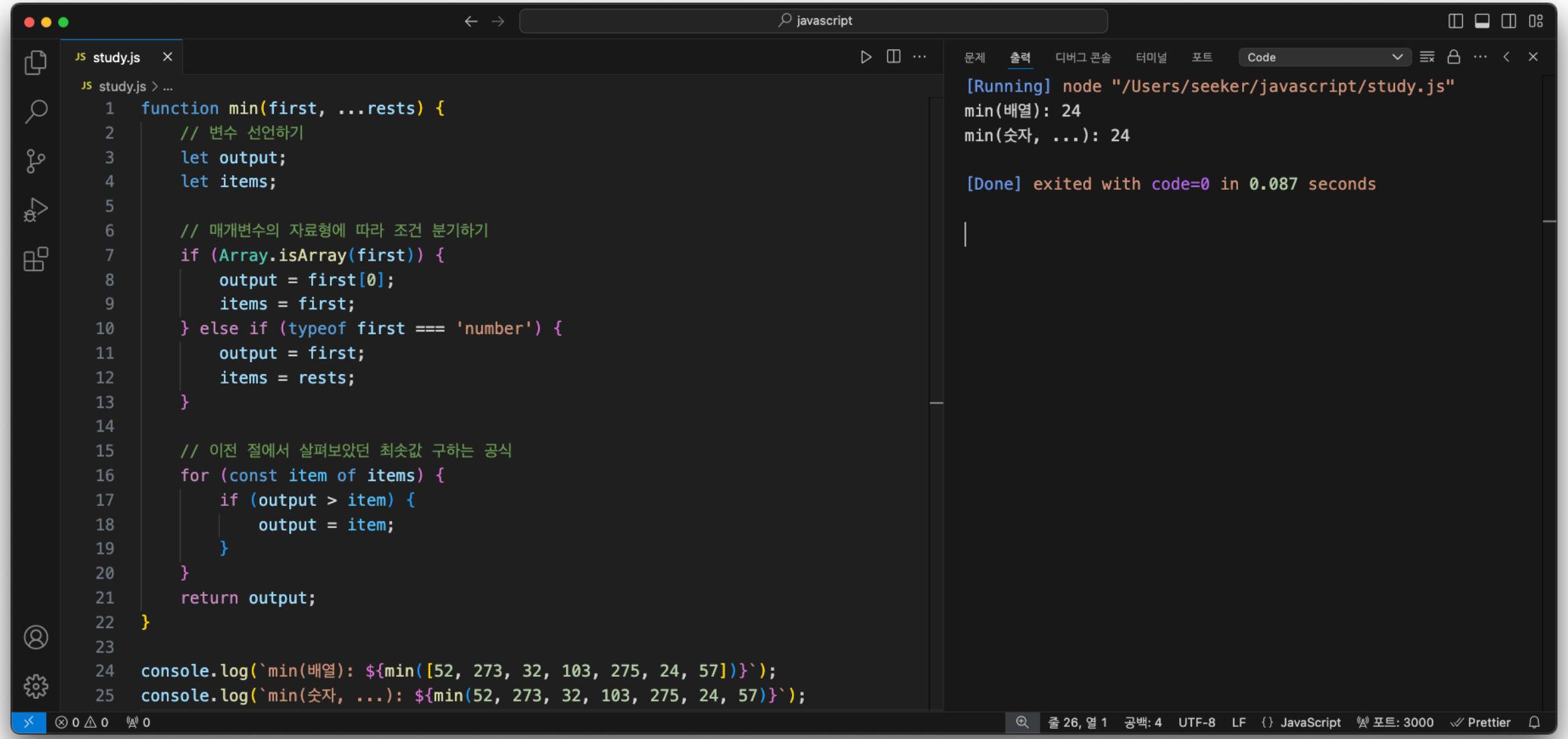
Part 5

나머지 매개변수 > 일반 매개변수 조합

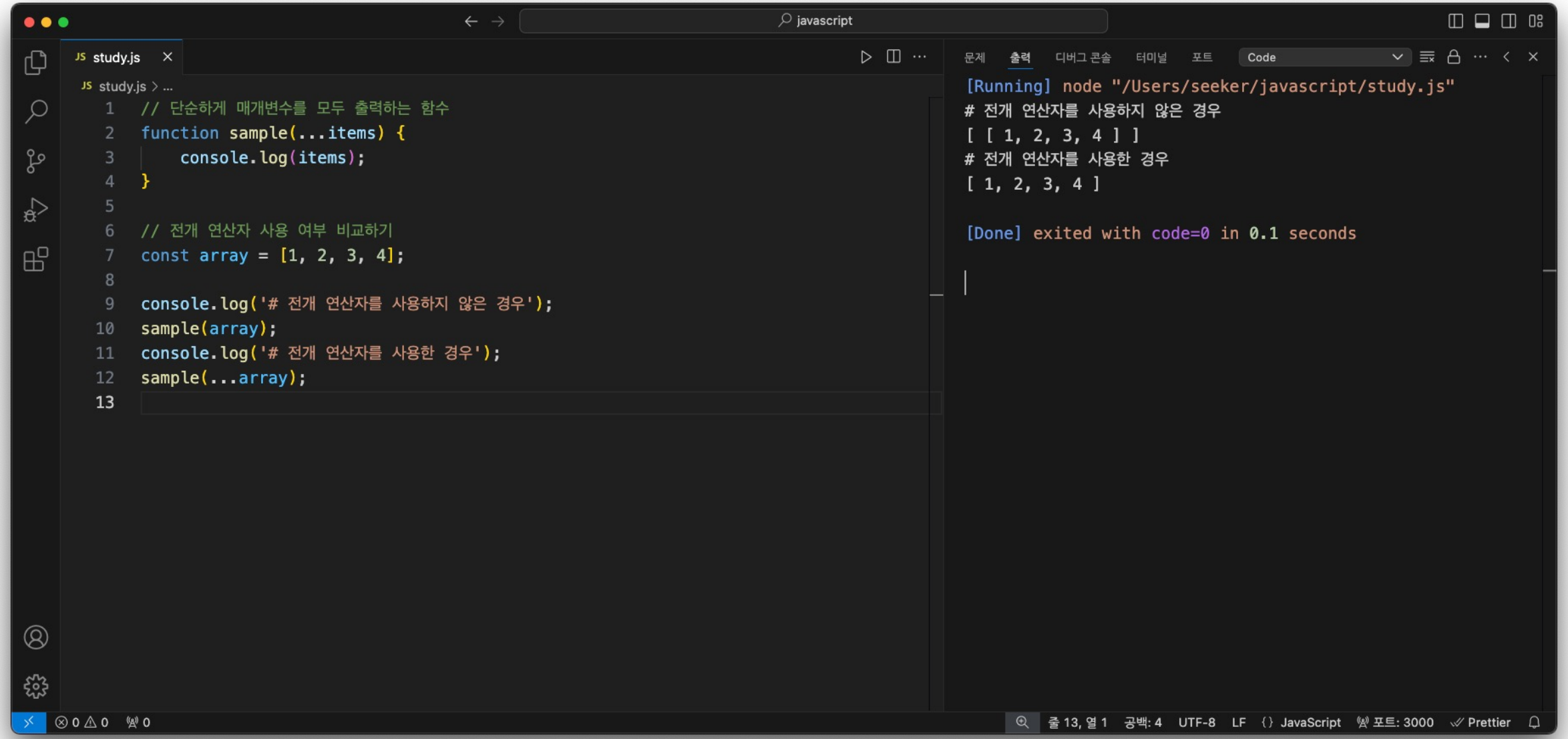


Part 5

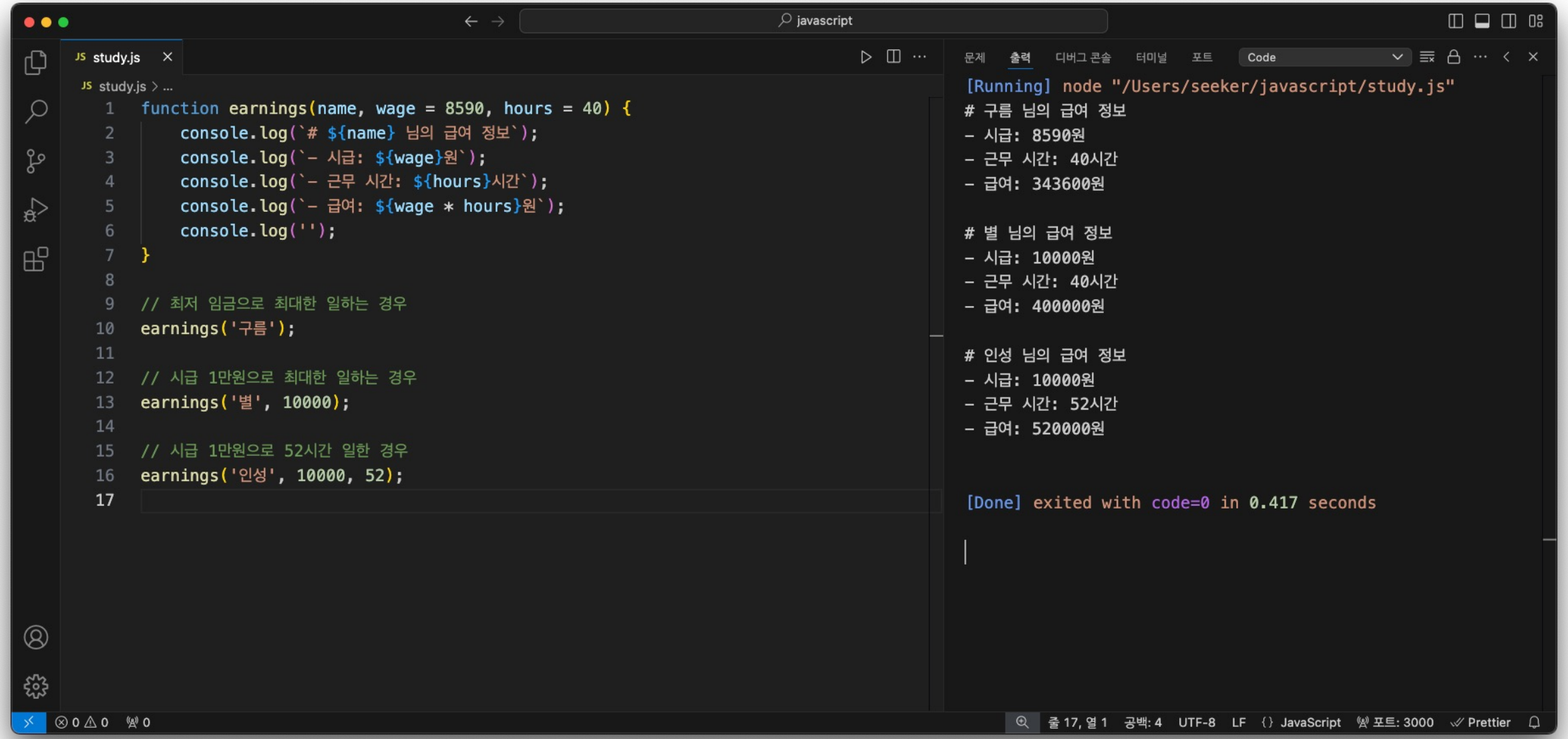
나머지 매개변수



Part 5 전개 연산자

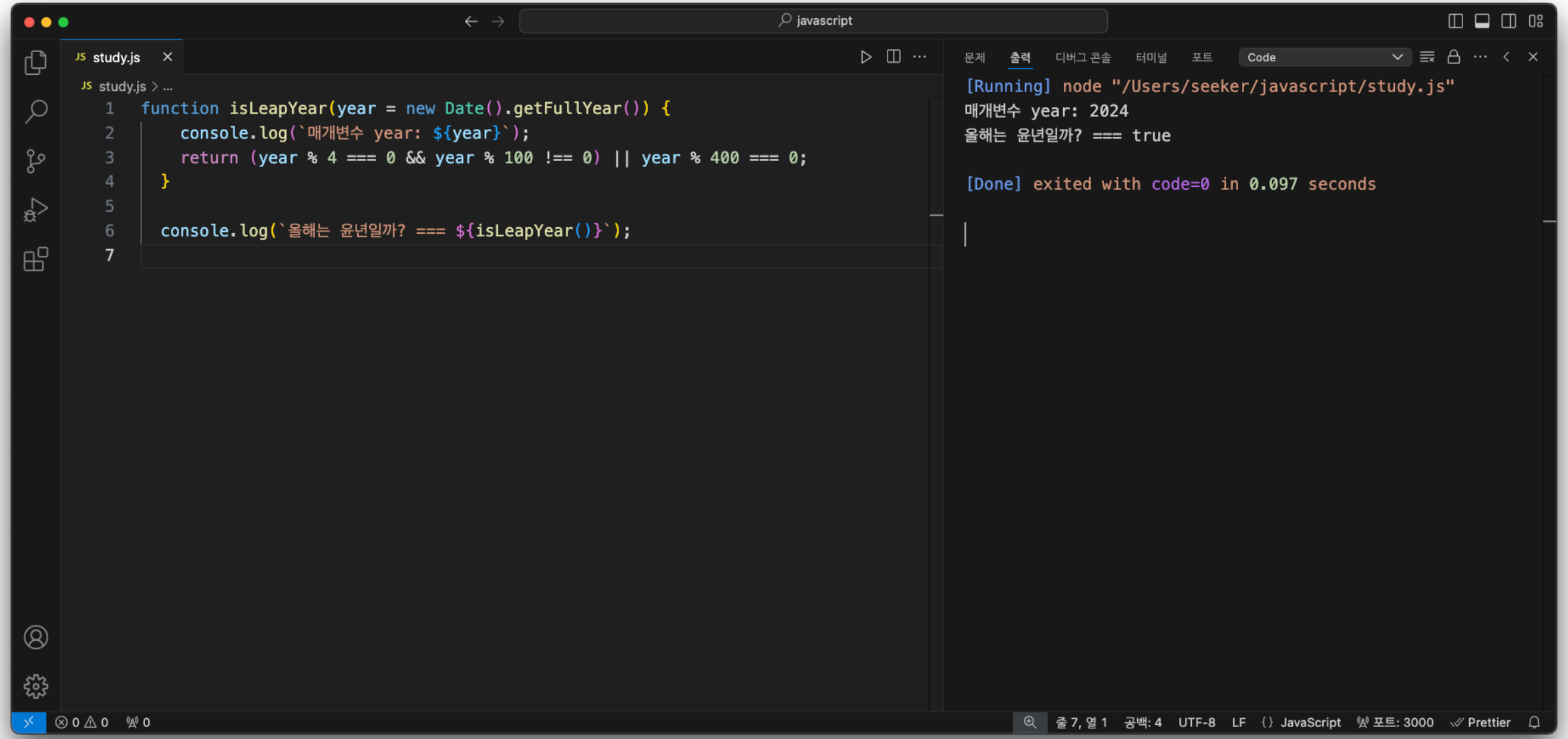

javascript
study.js
study.js > ...
1 // 단순하게 매개변수를 모두 출력하는 함수
2 function sample(...items) {
3 console.log(items);
4 }
5
6 // 전개 연산자 사용 여부 비교하기
7 const array = [1, 2, 3, 4];
8
9 console.log('# 전개 연산자를 사용하지 않은 경우');
10 sample(array);
11 console.log('# 전개 연산자를 사용한 경우');
12 sample(...array);
13
문제 출력 디버그 콘솔 터미널 포트 Code
[Running] node "/Users/seeker/javascript/study.js"
전개 연산자를 사용하지 않은 경우
[[1, 2, 3, 4]]
전개 연산자를 사용한 경우
[1, 2, 3, 4]
[Done] exited with code=0 in 0.1 seconds
⊗ 0 ⚠ 0 0
줄 13, 열 1 공백: 4 UTF-8 LF { } JavaScript 포트: 3000 Prettier

Part 5 기본 매개변수

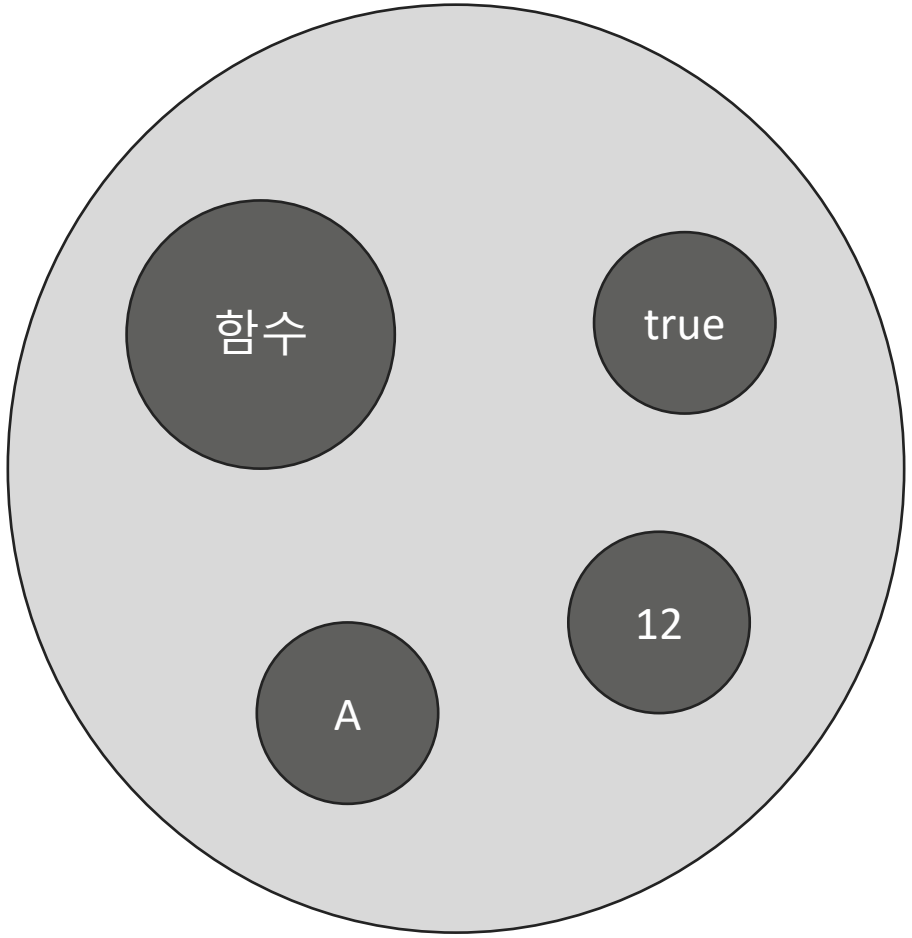


Part 5

기본 매개변수



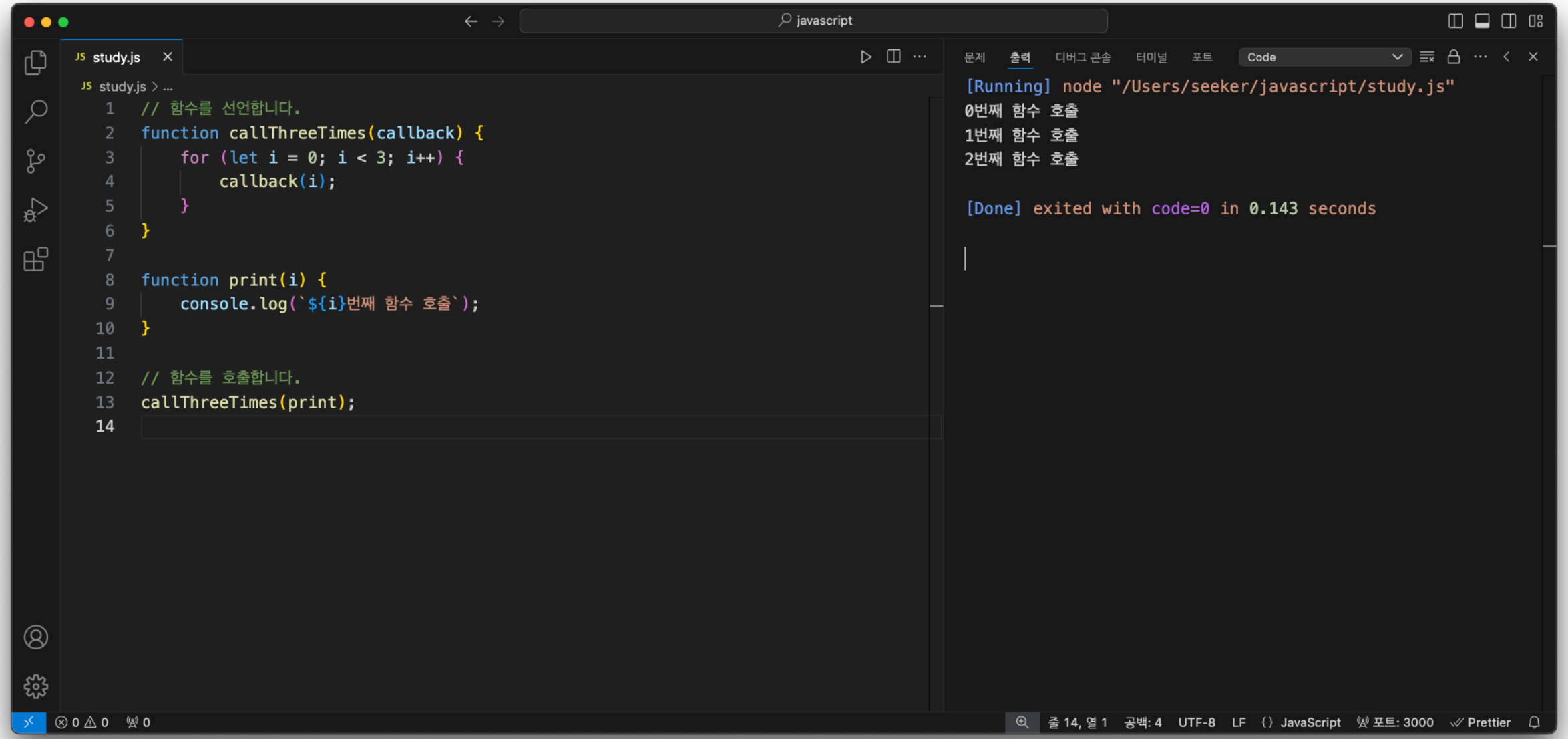
함수 고급



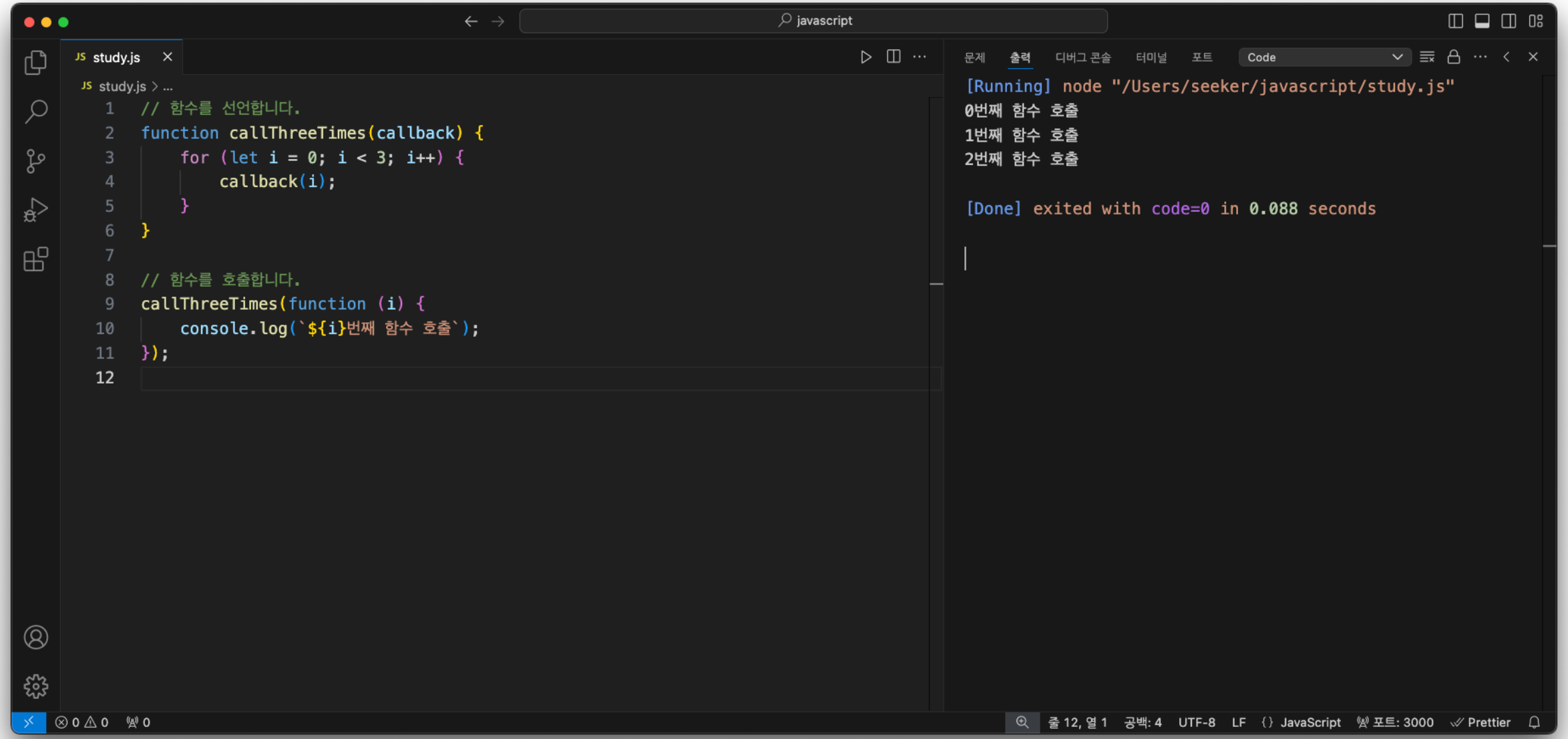
매개변수

Part 5

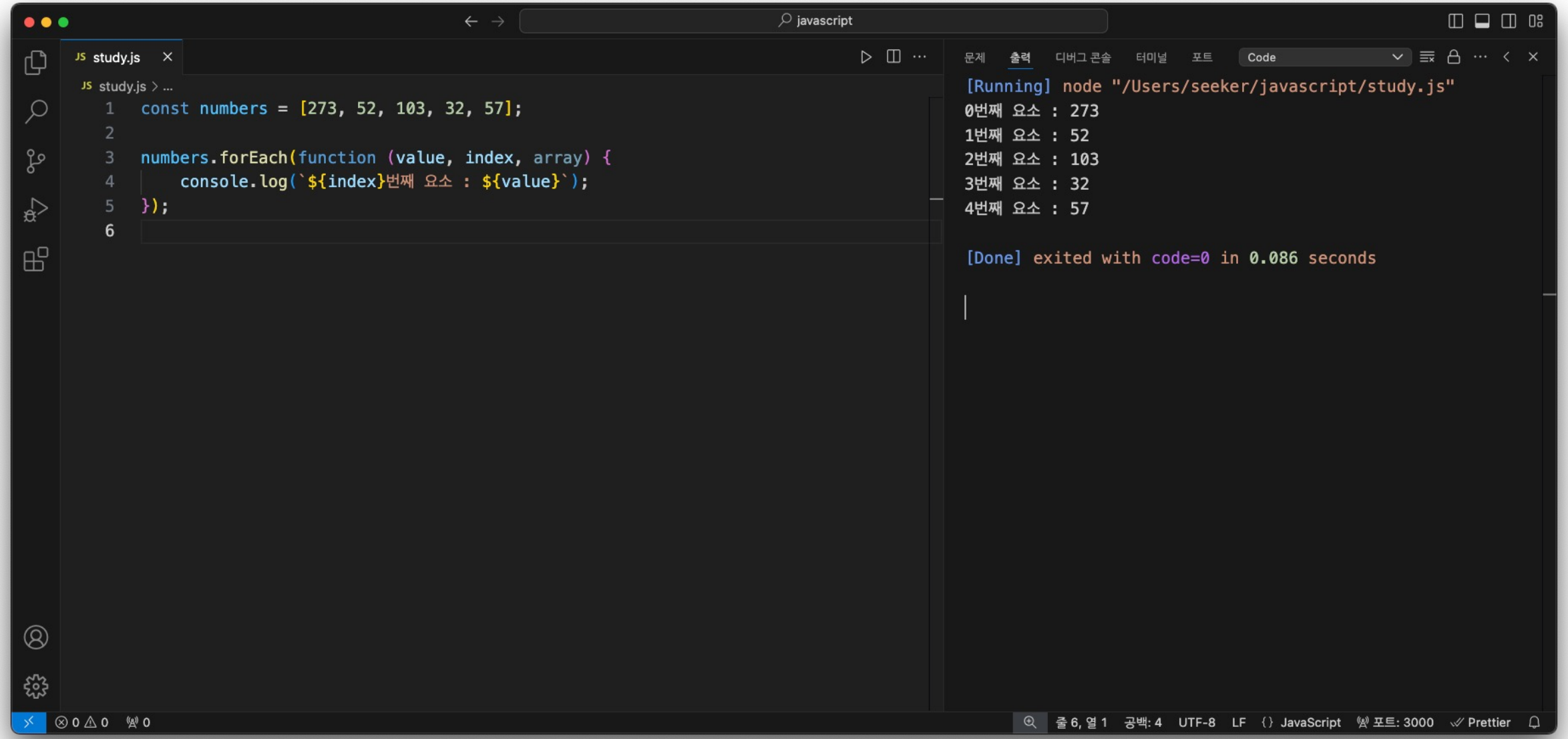
콜백 함수


javascript
study.js
study.js > ...
1 // 함수를 선언합니다.
2 function callThreeTimes(callback) {
3 for (let i = 0; i < 3; i++) {
4 callback(i);
5 }
6 }
7
8 function print(i) {
9 console.log(`${i}번째 함수 호출`);
10 }
11
12 // 함수를 호출합니다.
13 callThreeTimes(print);
14
문제
출력
디버그 콘솔
터미널
포트
Code
[Running] node "/Users/seeker/javascript/study.js"
0번째 함수 호출
1번째 함수 호출
2번째 함수 호출
[Done] exited with code=0 in 0.143 seconds
⊗ 0 ⚠ 0
0
줄 14, 열 1
공백: 4
UTF-8
LF
{} JavaScript
포트: 3000
Prettier

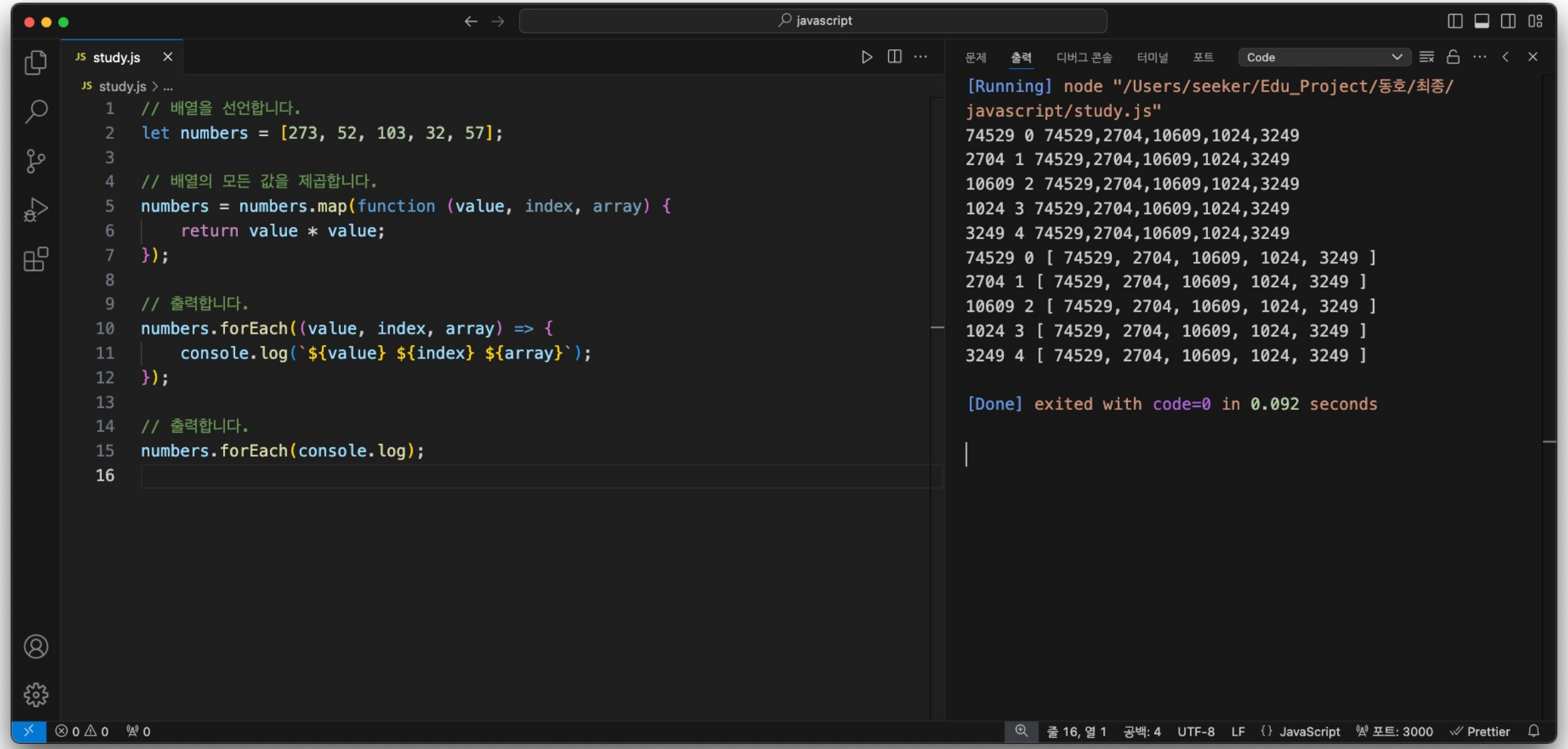
Part 5 콜백 함수


javascript
study.js
study.js > ...
1 // 함수를 선언합니다.
2 function callThreeTimes(callback) {
3 for (let i = 0; i < 3; i++) {
4 callback(i);
5 }
6 }
7
8 // 함수를 호출합니다.
9 callThreeTimes(function (i) {
10 console.log(`${i}번째 함수 호출`);
11 });
12
문제 출력 디버그 콘솔 터미널 포트
Code
[Running] node "/Users/seeker/javascript/study.js"
0번째 함수 호출
1번째 함수 호출
2번째 함수 호출
[Done] exited with code=0 in 0.088 seconds
0 0 0
줄 12, 열 1 공백: 4 UTF-8 LF {} JavaScript 포트: 3000 Prettier

Part 5 콜백 함수 활용 > forEach()

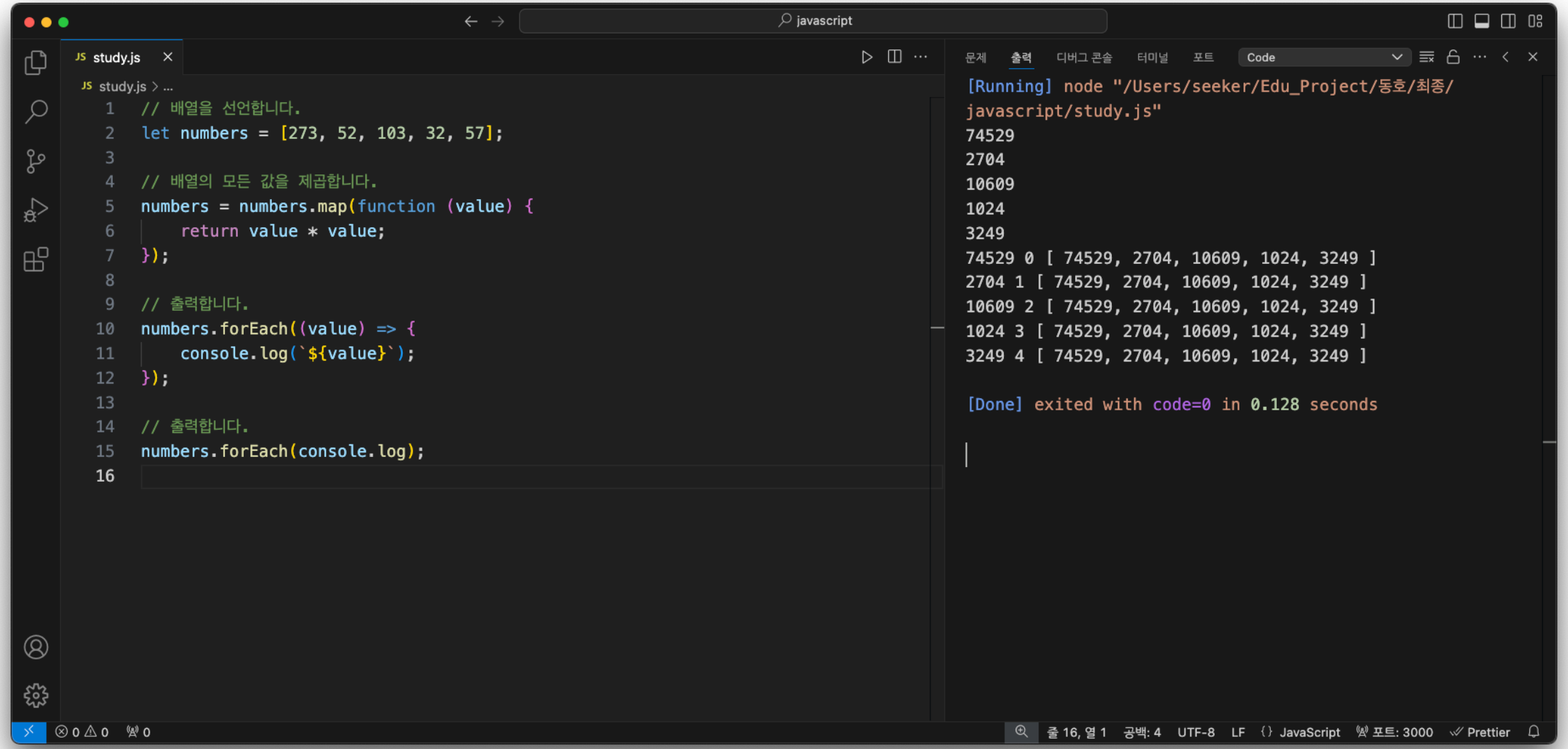


Part 5 콜백 함수 활용 > map()



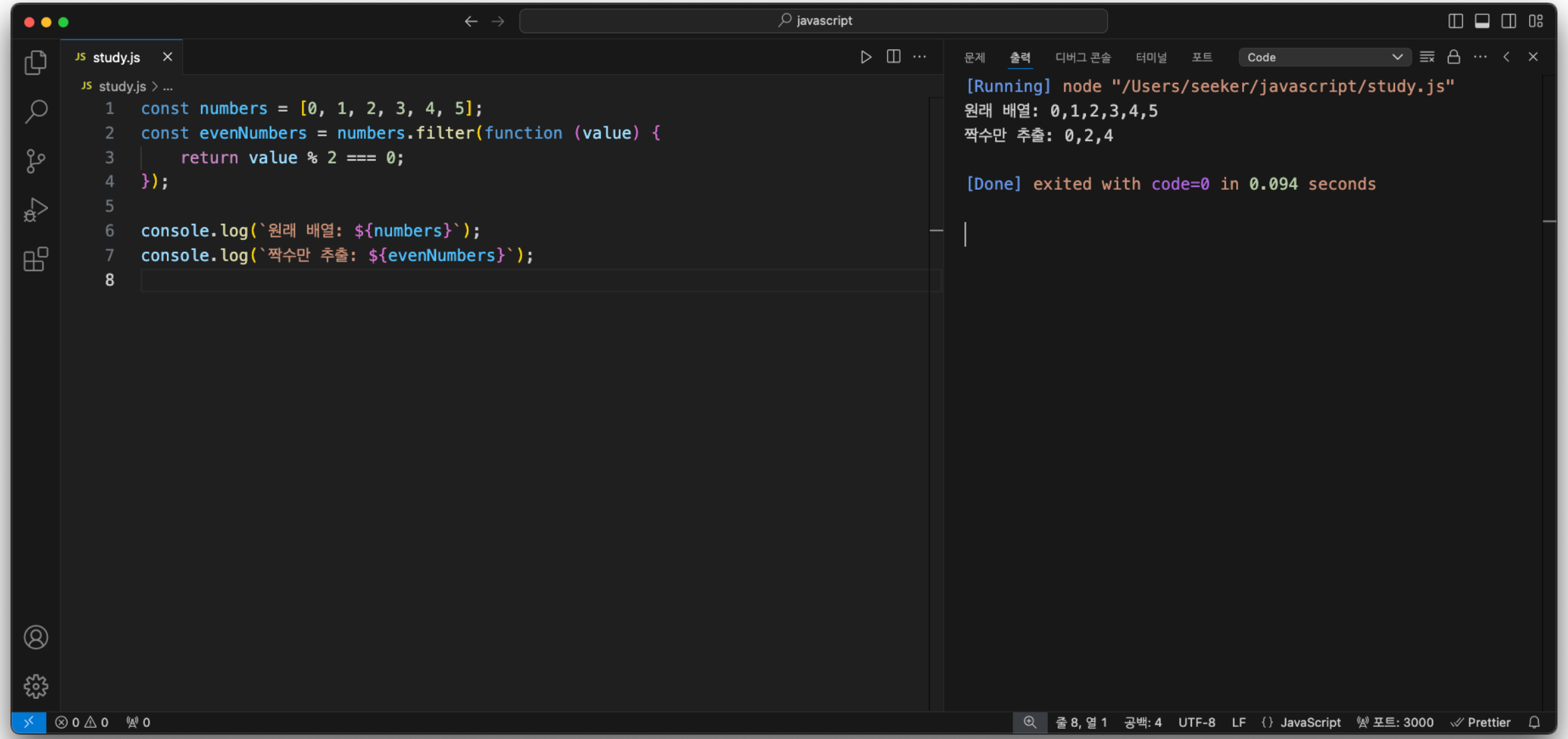
Part 5

콜백 함수 활용 > map()



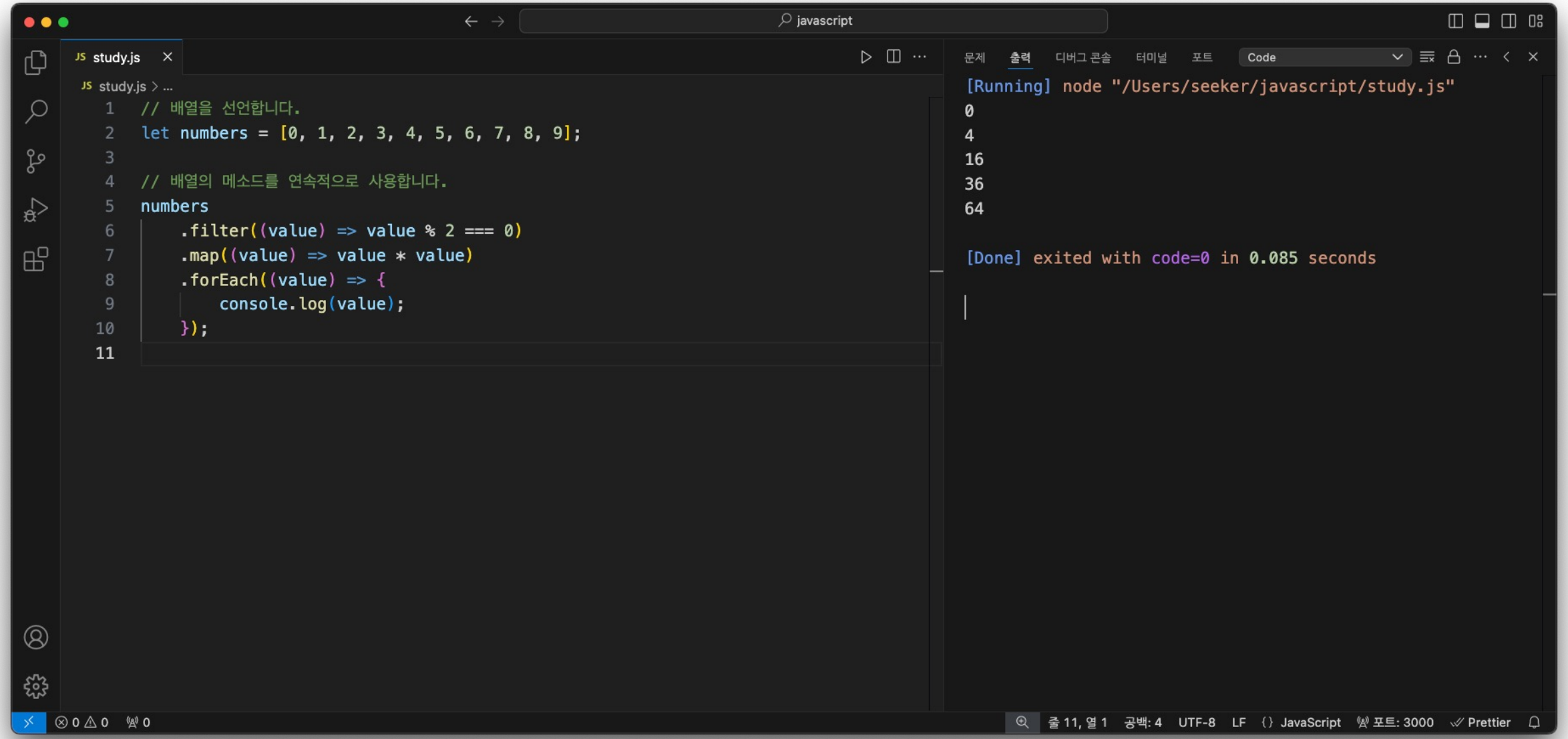
Part 5

콜백 함수 활용 > filter()



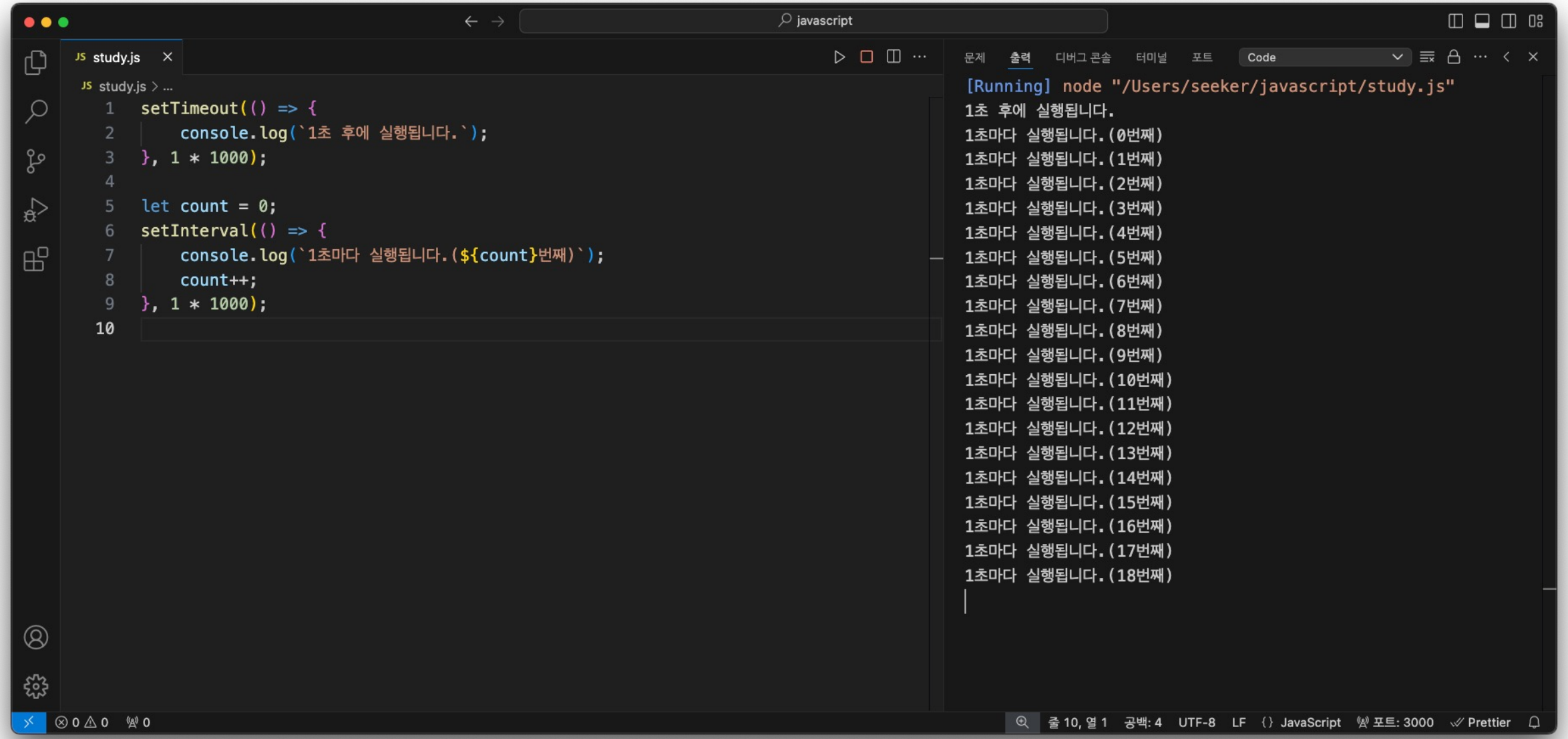
Part 5

화살표 함수


javascript
study.js
study.js > ...
1 // 배열을 선언합니다.
2 let numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];
3
4 // 배열의 메소드를 연속적으로 사용합니다.
5 numbers
6 .filter((value) => value % 2 === 0)
7 .map((value) => value * value)
8 .forEach((value) => {
9 console.log(value);
10 });
11
문제 출력 디버그 콘솔 터미널 포트 Code
[Running] node "/Users/seeker/javascript/study.js"
0
4
16
36
64
[Done] exited with code=0 in 0.085 seconds
⊗ 0 ⚠ 0 0
줄 11, 열 1 공백: 4 UTF-8 LF { } JavaScript 포트: 3000 Prettier

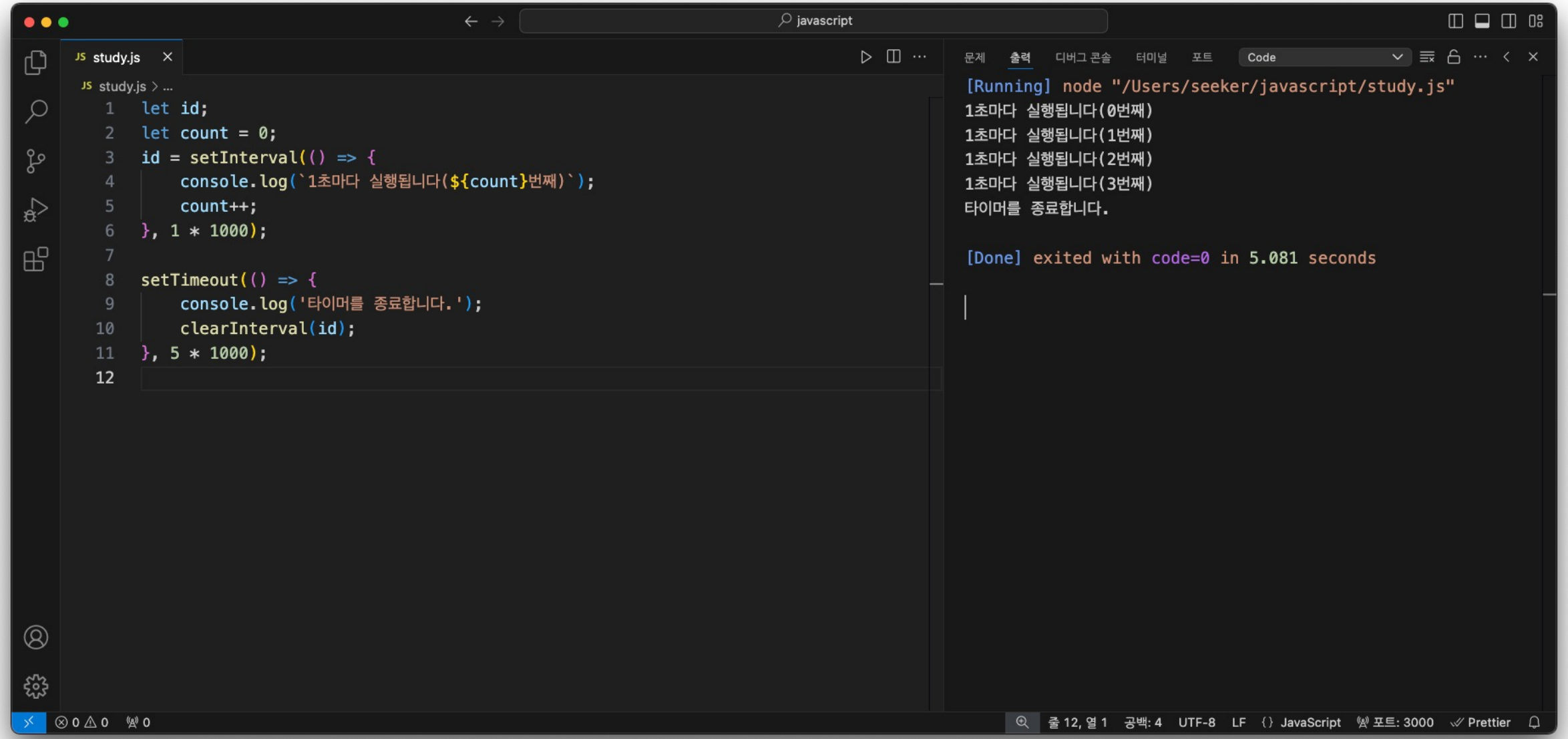
Part 5

타이머 함수

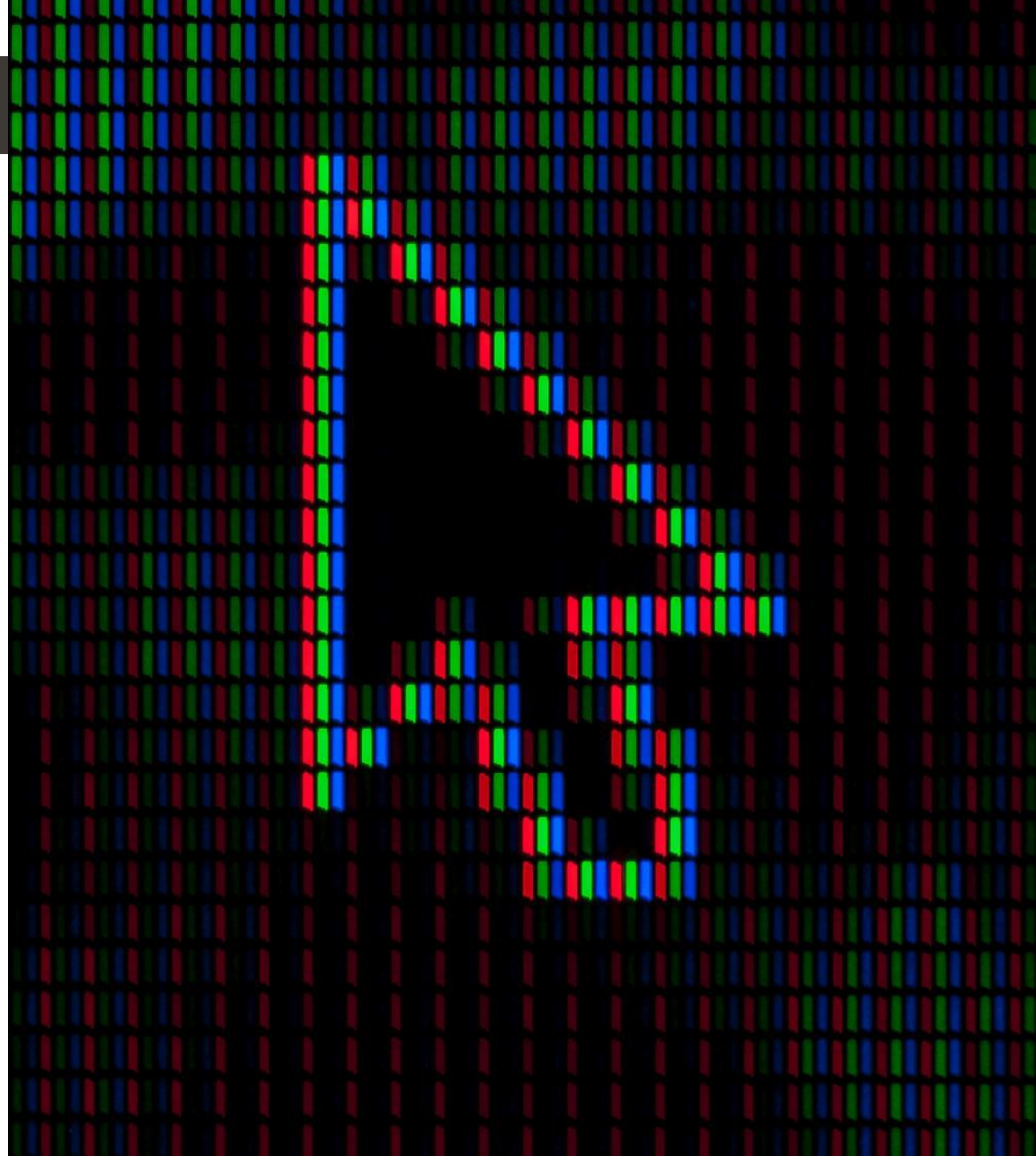

javascript
study.js
study.js > ...
1 setTimeout(() => {
2 console.log(`1초 후에 실행됩니다.`);
3 }, 1 * 1000);
4
5 let count = 0;
6 setInterval(() => {
7 console.log(`1초마다 실행됩니다.(${count}번째)`);
8 count++;
9 }, 1 * 1000);
10
문제 출력 디버그 콘솔 터미널 포트 Code
[Running] node "/Users/seeker/javascript/study.js"
1초 후에 실행됩니다.
1초마다 실행됩니다.(0번째)
1초마다 실행됩니다.(1번째)
1초마다 실행됩니다.(2번째)
1초마다 실행됩니다.(3번째)
1초마다 실행됩니다.(4번째)
1초마다 실행됩니다.(5번째)
1초마다 실행됩니다.(6번째)
1초마다 실행됩니다.(7번째)
1초마다 실행됩니다.(8번째)
1초마다 실행됩니다.(9번째)
1초마다 실행됩니다.(10번째)
1초마다 실행됩니다.(11번째)
1초마다 실행됩니다.(12번째)
1초마다 실행됩니다.(13번째)
1초마다 실행됩니다.(14번째)
1초마다 실행됩니다.(15번째)
1초마다 실행됩니다.(16번째)
1초마다 실행됩니다.(17번째)
1초마다 실행됩니다.(18번째)
⊗ 0 ⚠ 0 0
줄 10, 열 1 공백: 4 UTF-8 LF {} JavaScript 포트: 3000 Prettier

Part 5

타이머 함수


javascript
study.js
study.js > ...
1 let id;
2 let count = 0;
3 id = setInterval(() => {
4 console.log(`1초마다 실행됩니다(${count}번째)`);
5 count++;
6 }, 1 * 1000);
7
8 setTimeout(() => {
9 console.log('타이머를 종료합니다.');
10 clearInterval(id);
11 }, 5 * 1000);
12
문제 출력 디버그 콘솔 터미널 포트
Code
[Running] node "/Users/seeker/javascript/study.js"
1초마다 실행됩니다(0번째)
1초마다 실행됩니다(1번째)
1초마다 실행됩니다(2번째)
1초마다 실행됩니다(3번째)
타이머를 종료합니다.
[Done] exited with code=0 in 5.081 seconds
⊗ 0 ⚠ 0 0
줄 12, 열 1 공백: 4 UTF-8 LF { } JavaScript 포트: 3000 Prettier

목차 a table of contents



객체 기본

객체

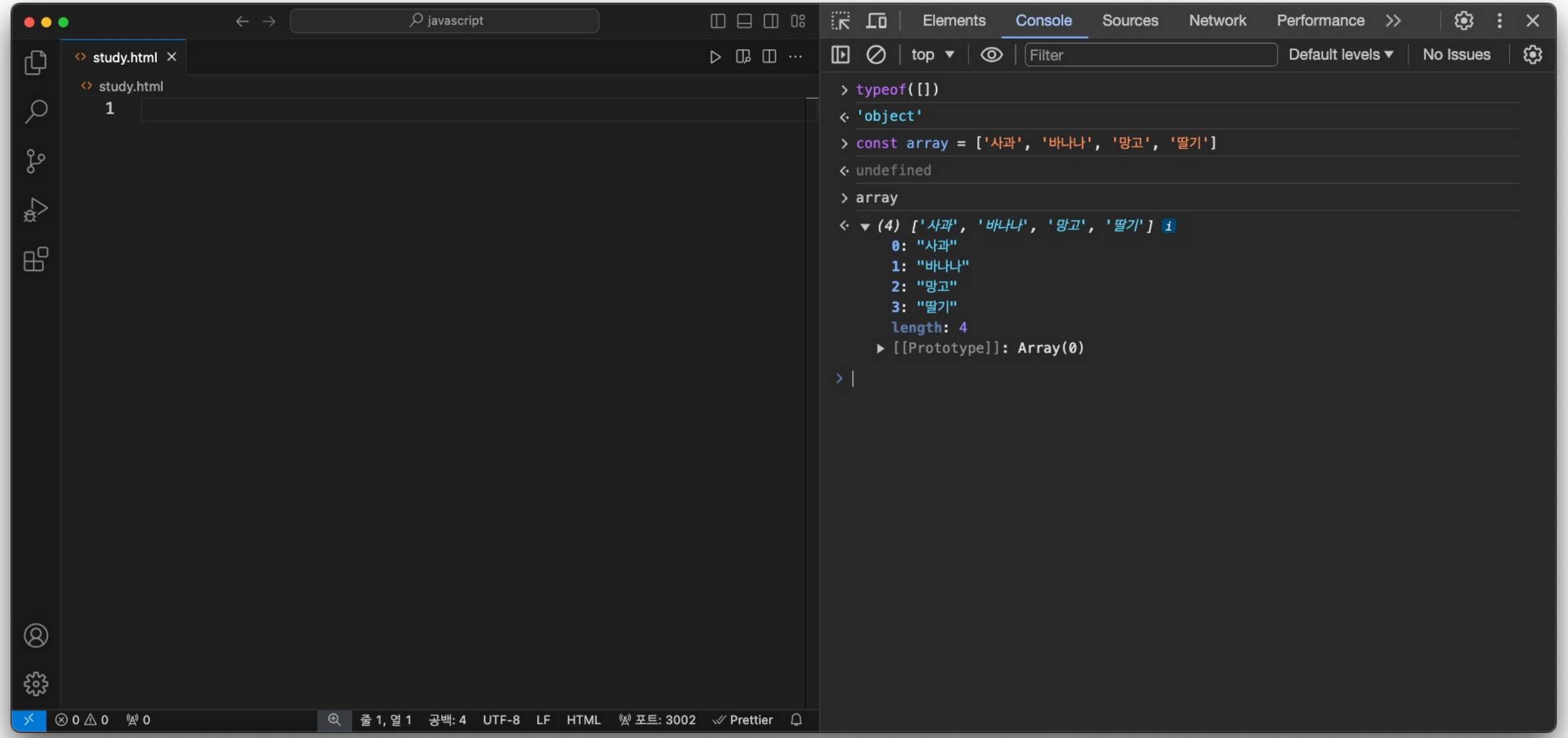
이름 : 구름
나이 : 7살

속성

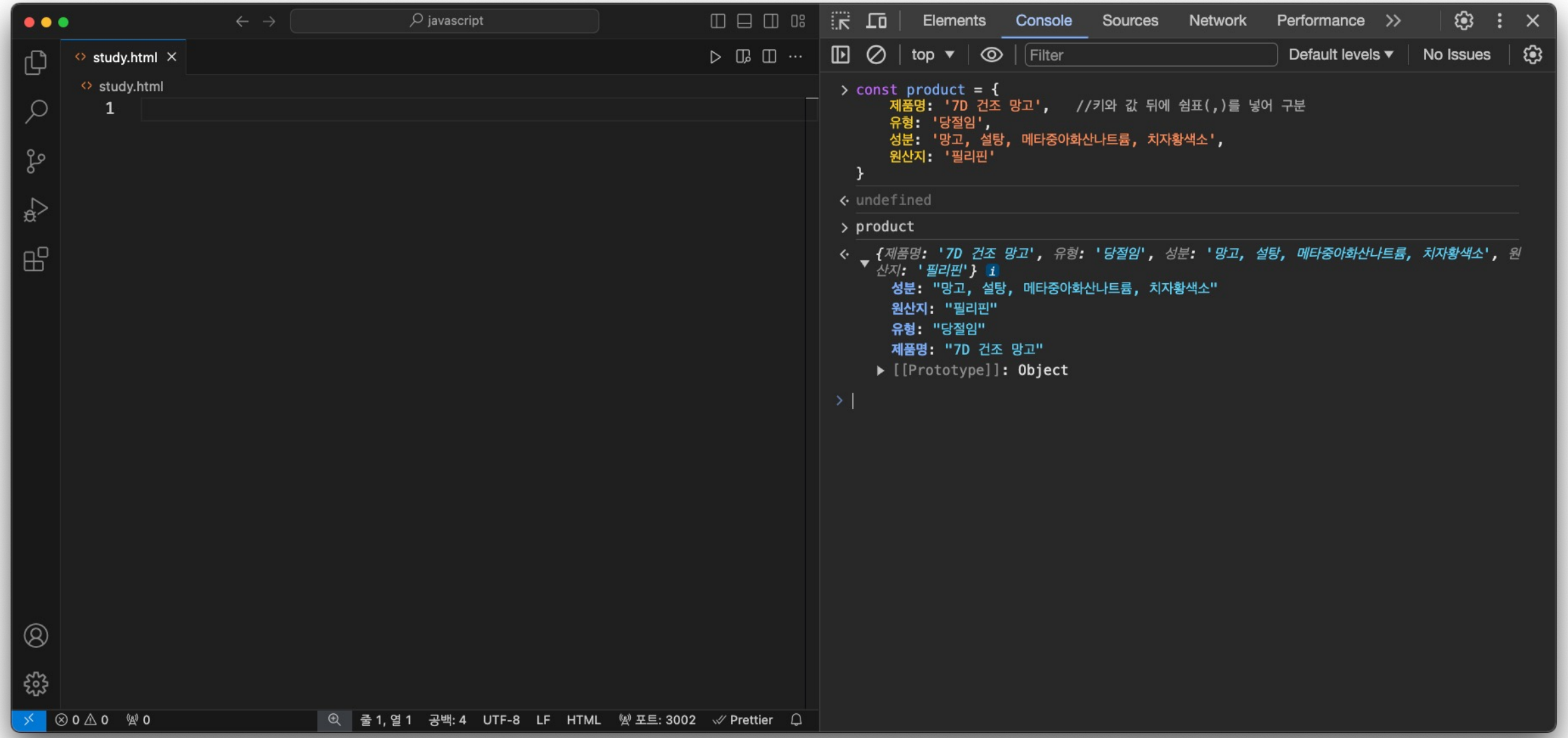
산책하기( )
밥먹기( )

메소드

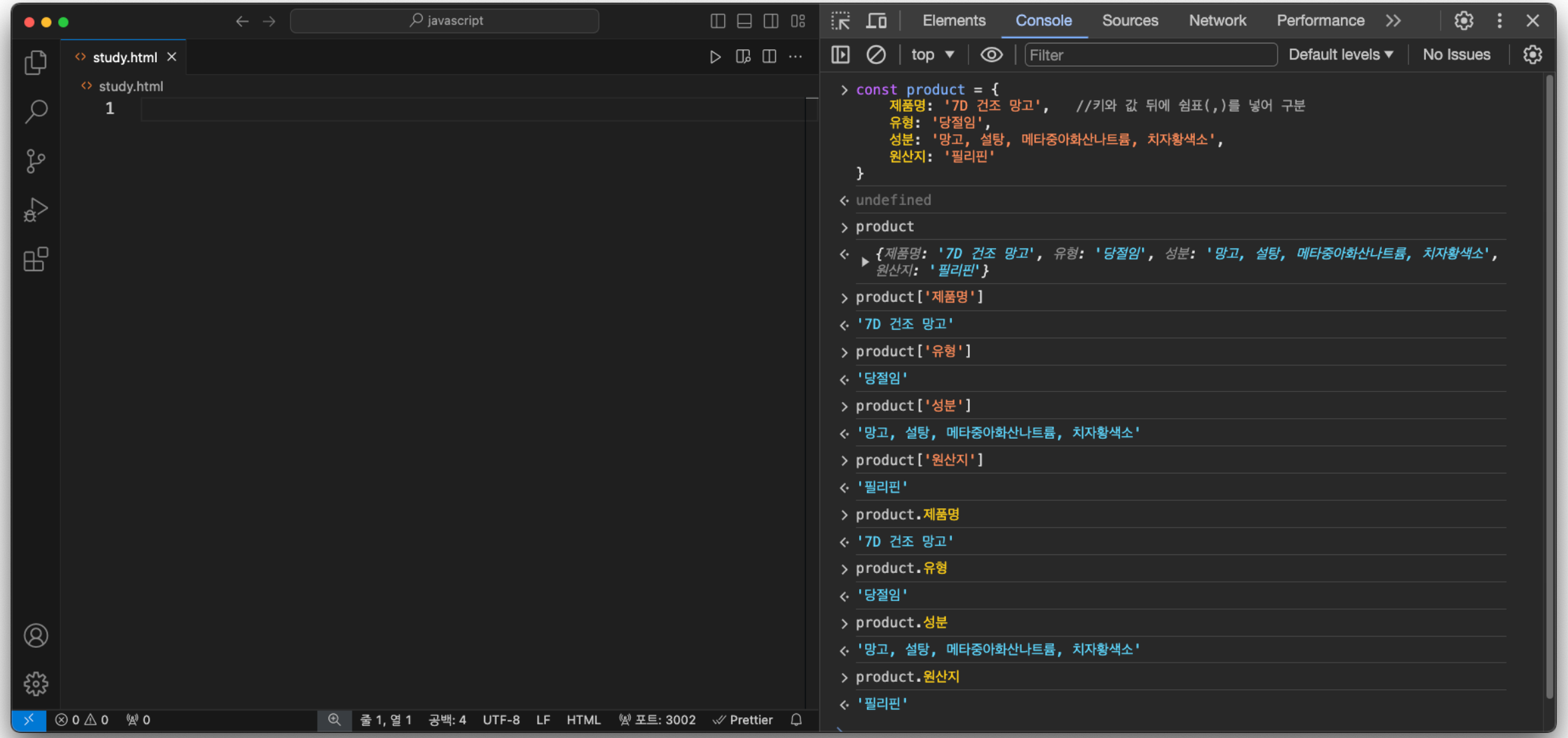
chap 6 객체


javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
> typeof([])
'object'
> const array = ['사과', '바나나', '망고', '딸기']
undefined
> array
(4) ['사과', '바나나', '망고', '딸기']
0: "사과"
1: "바나나"
2: "망고"
3: "딸기"
length: 4
[[Prototype]]: Array(0)
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

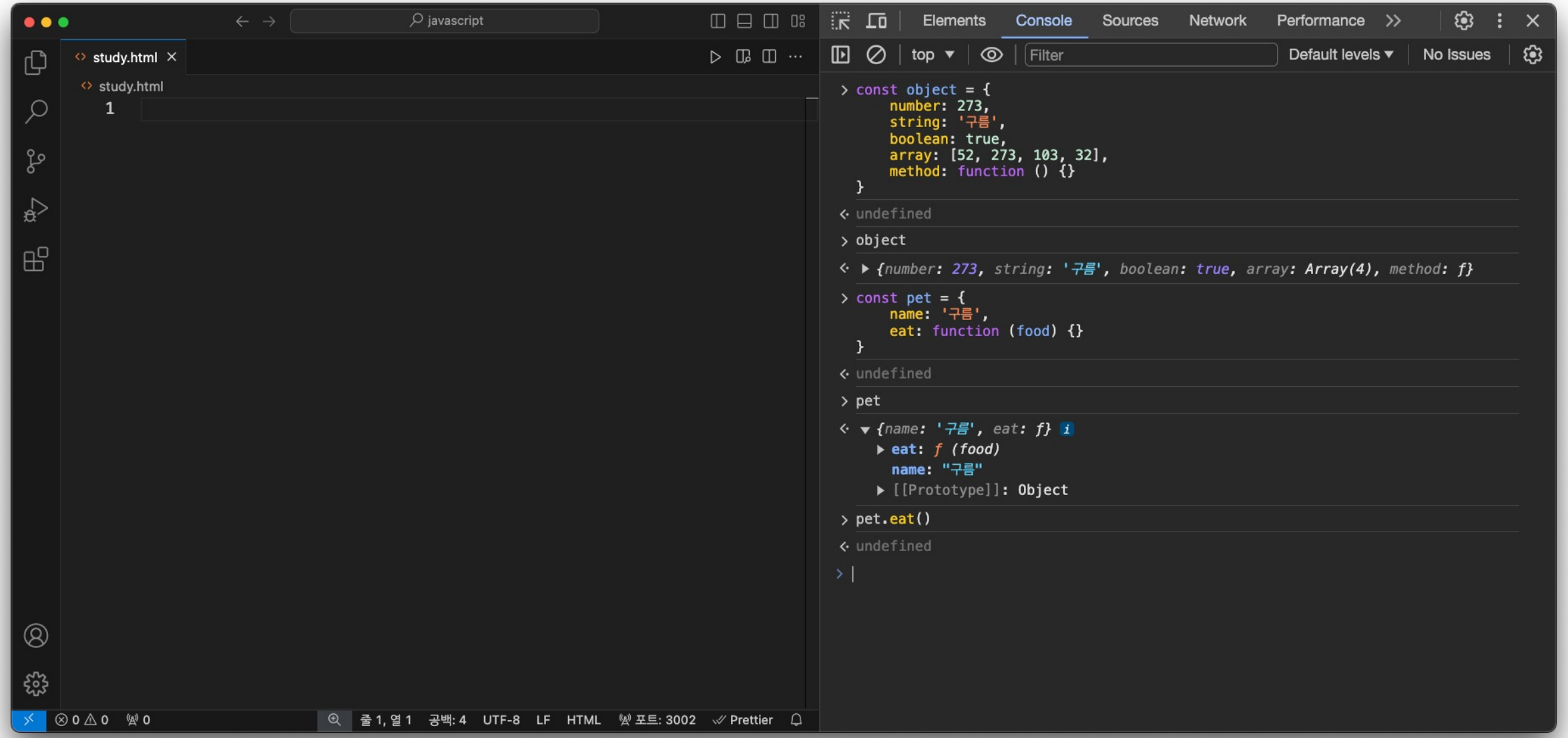
chap 6 객체


javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
> const product = {
제품명: '7D 건조 망고', //키와 값 뒤에 쉼표(,)를 넣어 구분
유형: '당절임',
성분: '망고, 설탕, 메타중아화산나트륨, 치자황색소',
원산지: '필리핀'
}
undefined
> product
{제품명: '7D 건조 망고', 유형: '당절임', 성분: '망고, 설탕, 메타중아화산나트륨, 치자황색소', 원산지: '필리핀'}
성분: "망고, 설탕, 메타중아화산나트륨, 치자황색소"
원산지: "필리핀"
유형: "당절임"
제품명: "7D 건조 망고"
[[Prototype]]: Object
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

chap 6 객체

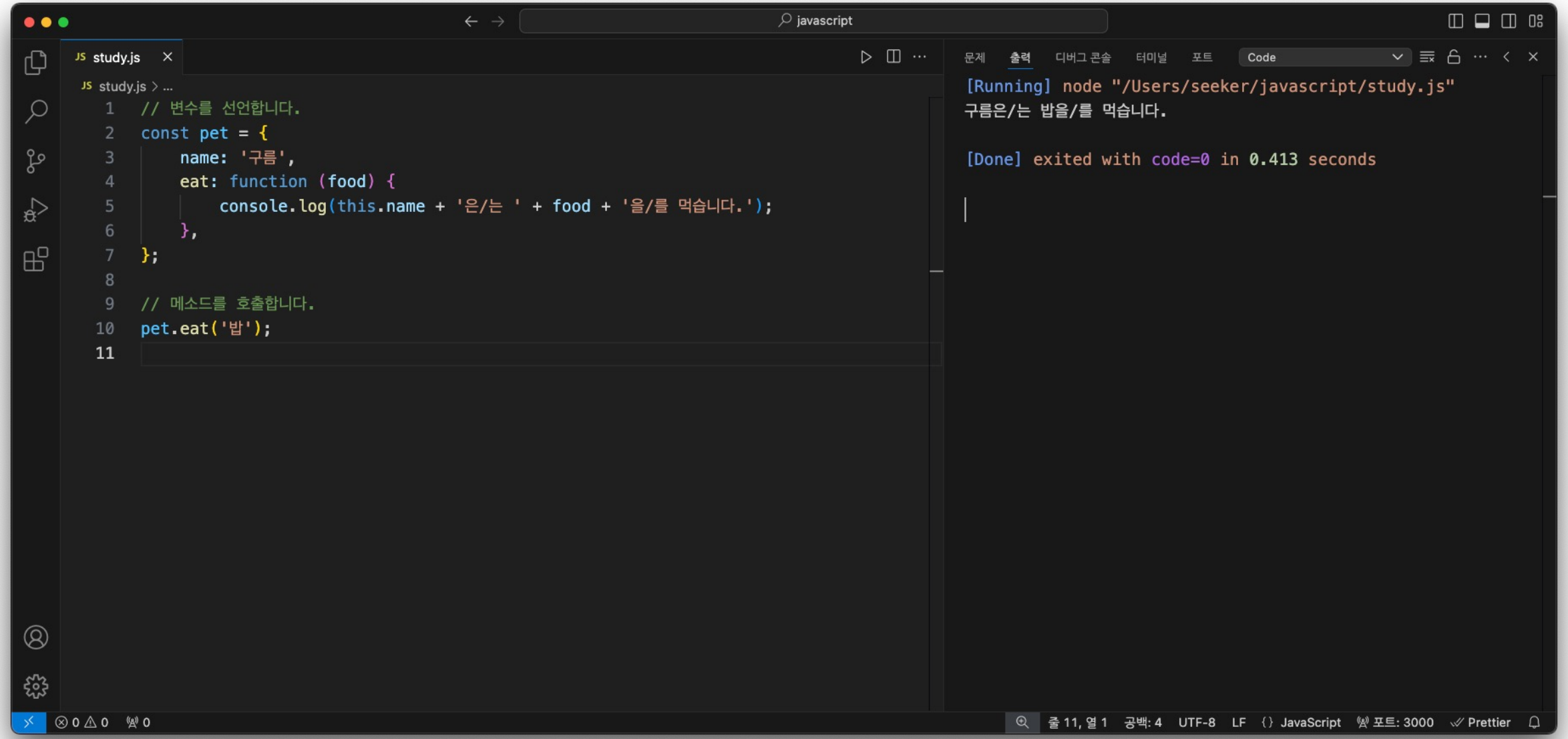

javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
const product = {
제품명: '7D 건조 망고', //키와 값 뒤에 쉼표(,)를 넣어 구분
유형: '당절임',
성분: '망고, 설탕, 메타중아화산나트륨, 치자황색소',
원산지: '필리핀'
}
undefined
product
{제품명: '7D 건조 망고', 유형: '당절임', 성분: '망고, 설탕, 메타중아화산나트륨, 치자황색소', 원산지: '필리핀'}
product['제품명']
'7D 건조 망고'
product['유형']
'당절임'
product['성분']
'망고, 설탕, 메타중아화산나트륨, 치자황색소'
product['원산지']
'필리핀'
product.제품명
'7D 건조 망고'
product.유형
'당절임'
product.성분
'망고, 설탕, 메타중아화산나트륨, 치자황색소'
product.원산지
'필리핀'
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

chap 6 속성과 메소드 > 구분하기



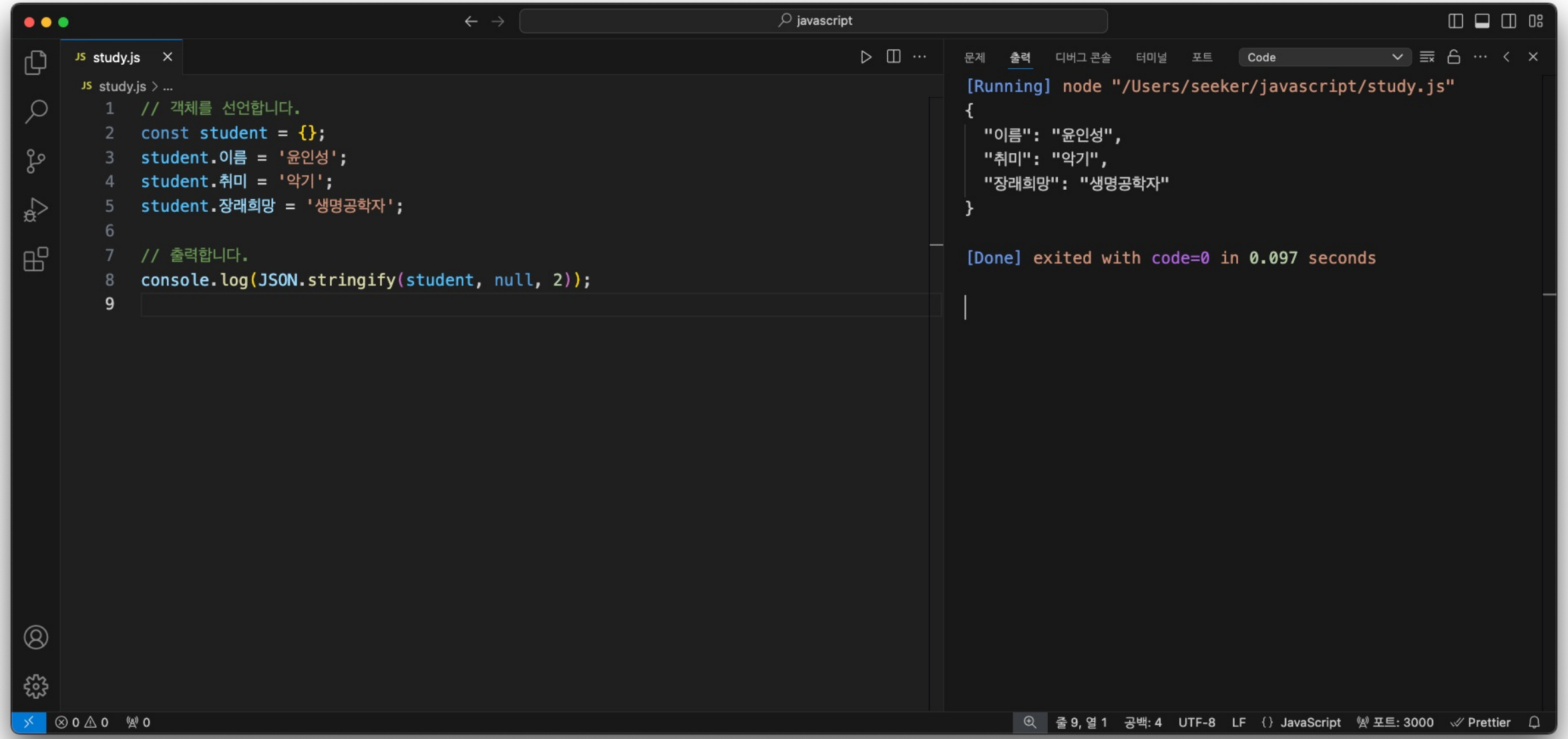
chap 6

속성과 메소드 > 메소드 내부에서 this 키워드 사용



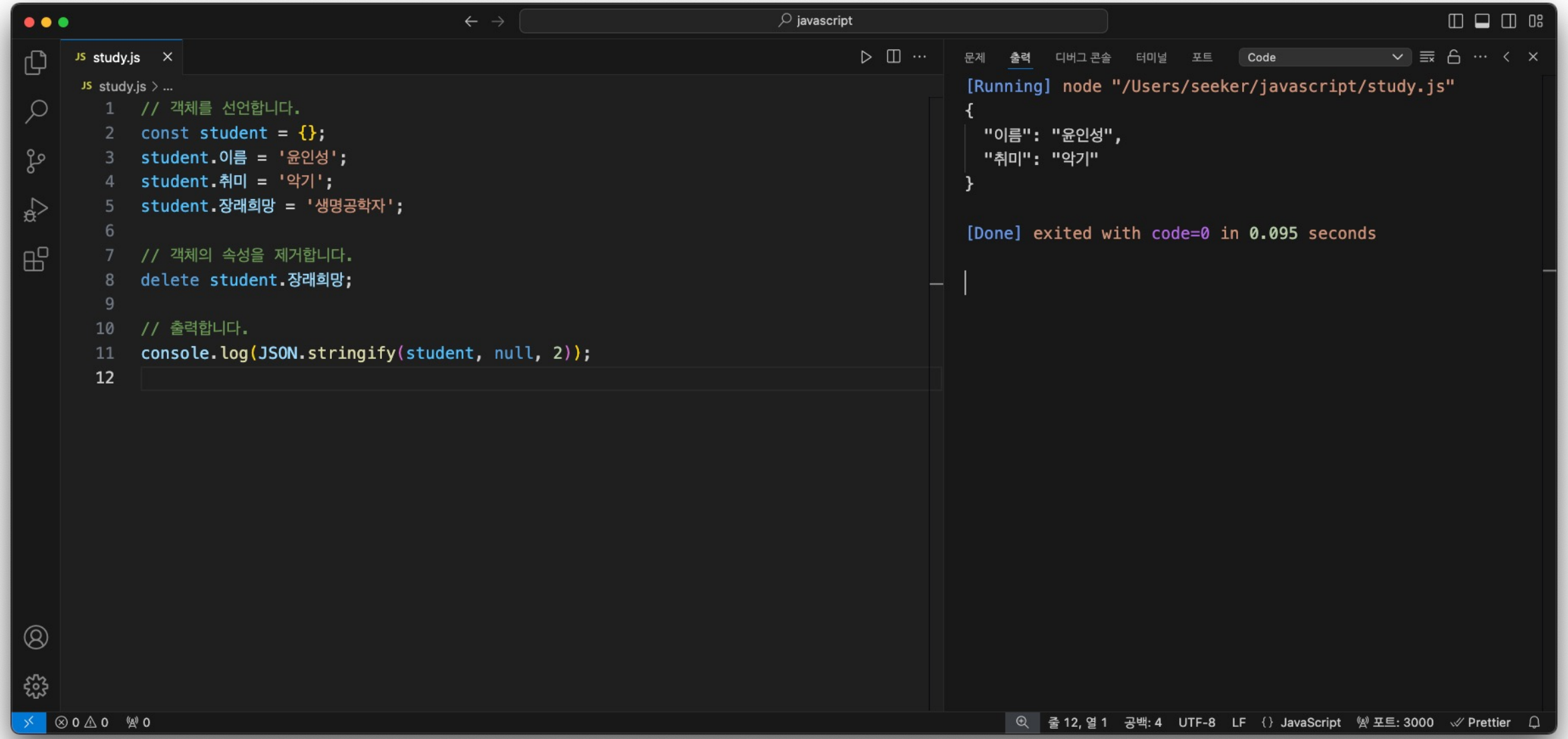
chap 6

동적으로 객체 속성 추가



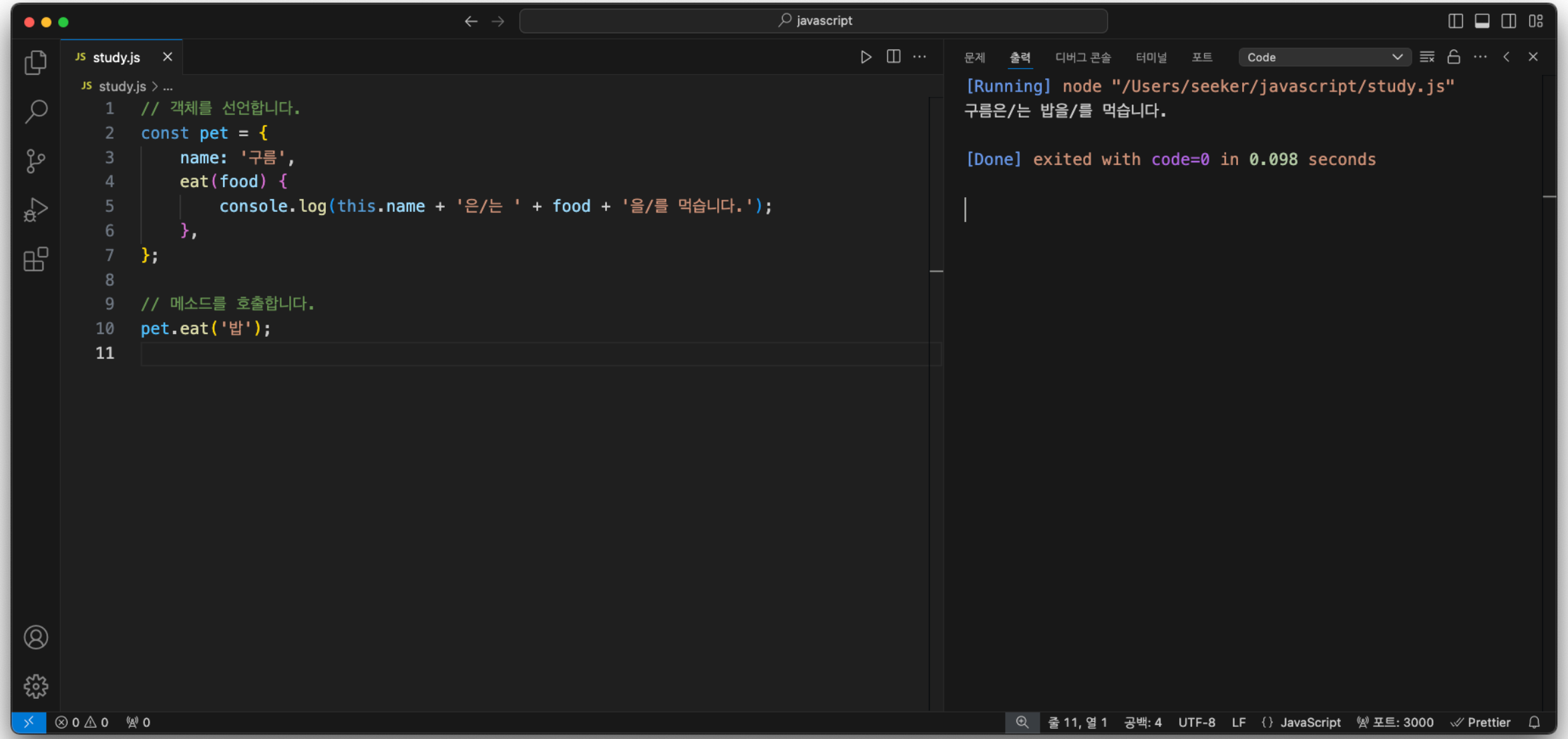
chap 6

동적으로 객체 속성 제거



chap 6

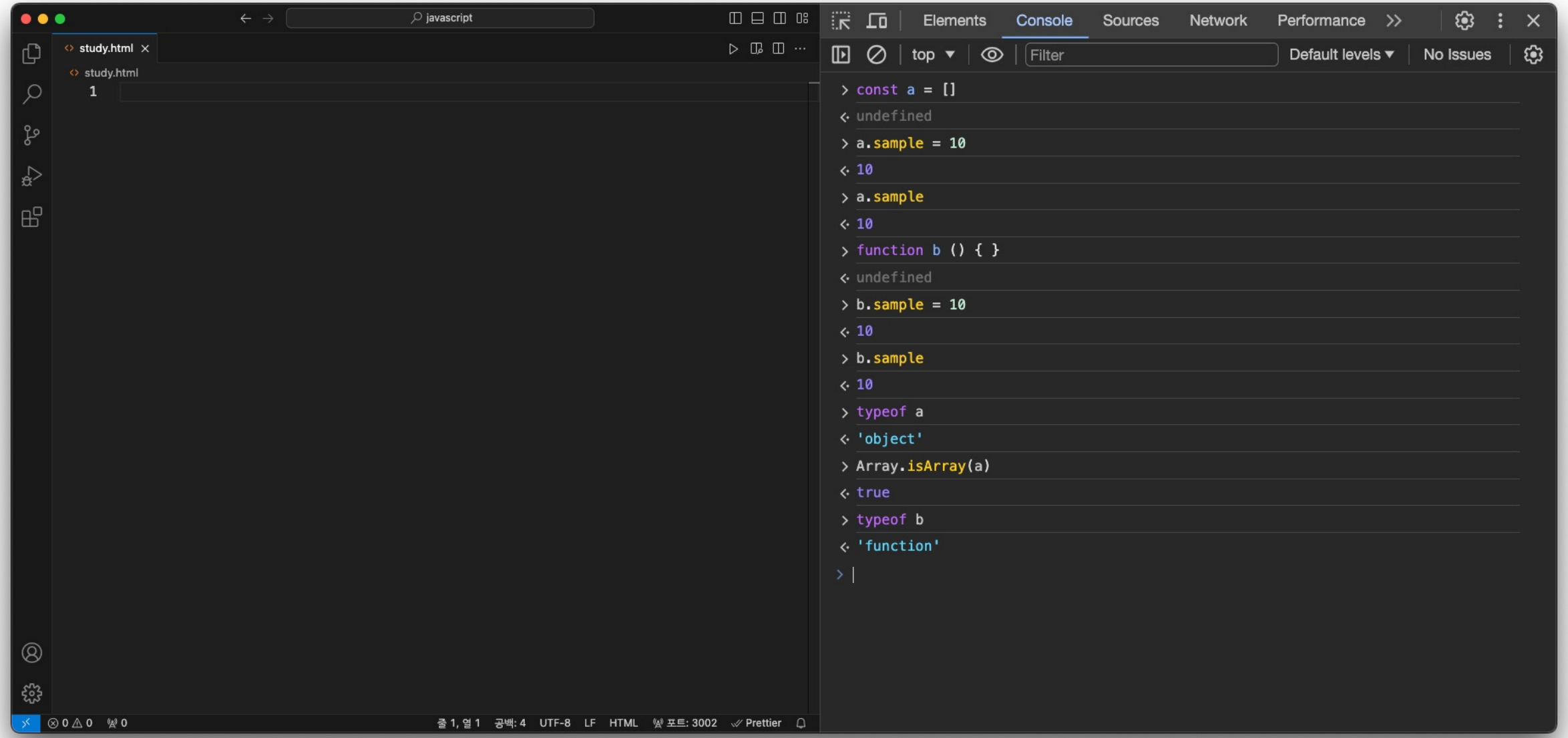
메소드 간단 선언 구문



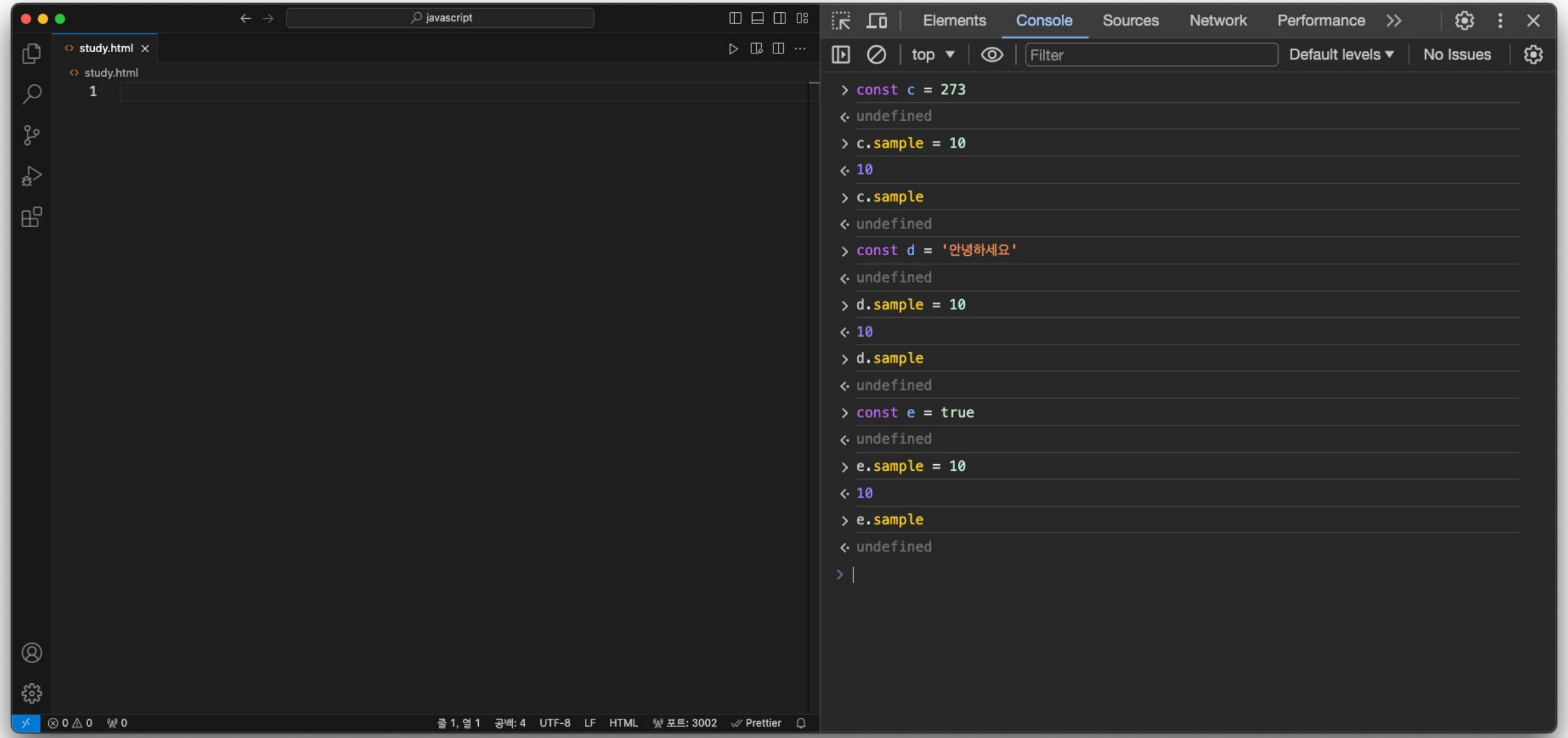
속성과 메소드

chap 6

객체 자료형

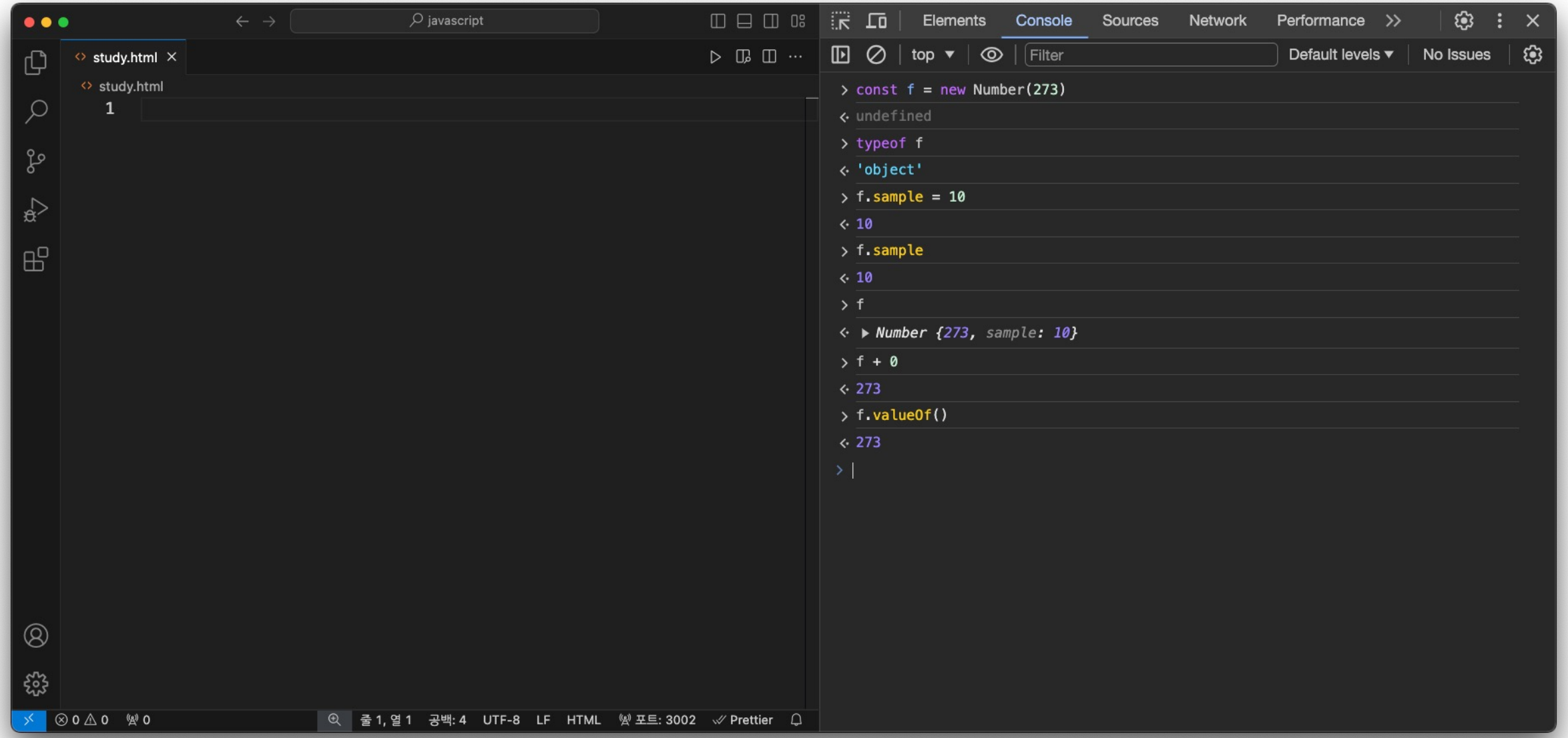

javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
> const a = []
< undefined
> a.sample = 10
< 10
> a.sample
< 10
> function b () { }
< undefined
> b.sample = 10
< 10
> b.sample
< 10
> typeof a
< 'object'
> Array.isArray(a)
< true
> typeof b
< 'function'
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

chap 6 기본 자료형


javascript
study.html
study.html
1
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
> const c = 273
< undefined
> c.sample = 10
< 10
> c.sample
< undefined
> const d = '안녕하세요'
< undefined
> d.sample = 10
< 10
> d.sample
< undefined
> const e = true
< undefined
> e.sample = 10
< 10
> e.sample
< undefined
줄 1, 열 1
공백: 4
UTF-8
LF
HTML
포트: 3002
Prettier

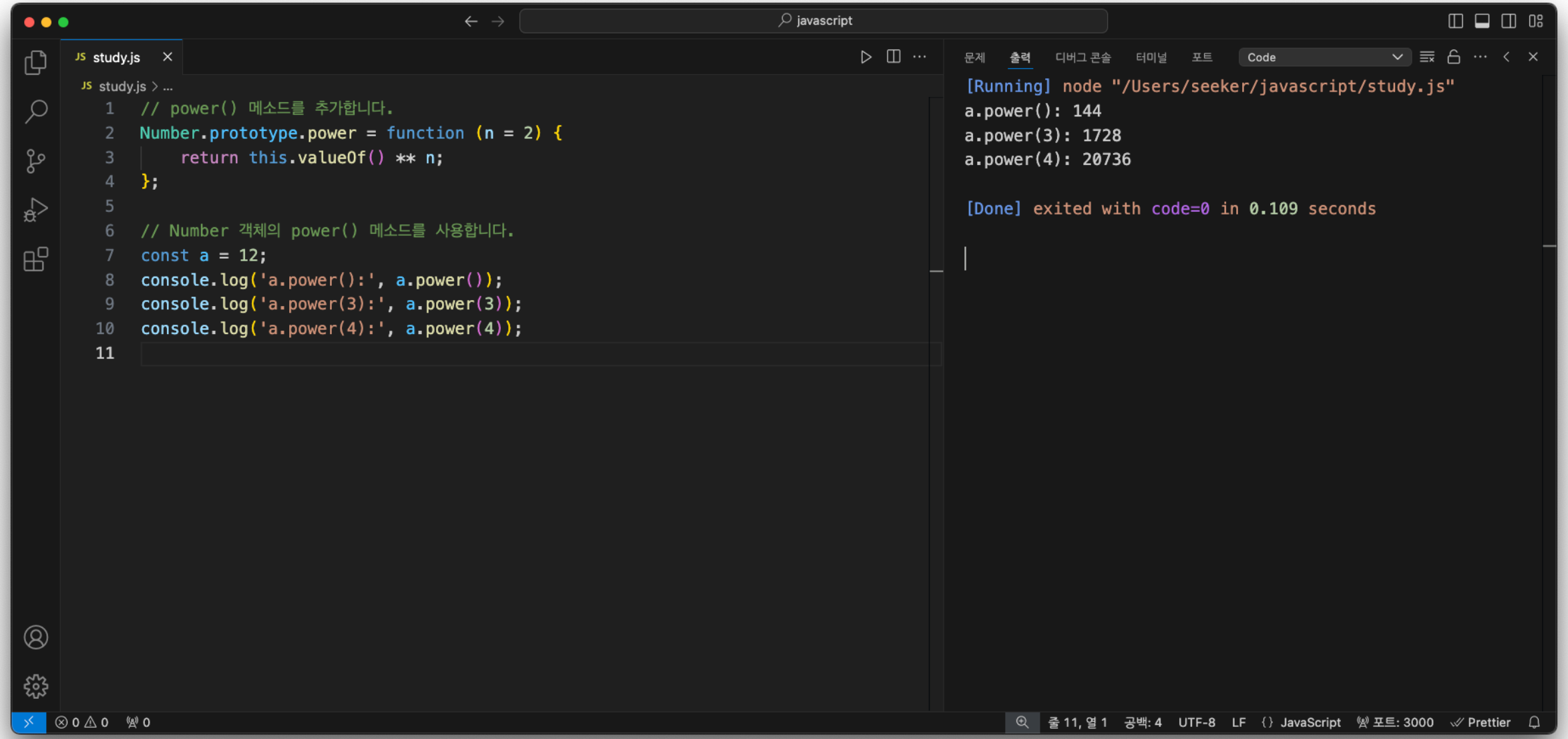
chap 6

기본 자료형을 객체로 선언하기



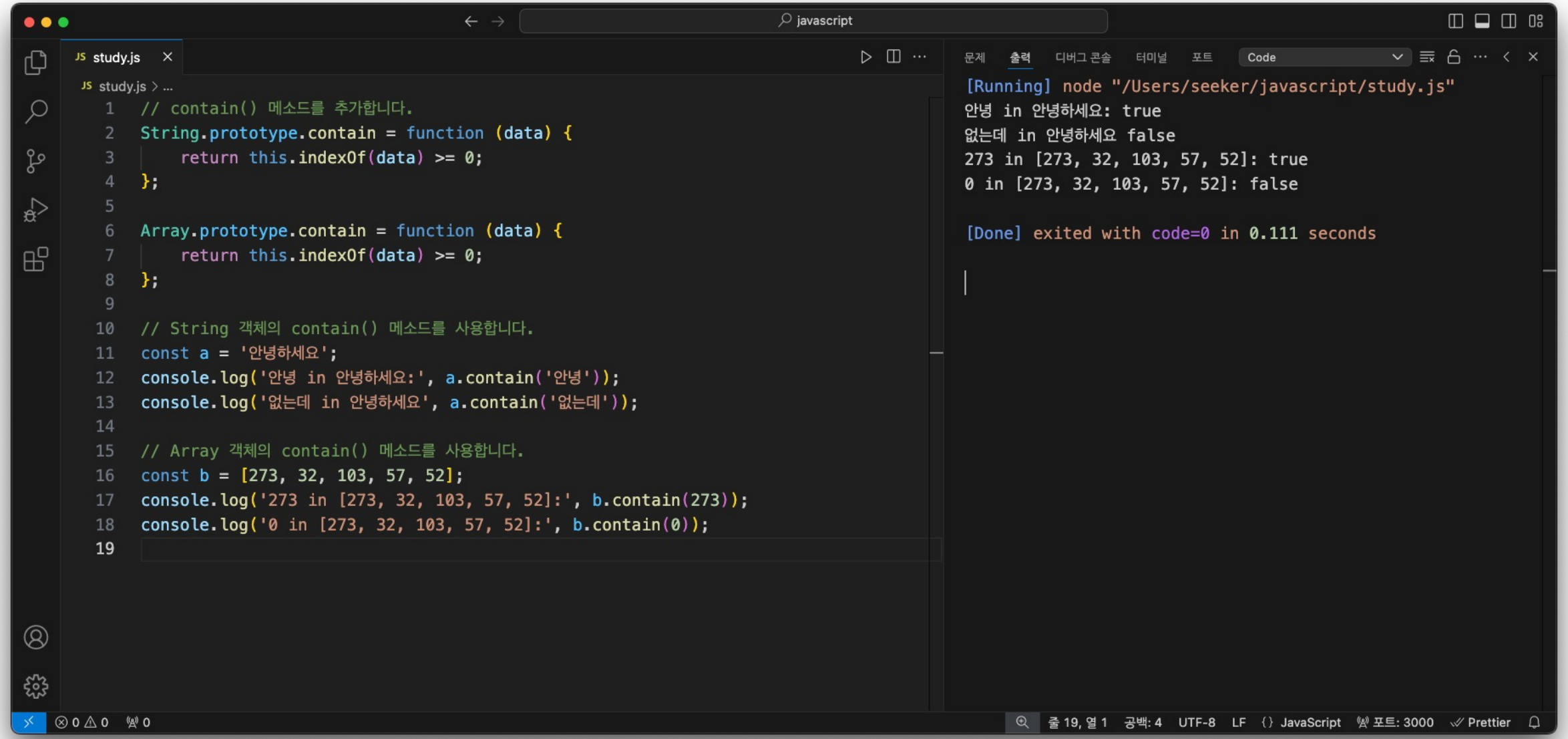
chap 6

프로토타입으로 메소드 추가하기

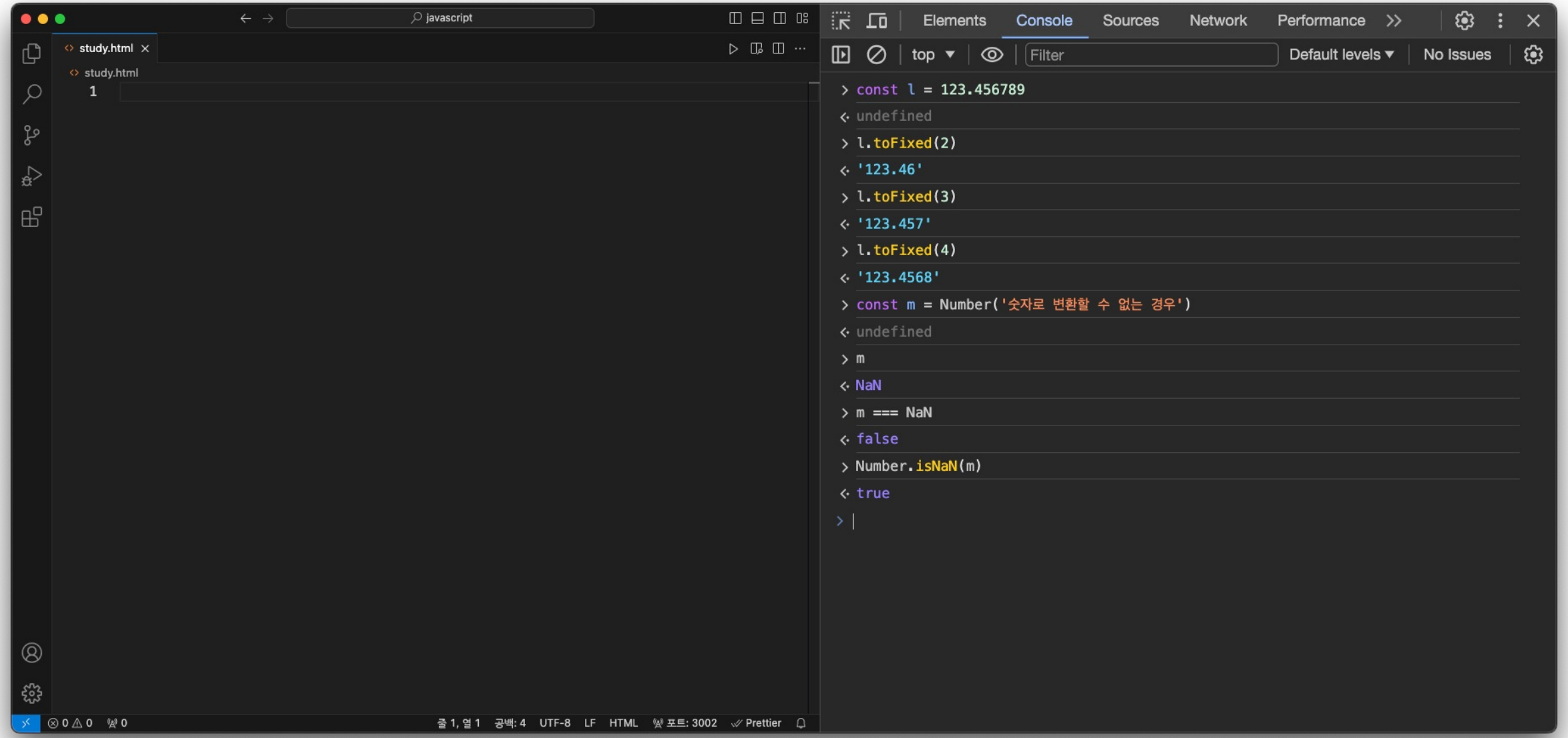


chap 6

프로토타입으로 메소드 추가하기

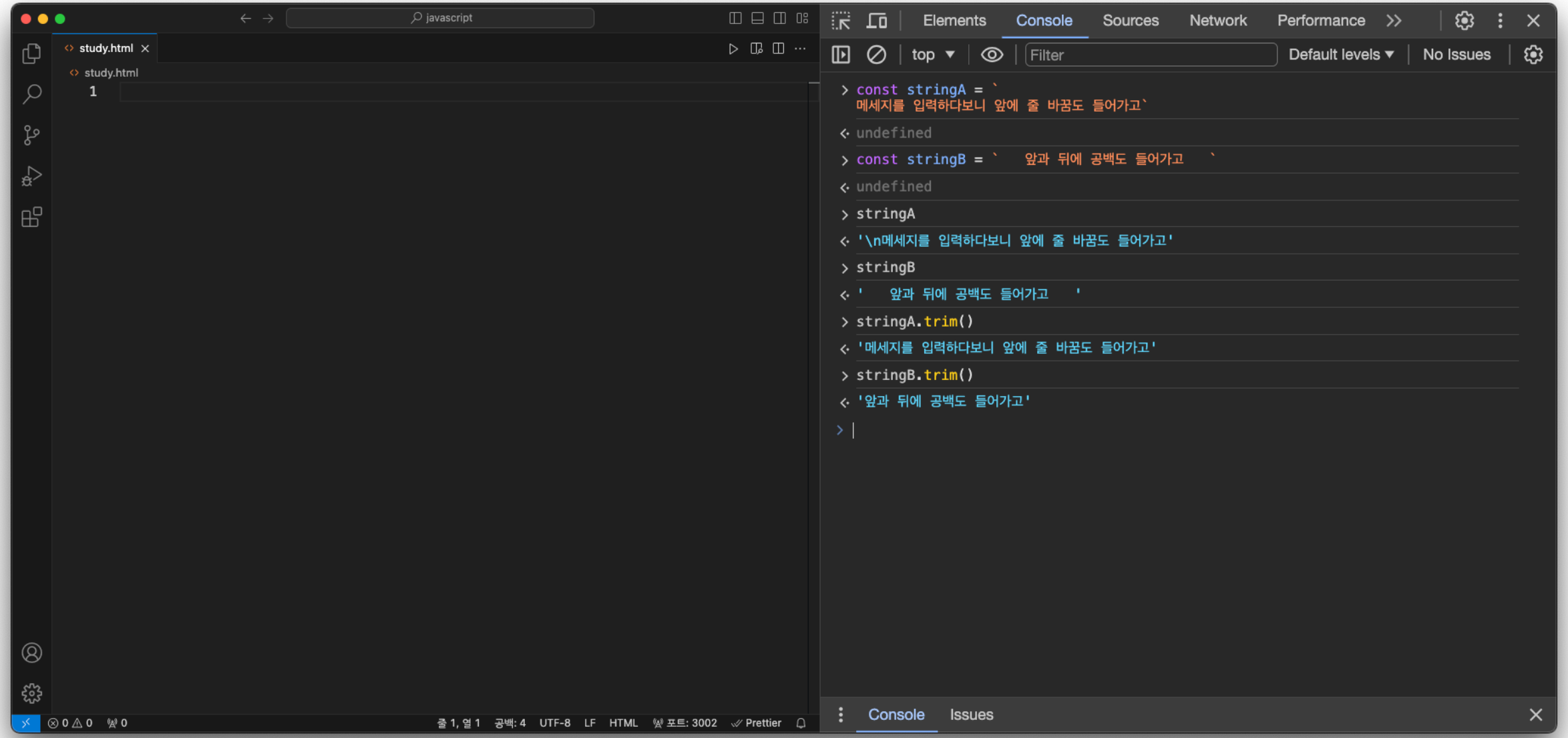


chap 6 Number 객체



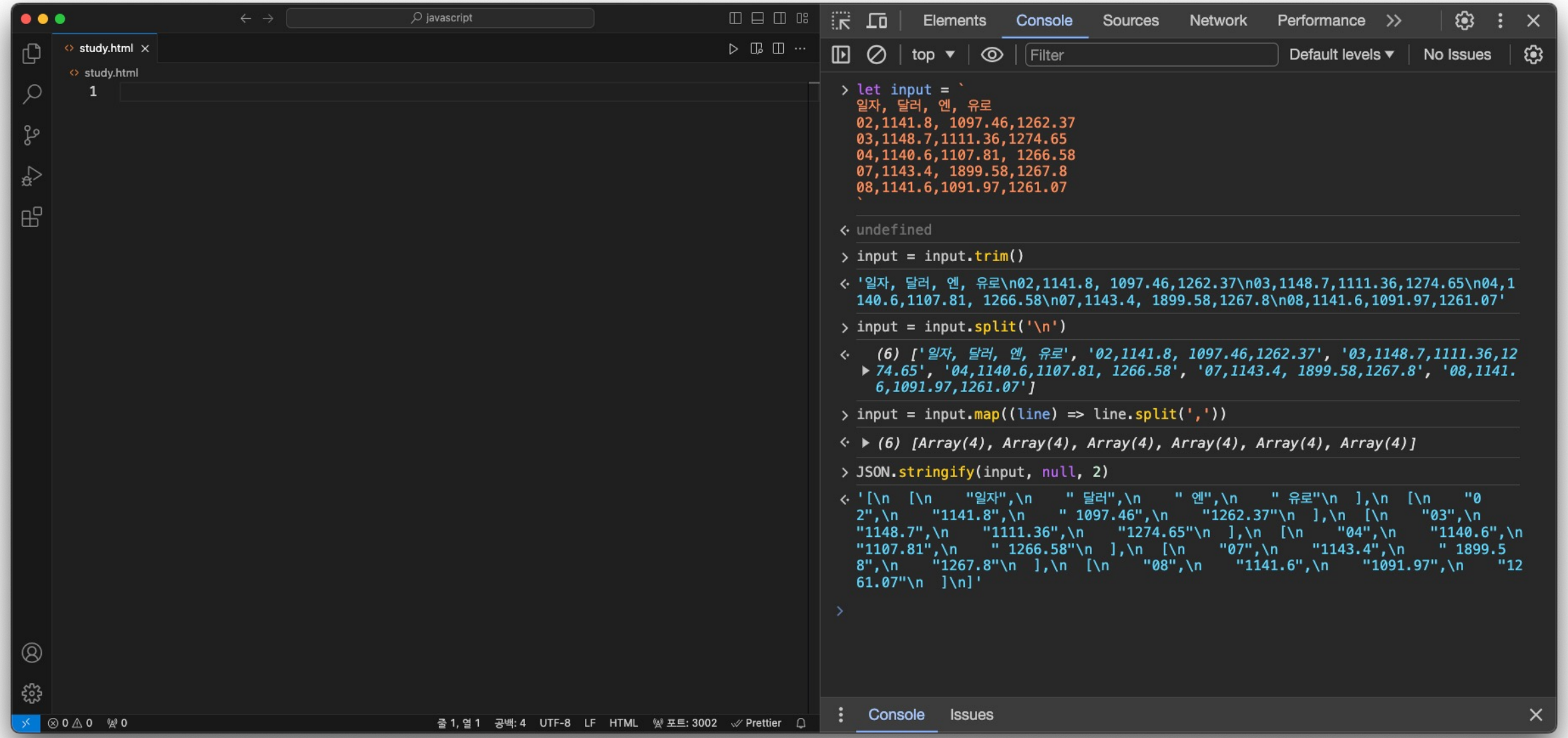
chap 6

String 객체 > trim() : 문자열 양쪽 끝의 공백 없애기

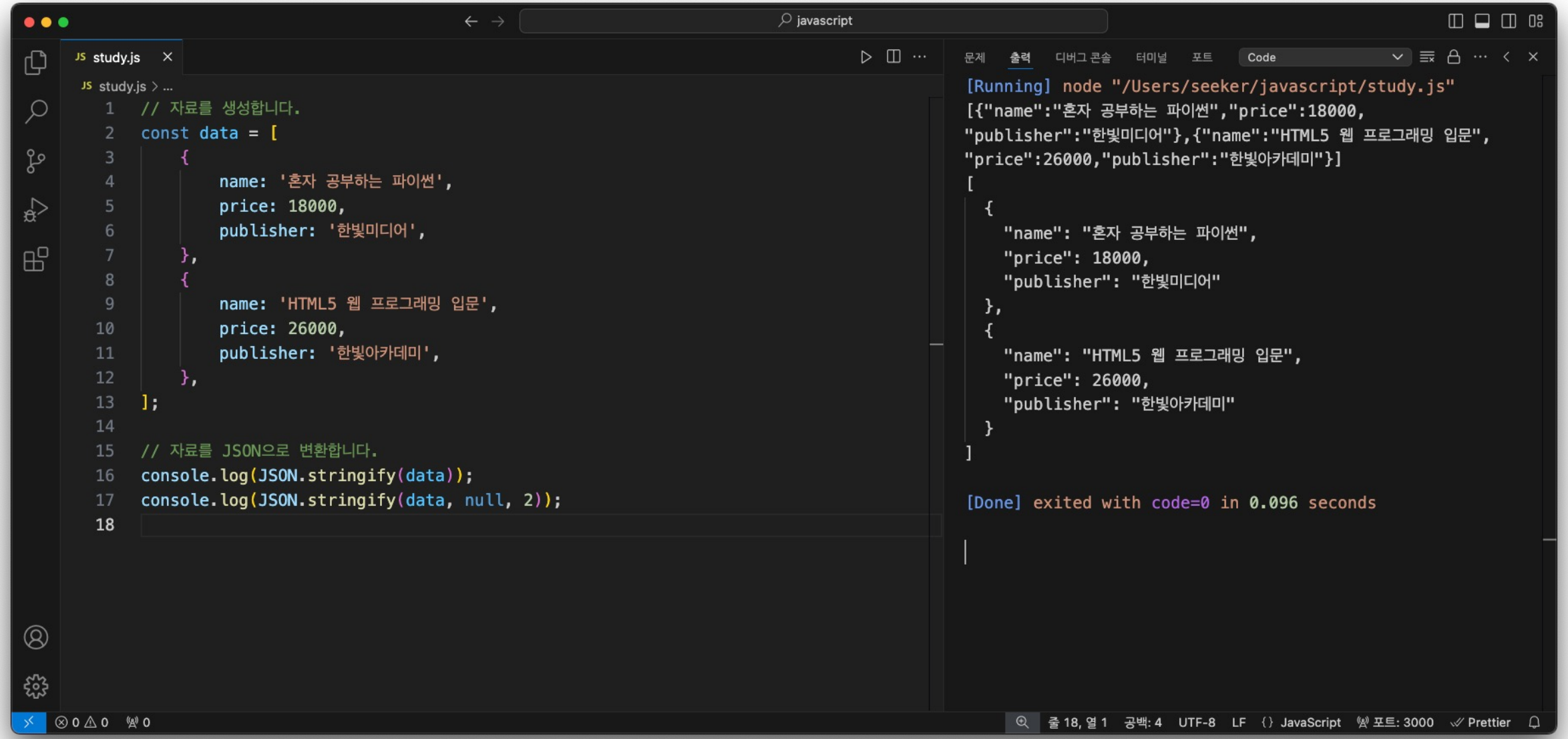


chap 6

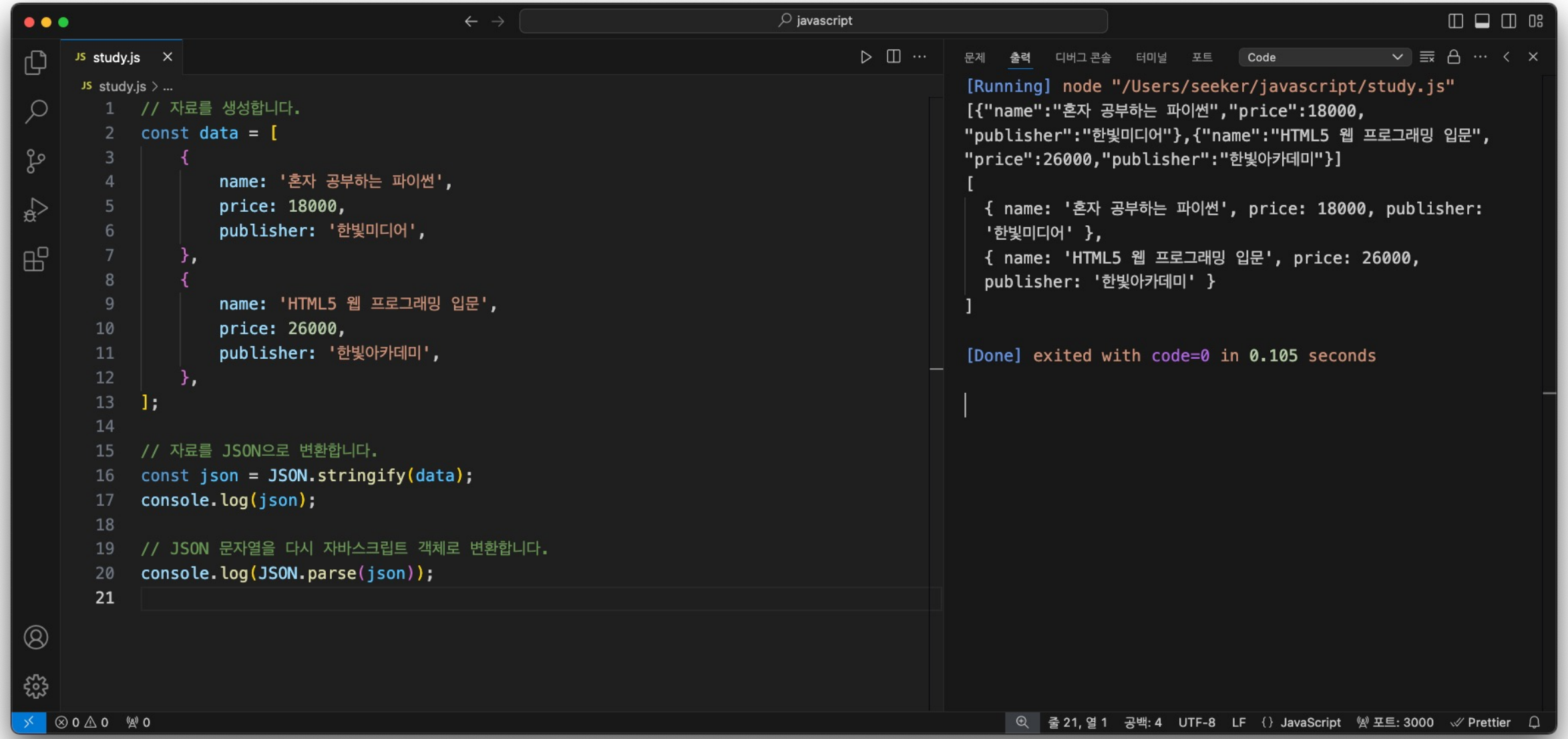
String 객체 > split() : 문자열을 특정 기호로 자르기



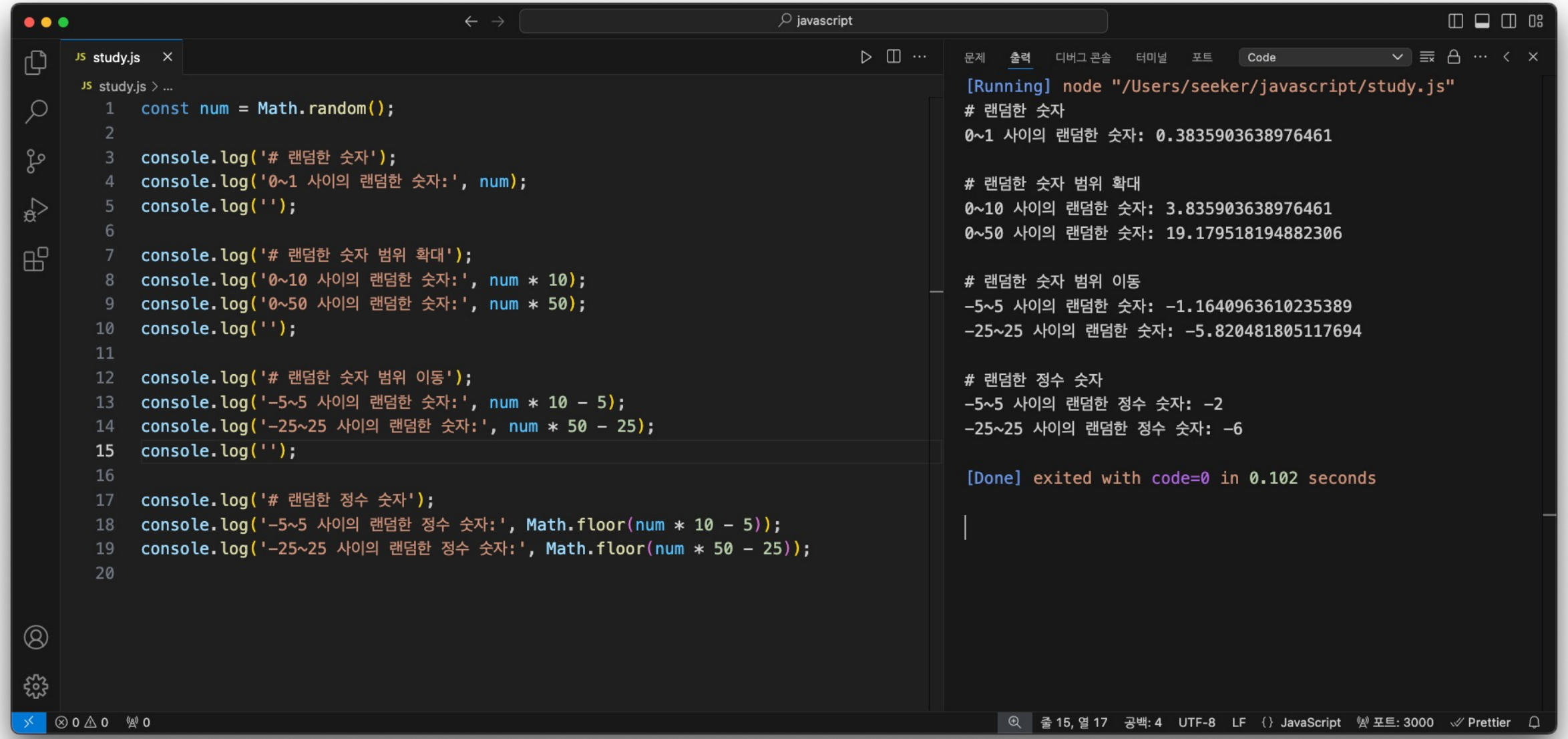
chap 6 JSON 객체



chap 6 JSON 객체



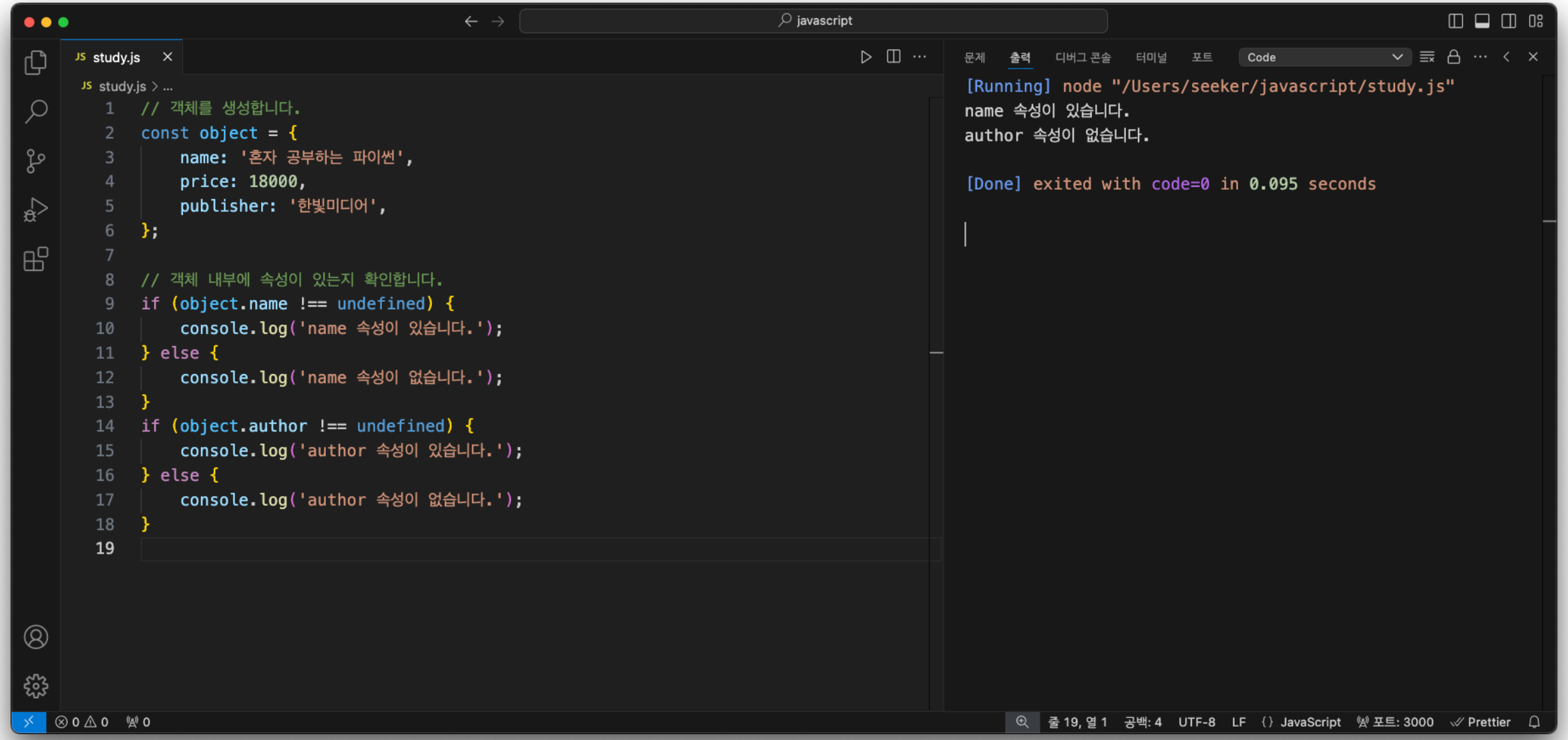
chap 6 Math 객체



객체/배열 고급

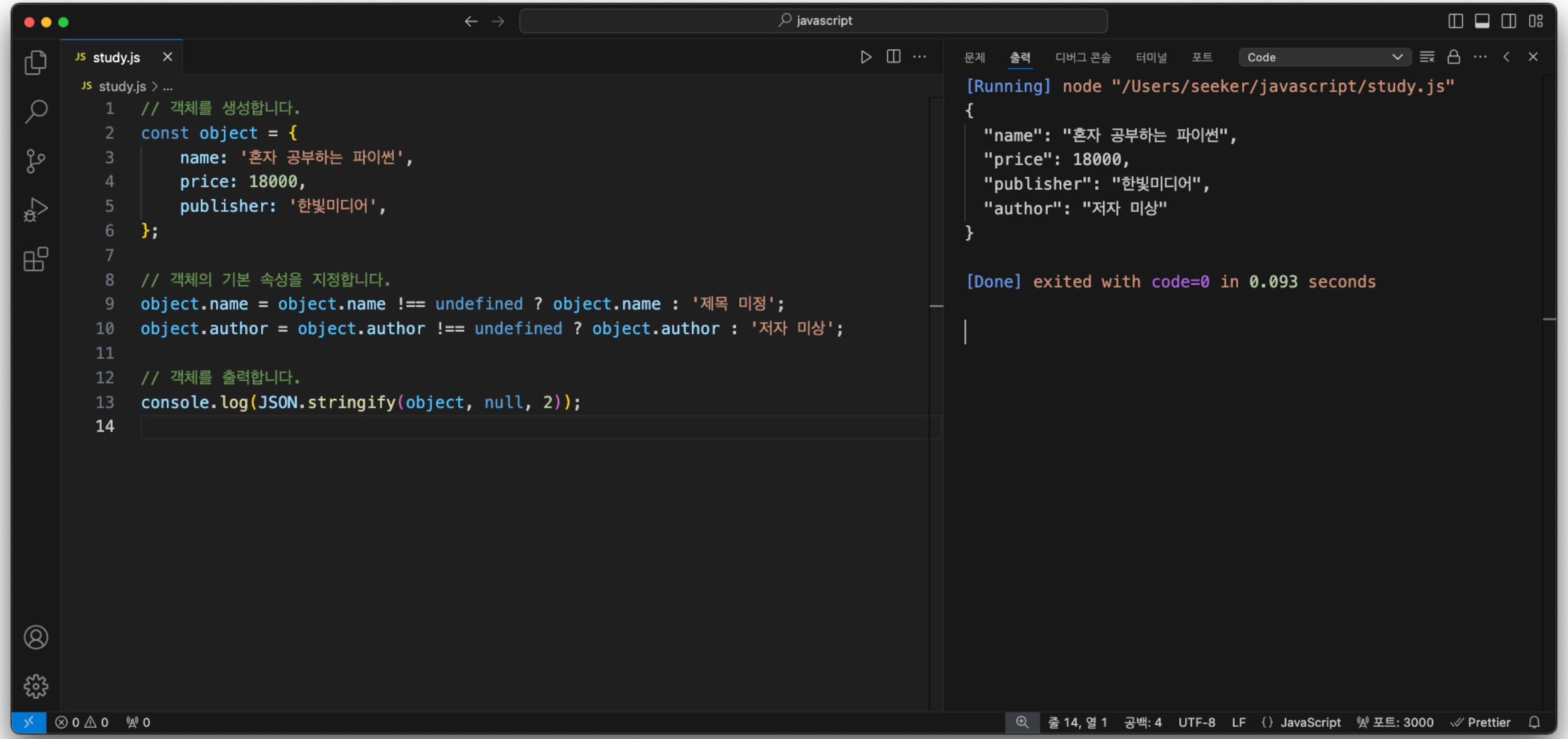
chap 6

속성 존재 여부 확인



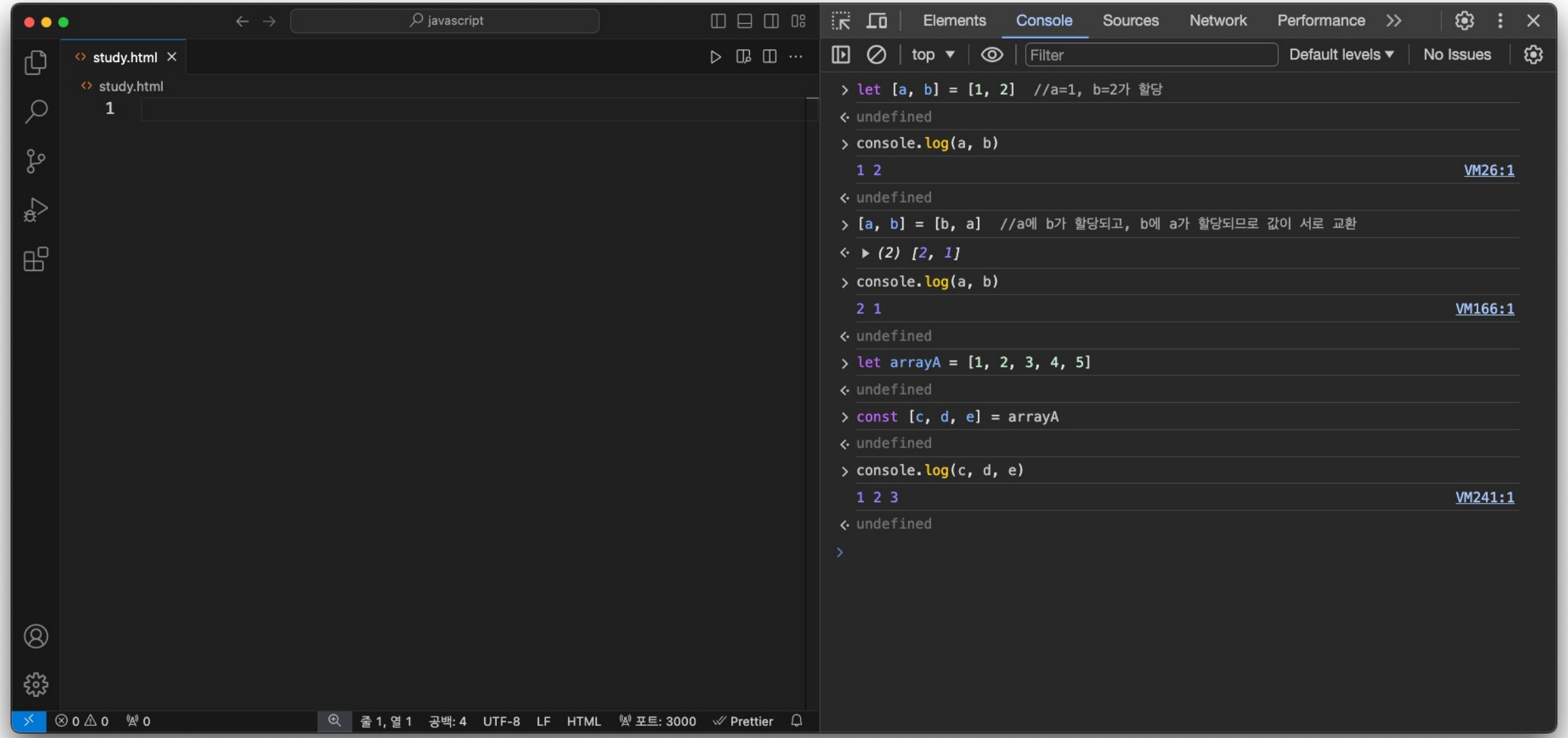
chap 6

속성 존재 여부 확인



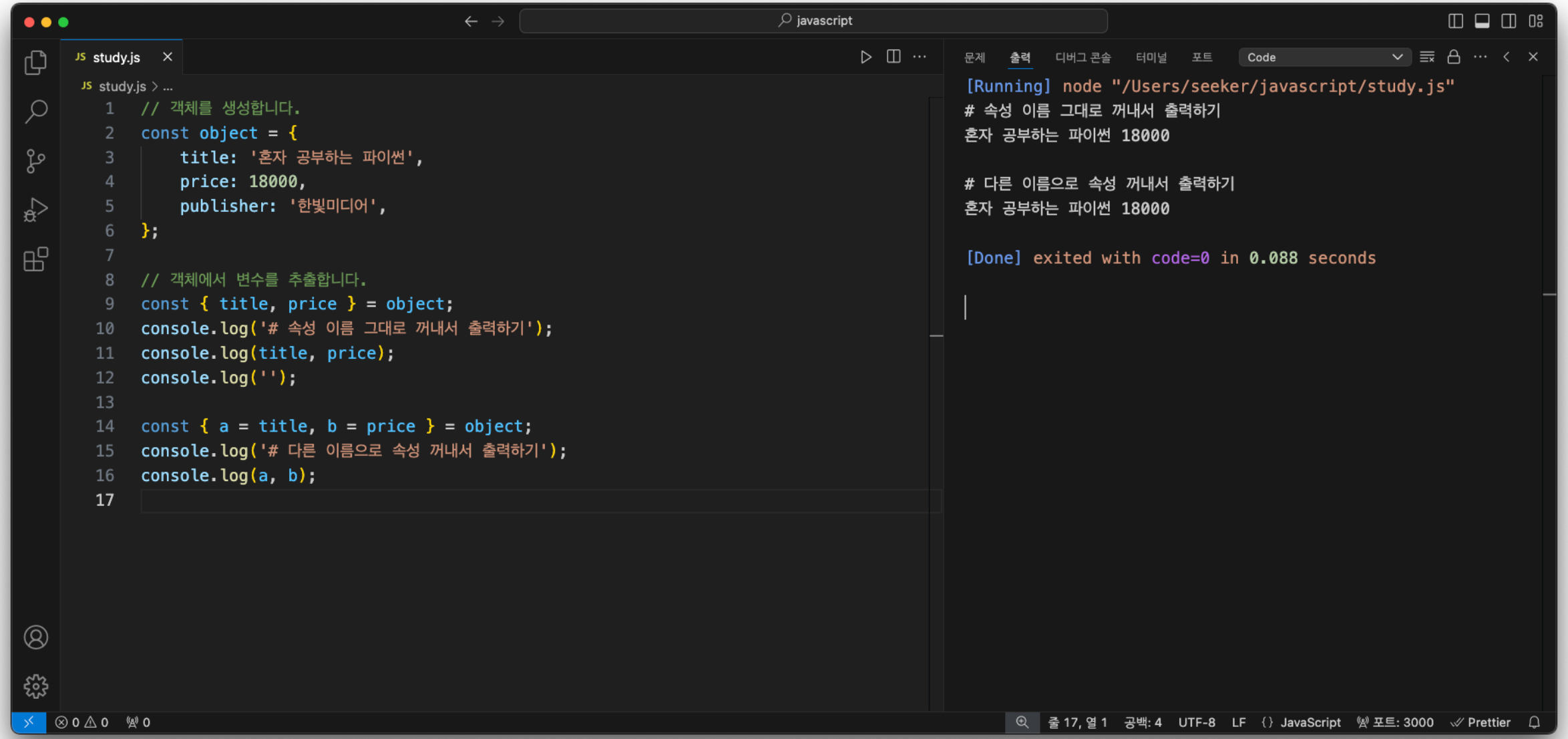
chap 6

배열 기반의 다중 할당



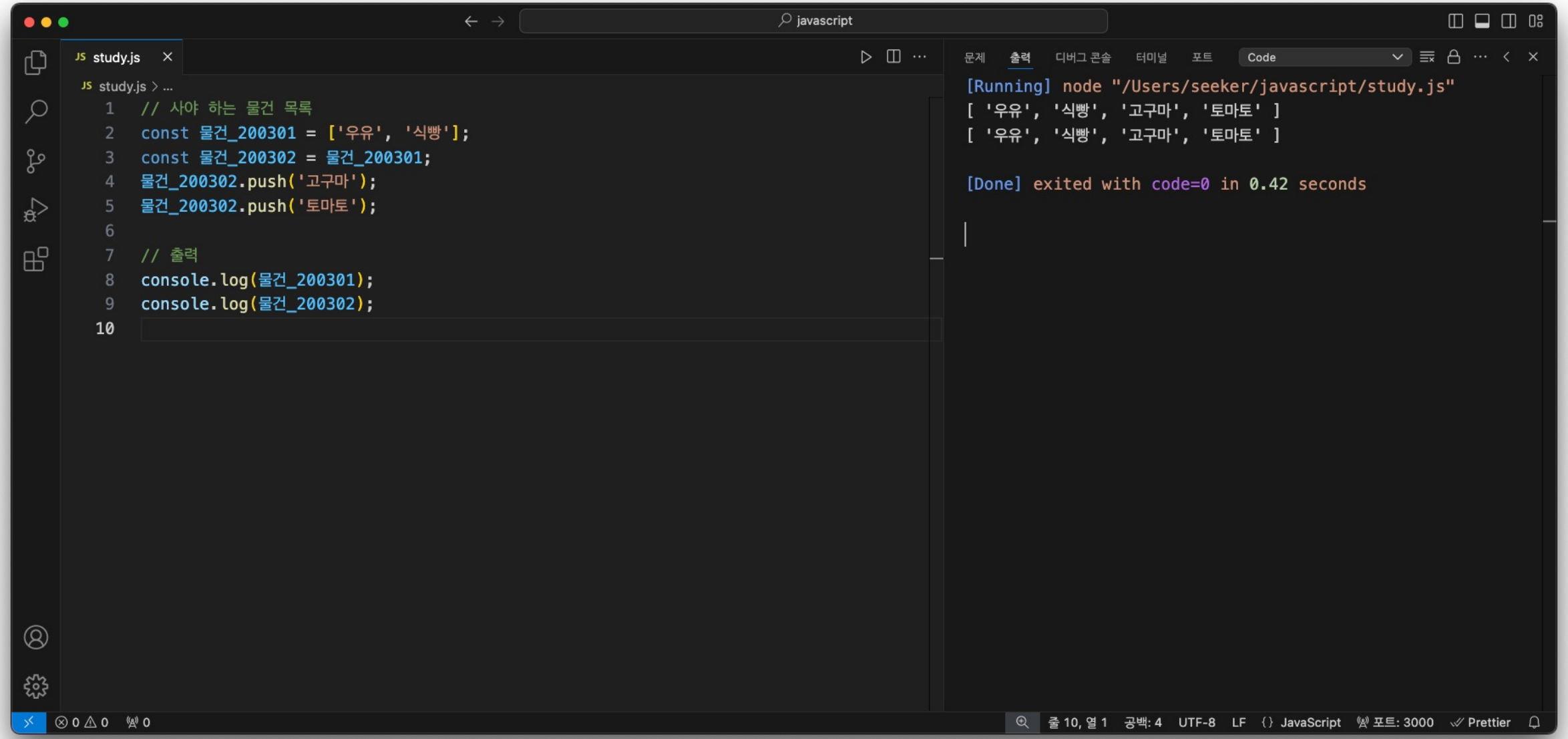
chap 6

객체 기반의 다중 할당



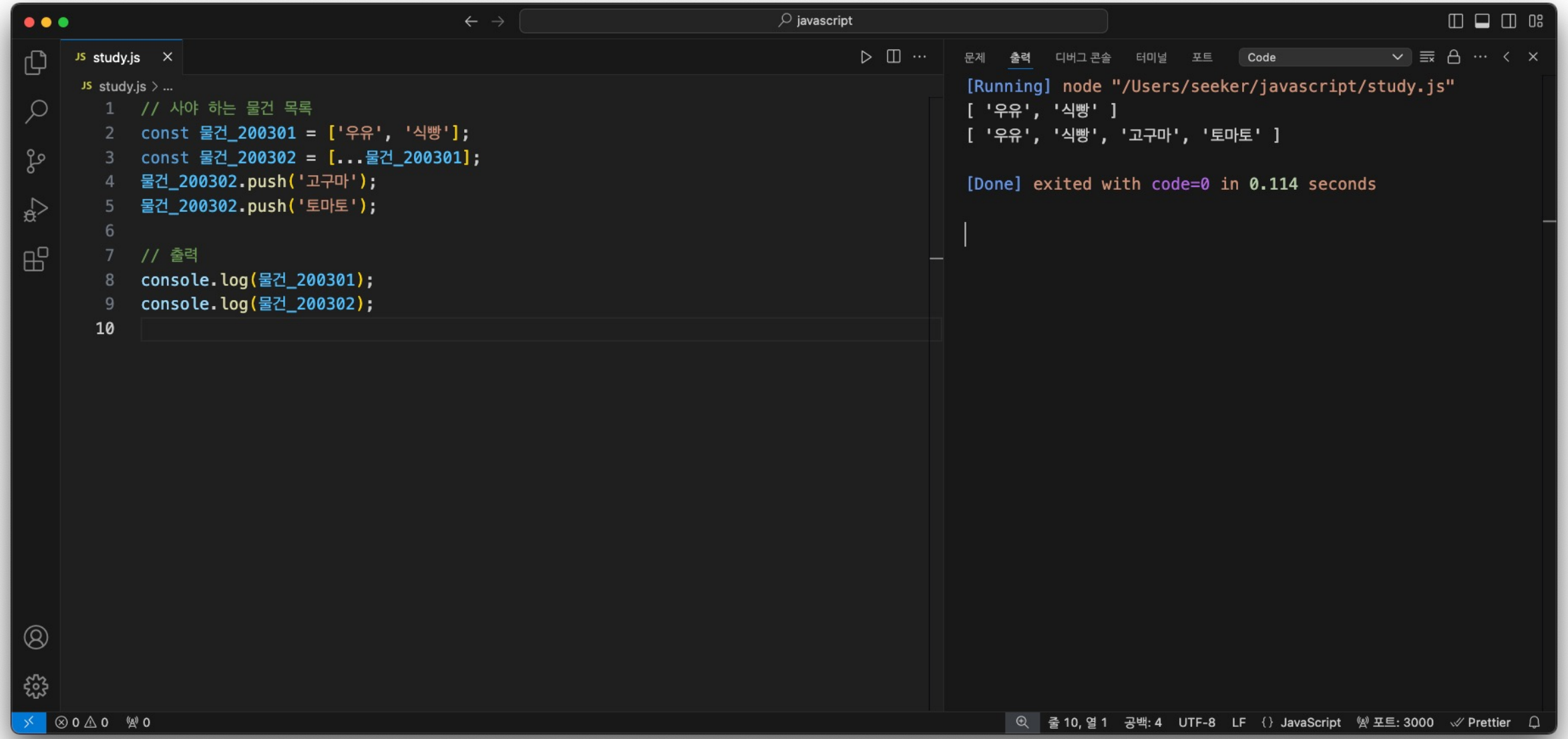
chap 6

배열 전개 연산자



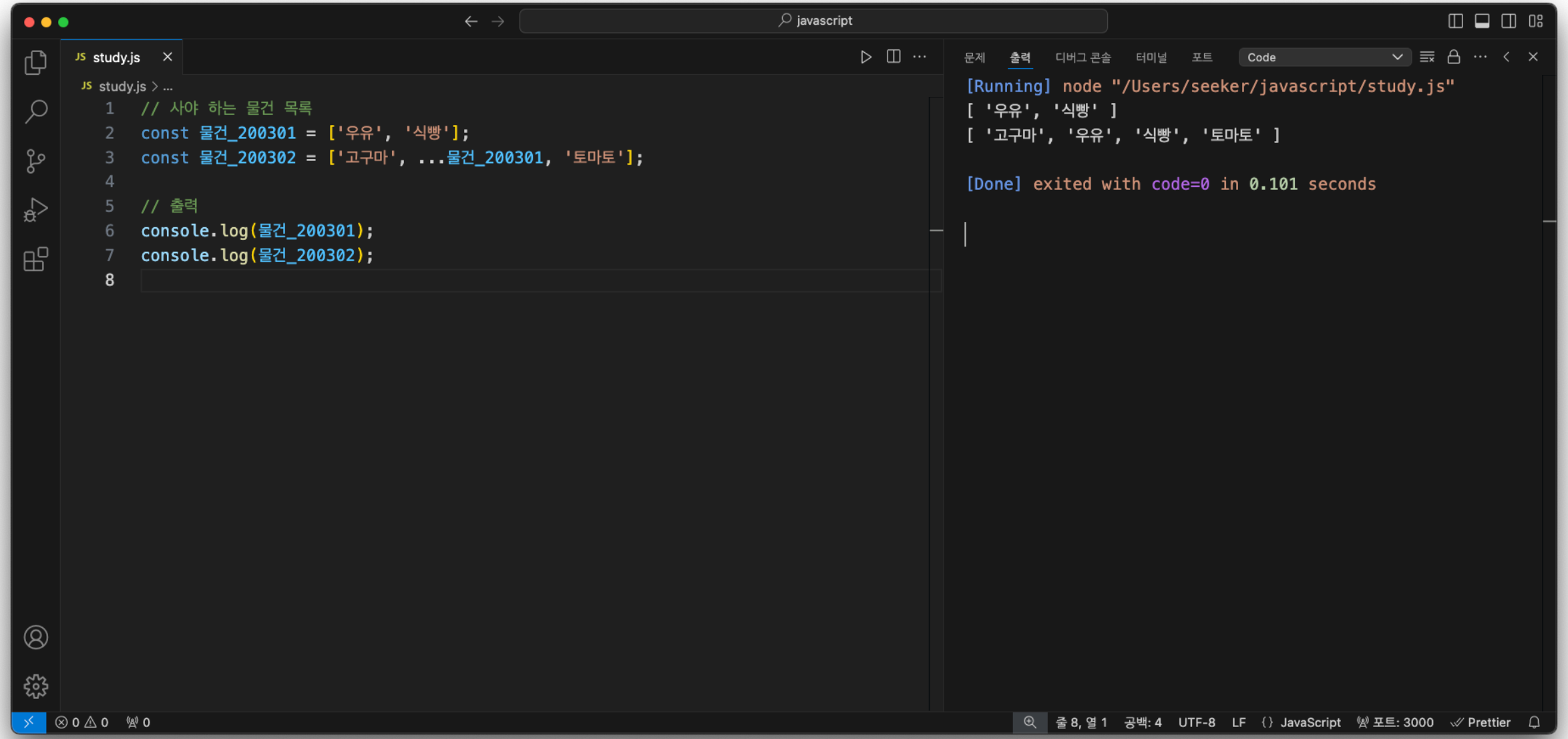
chap 6

배열 전개 연산자



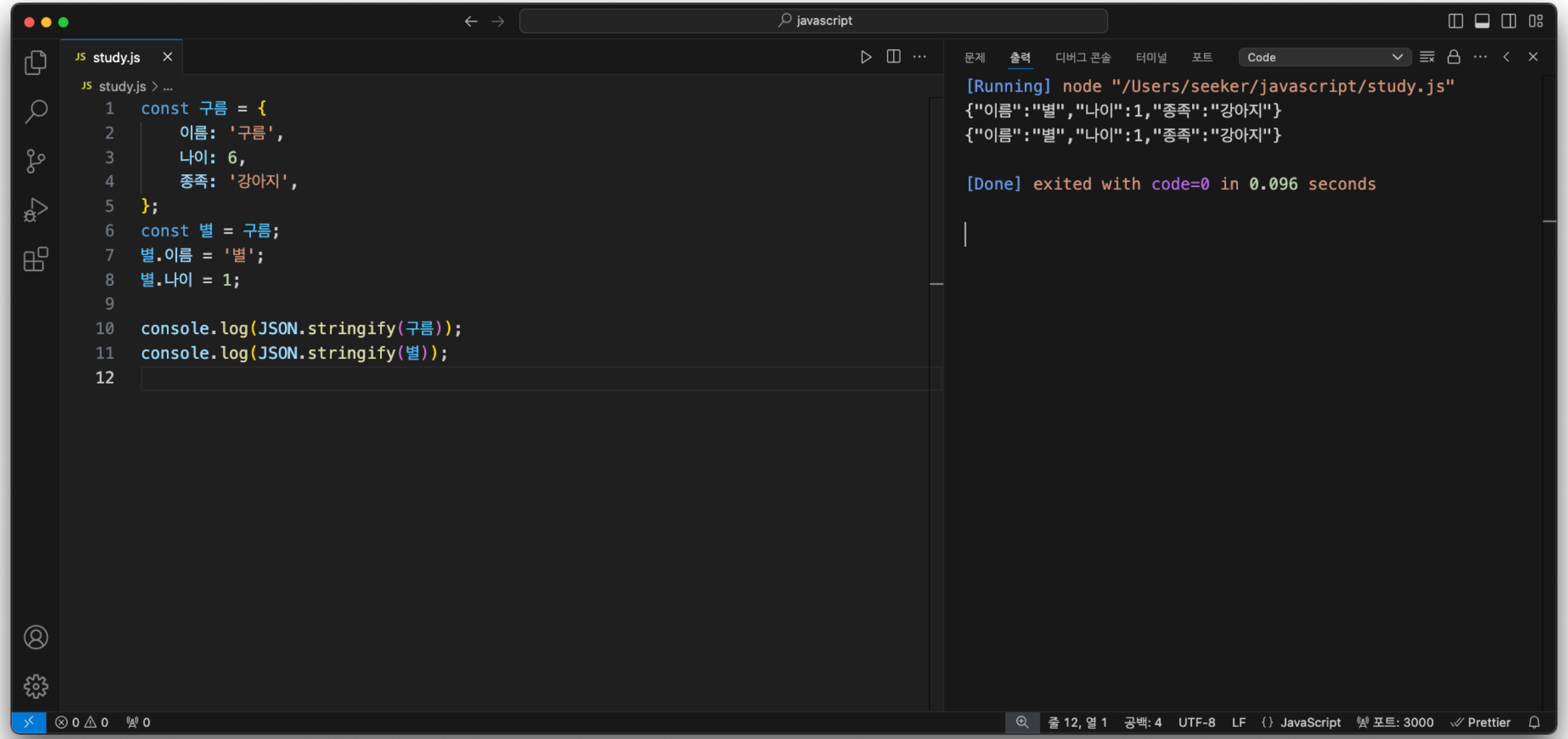
chap 6

배열 전개 연산자



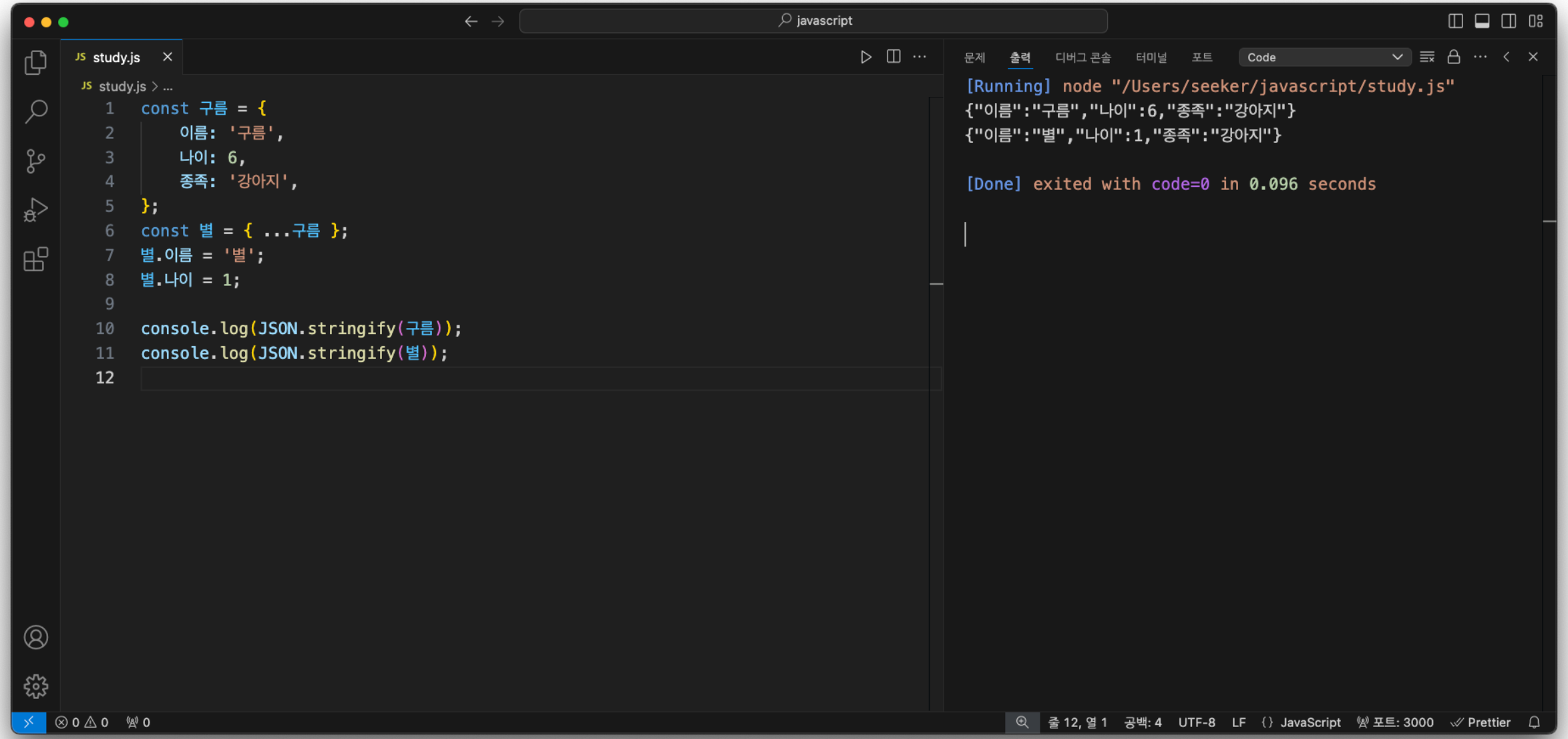
chap 6

객체 전개 연산자



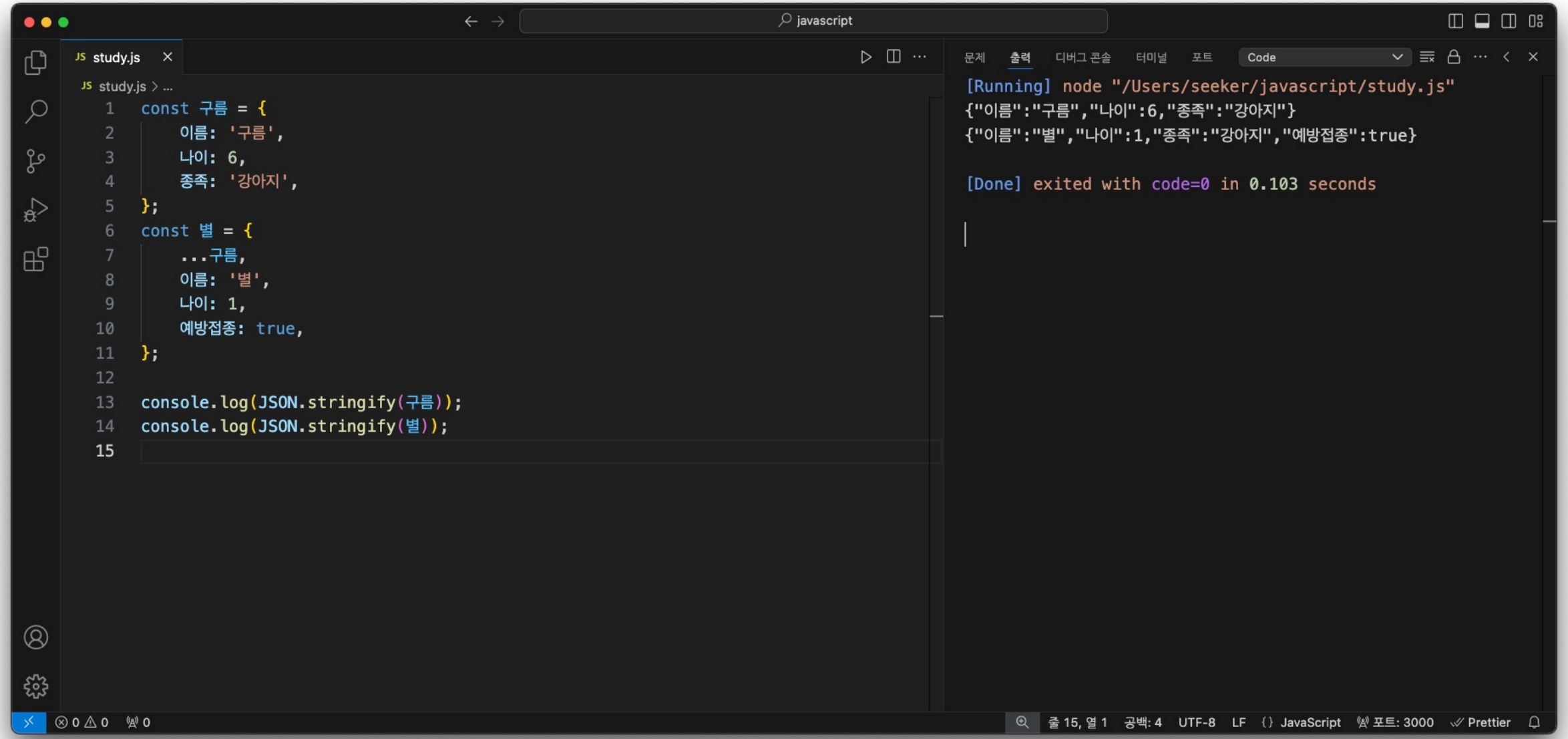
chap 6

객체 전개 연산자



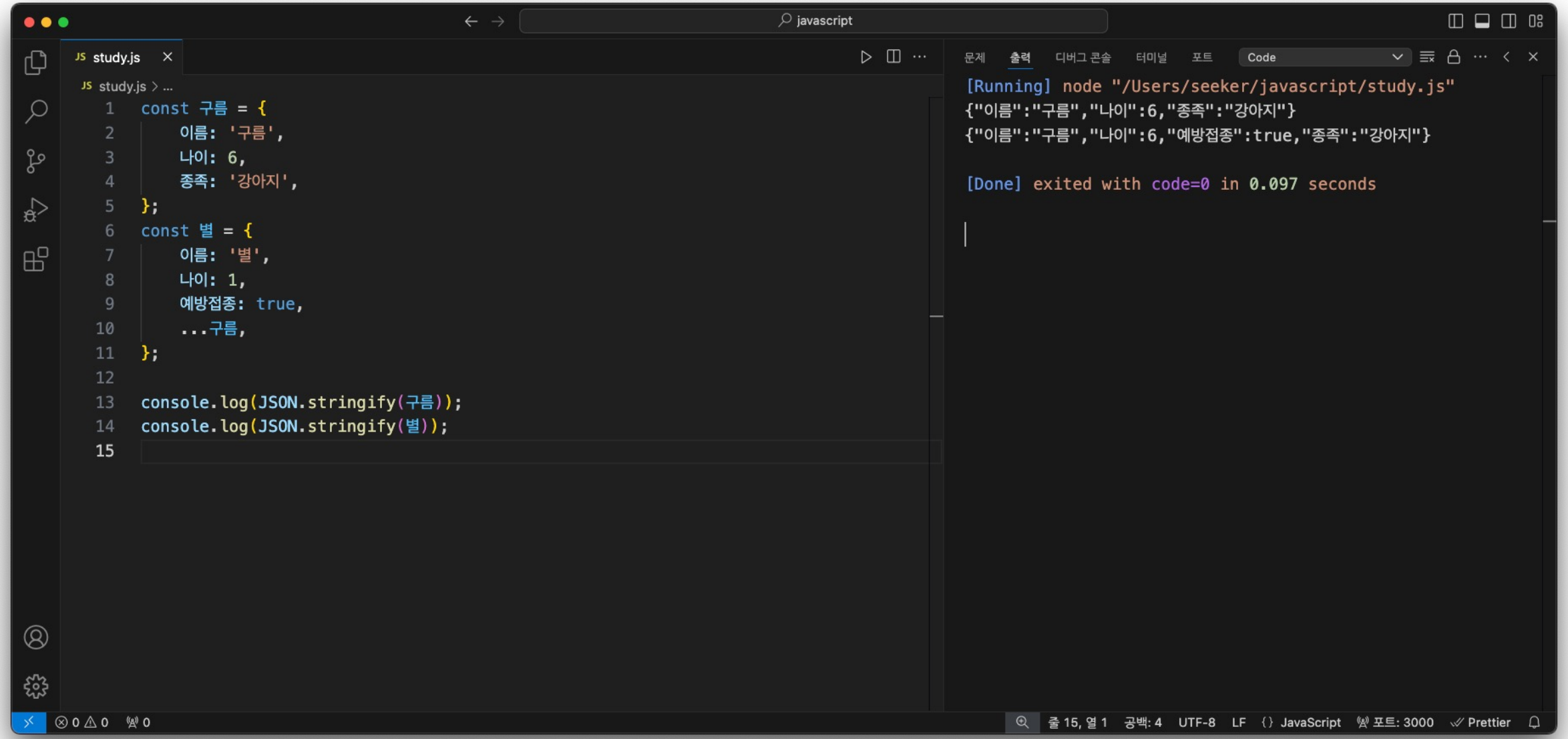
chap 6

객체 전개 연산자

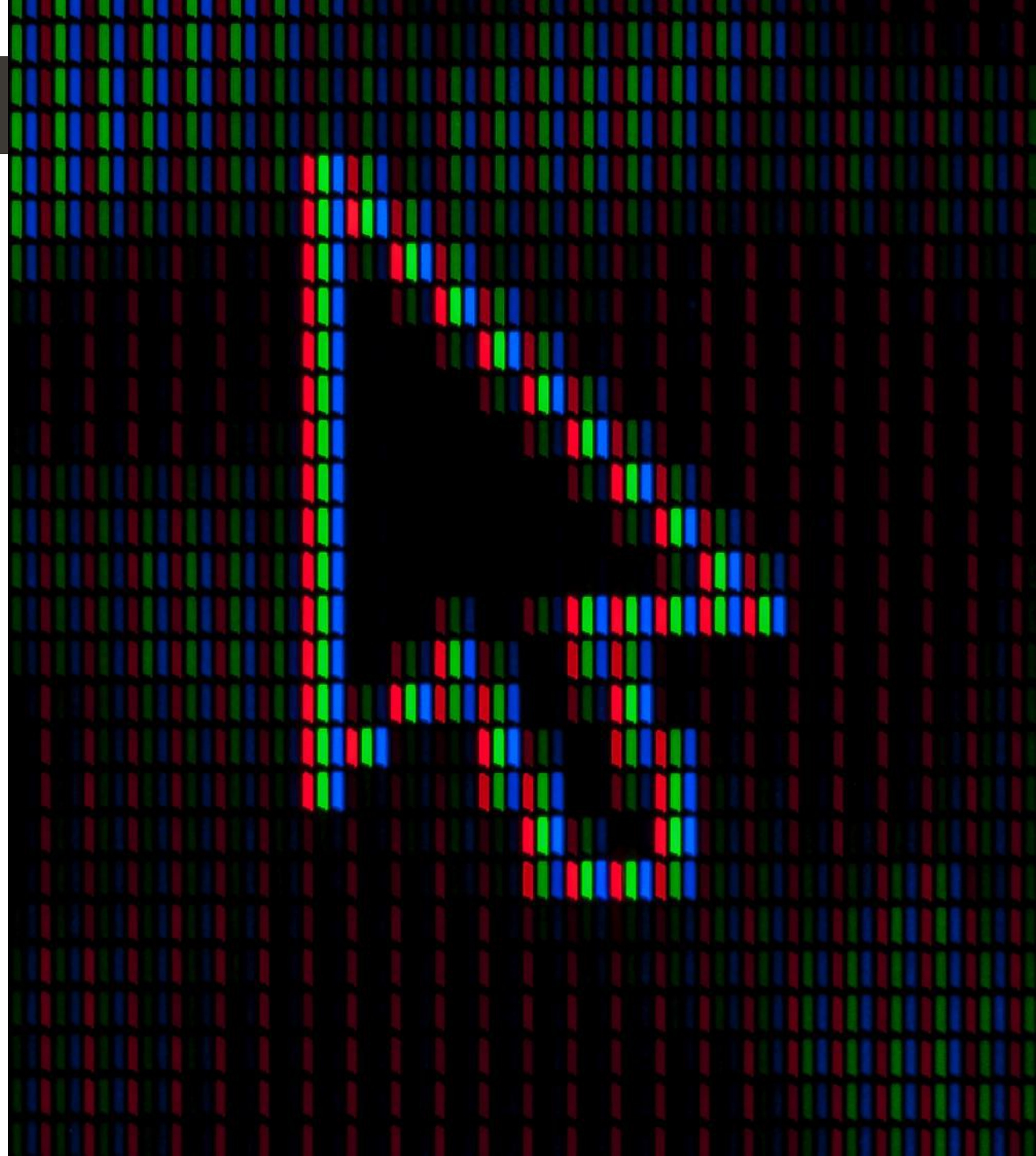


chap 6

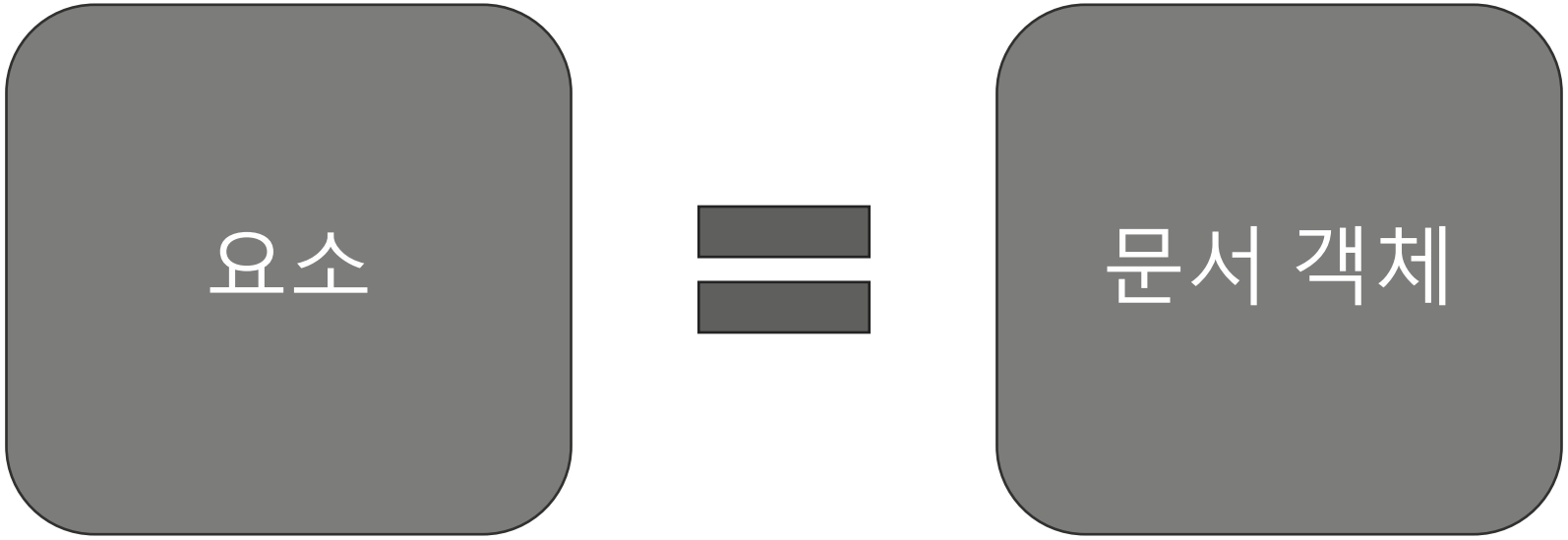
객체 전개 연산자



목차 a table of contents



DOM조작

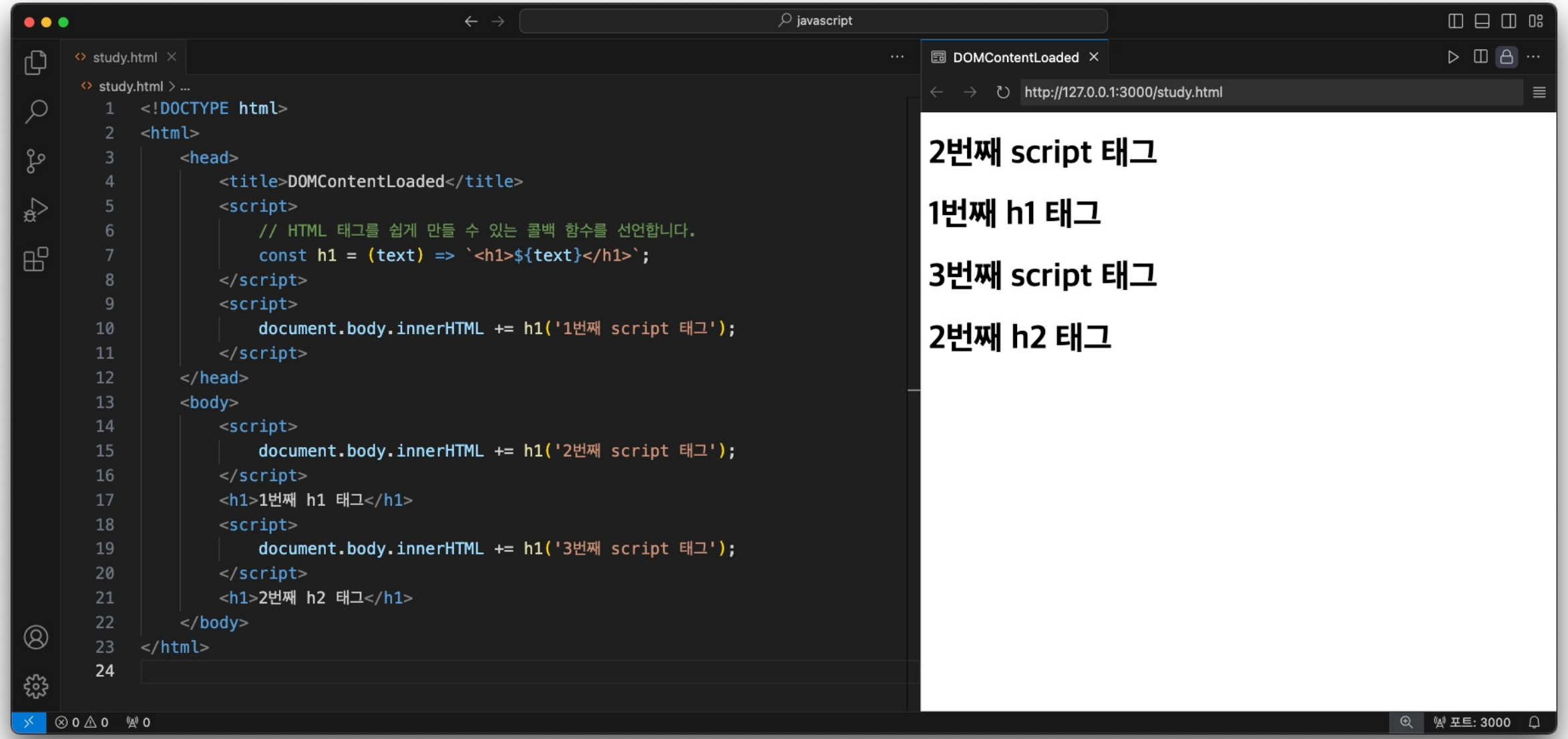


HTML 세계

자바스크립트 세계

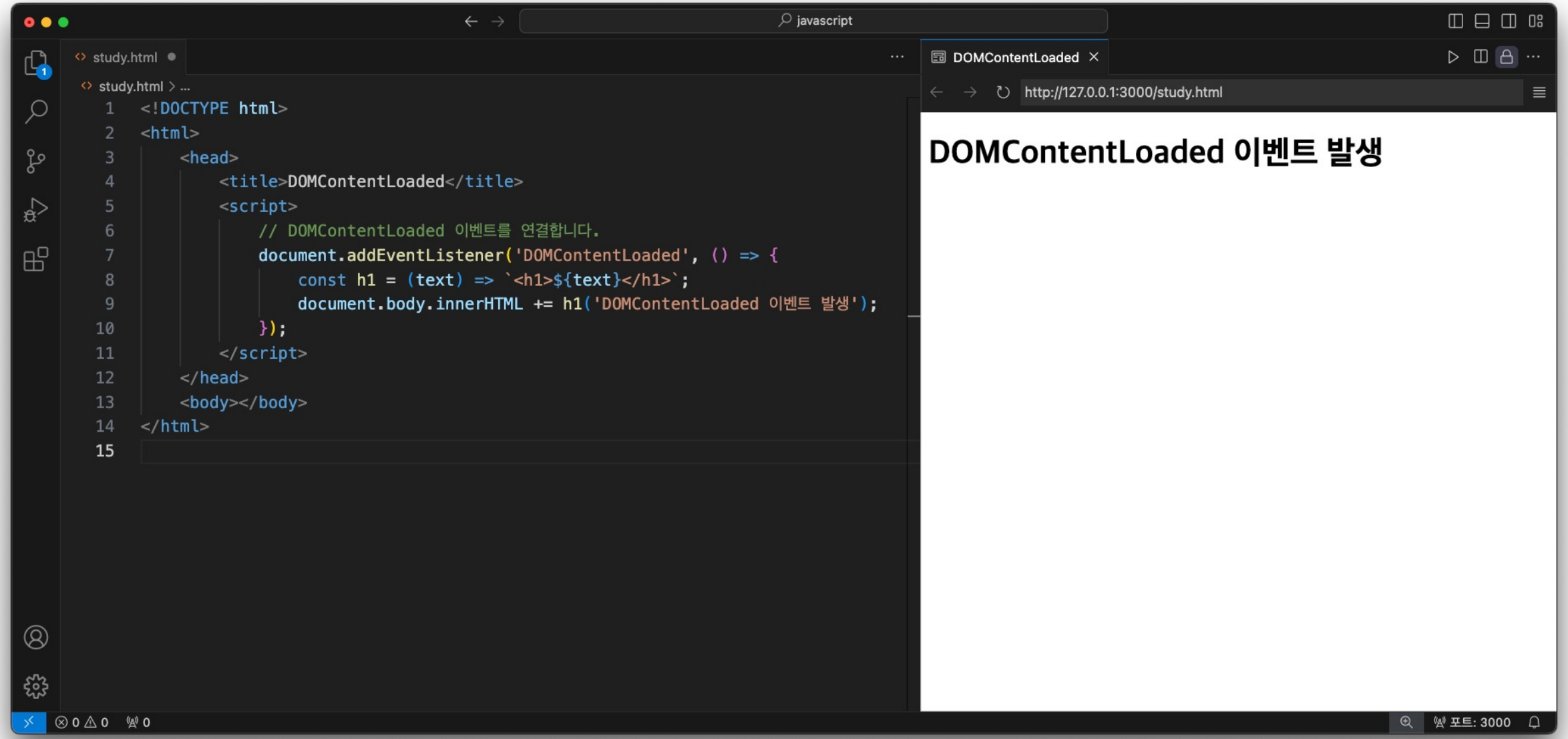
chap 7

DOMContentLoaded 이벤트



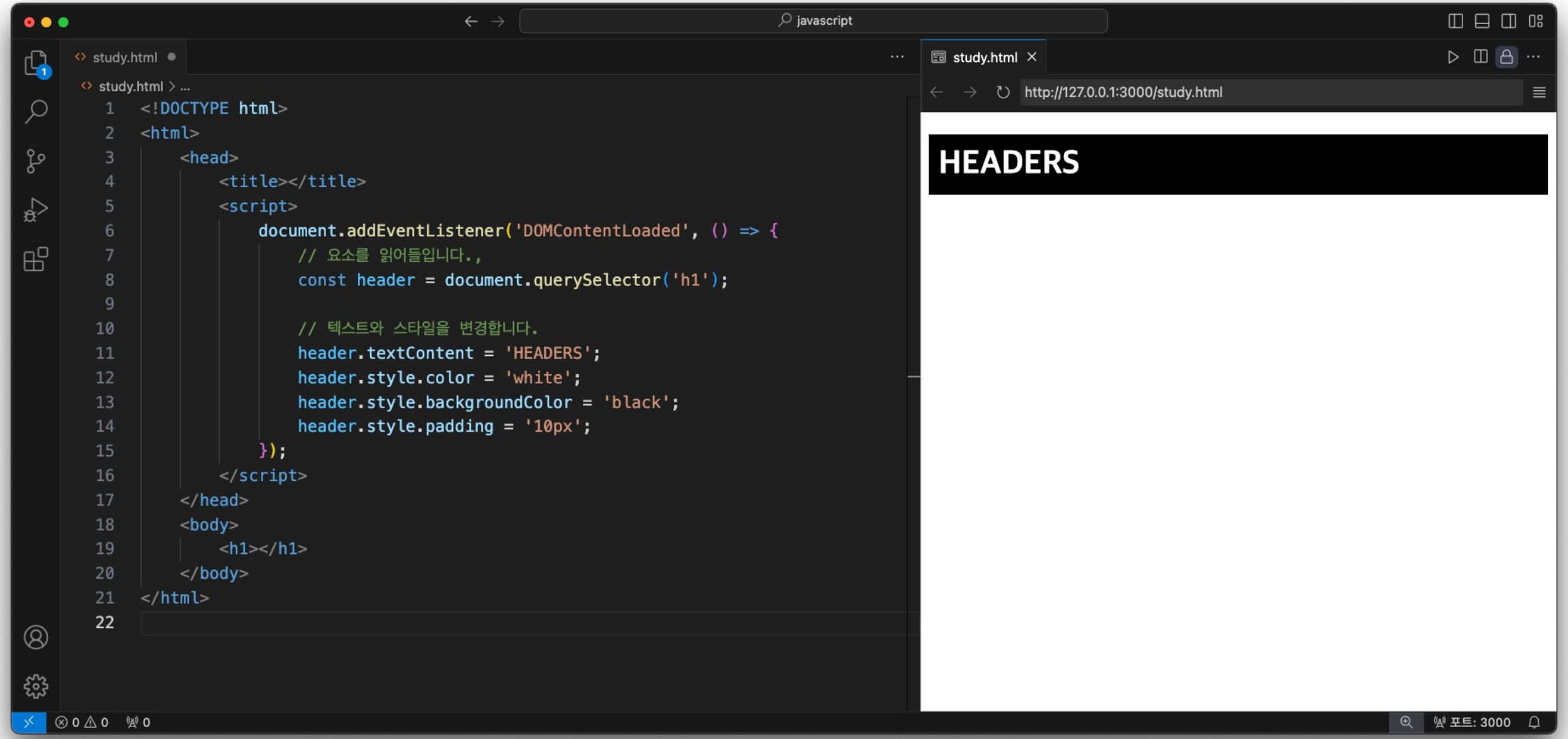
chap 7

DOMContentLoaded 이벤트



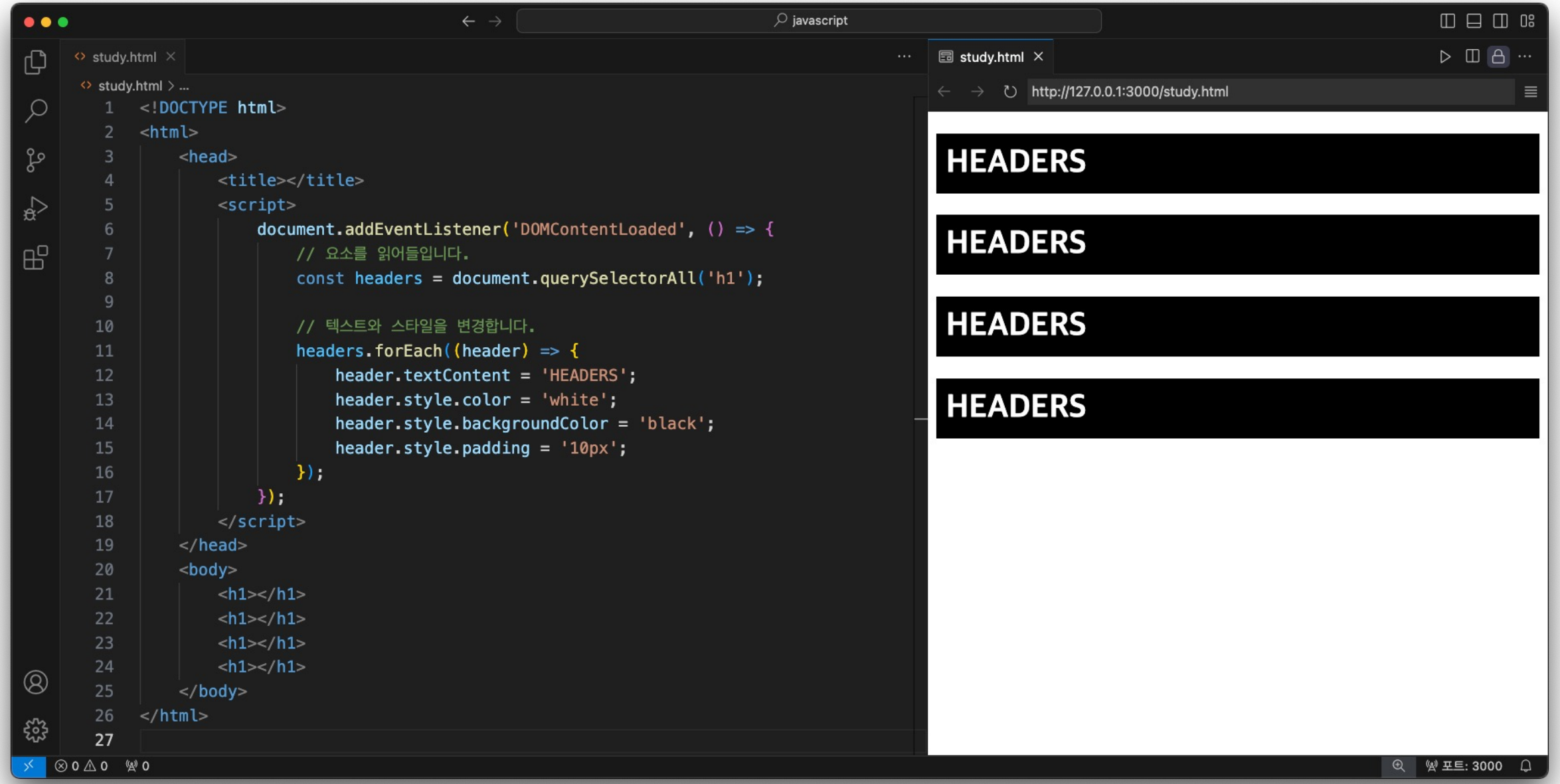
chap 7

문서 객체 가져오기

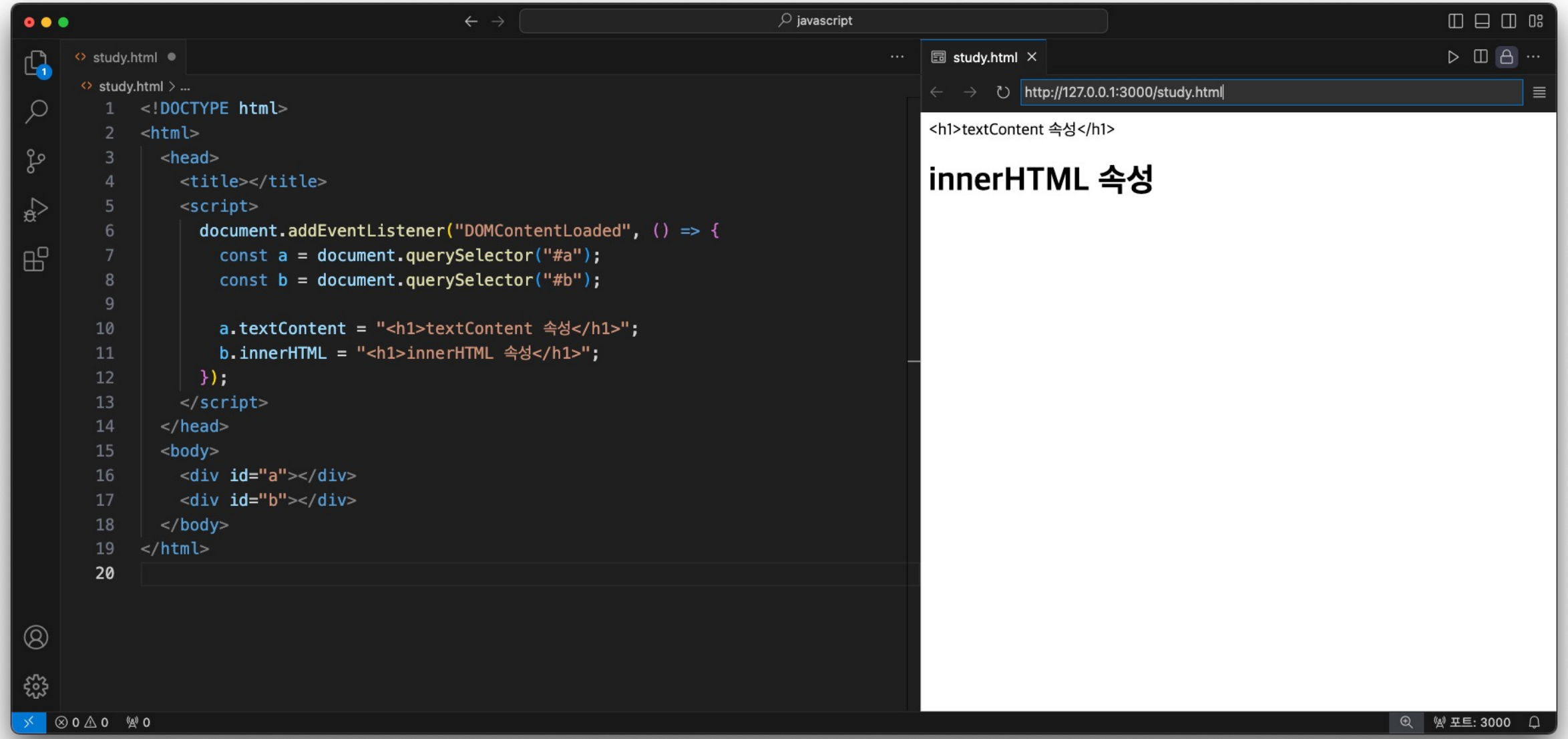


chap 7

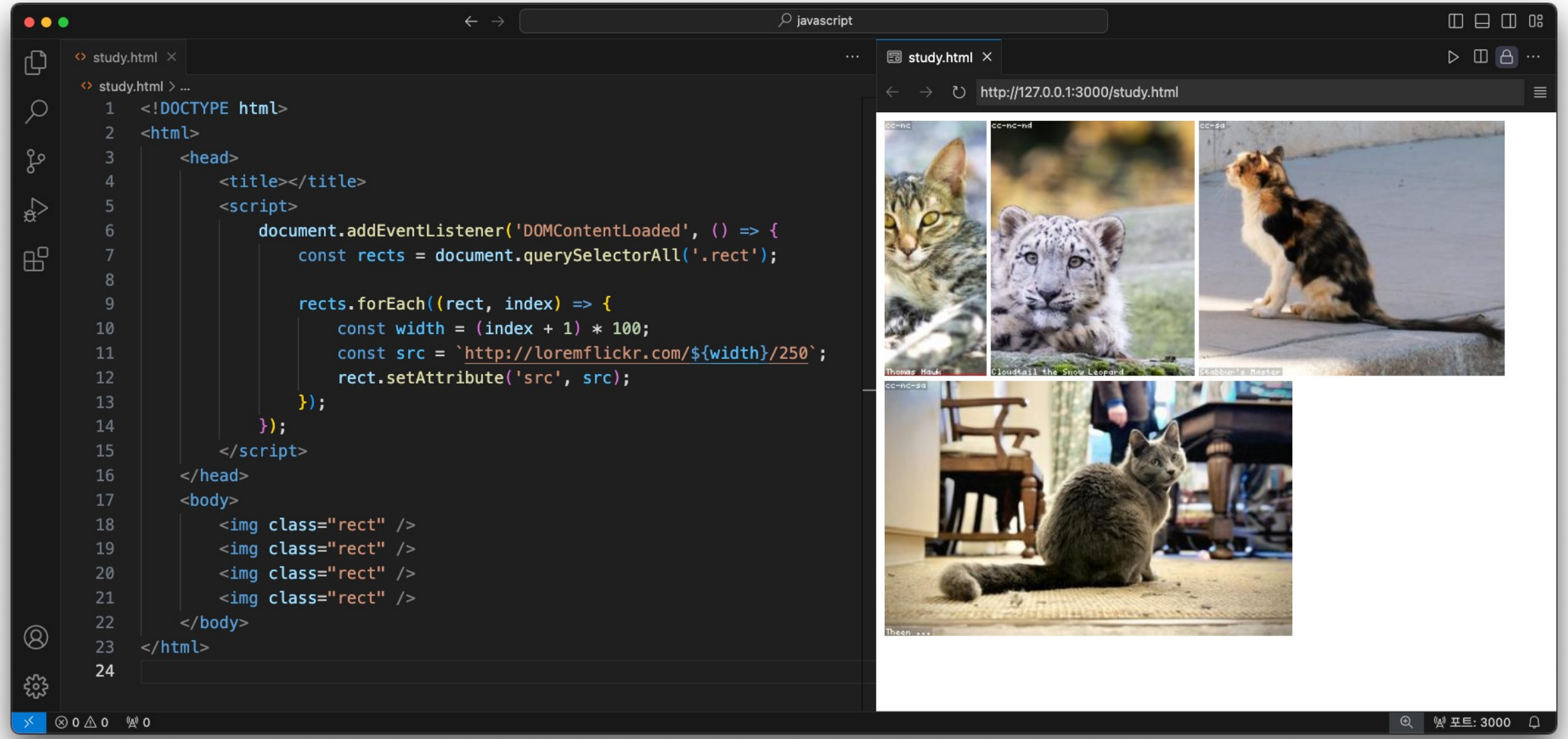
문서 객체 가져오기



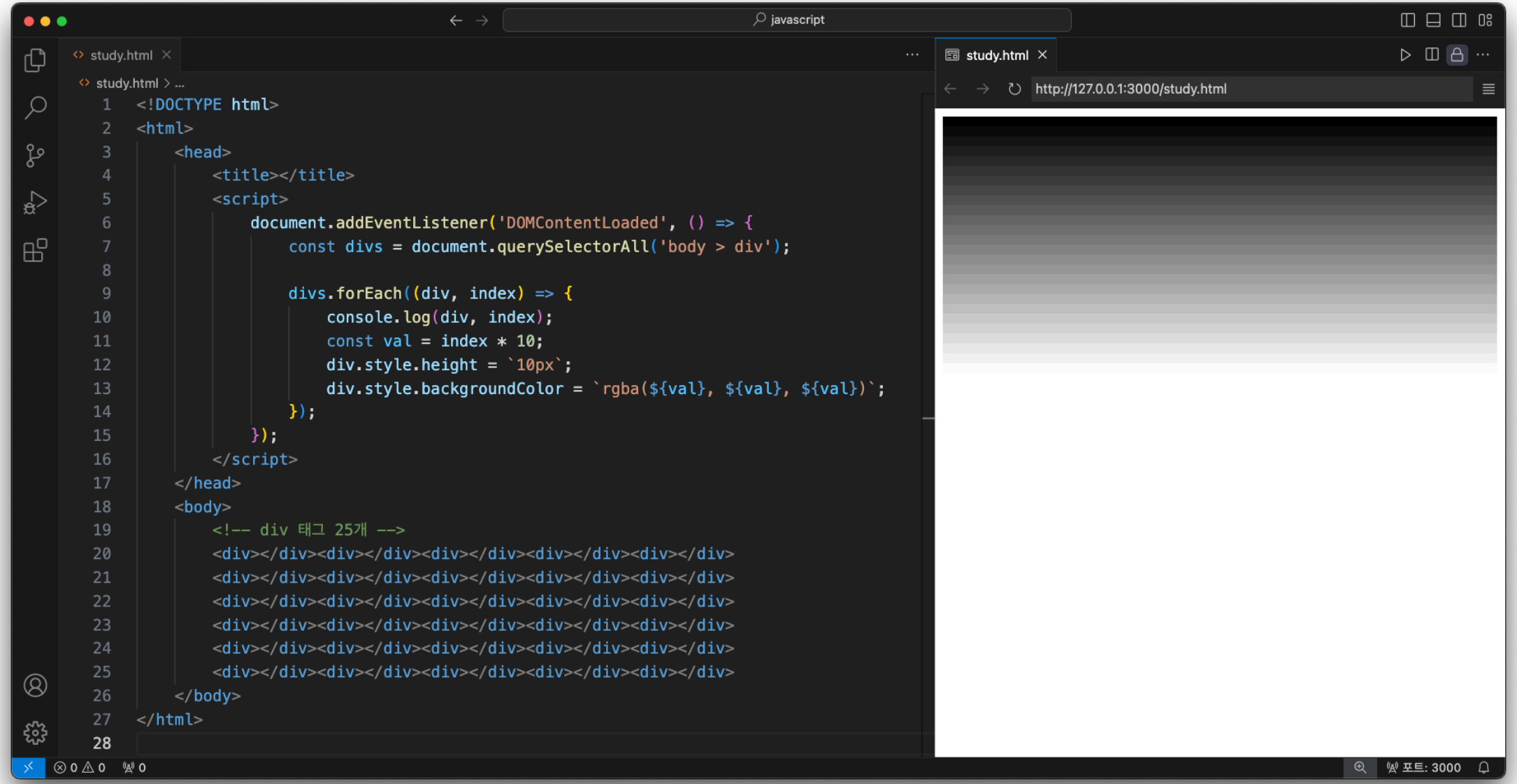
chap 7 글자 조작하기



chap 7 속성 조작하기

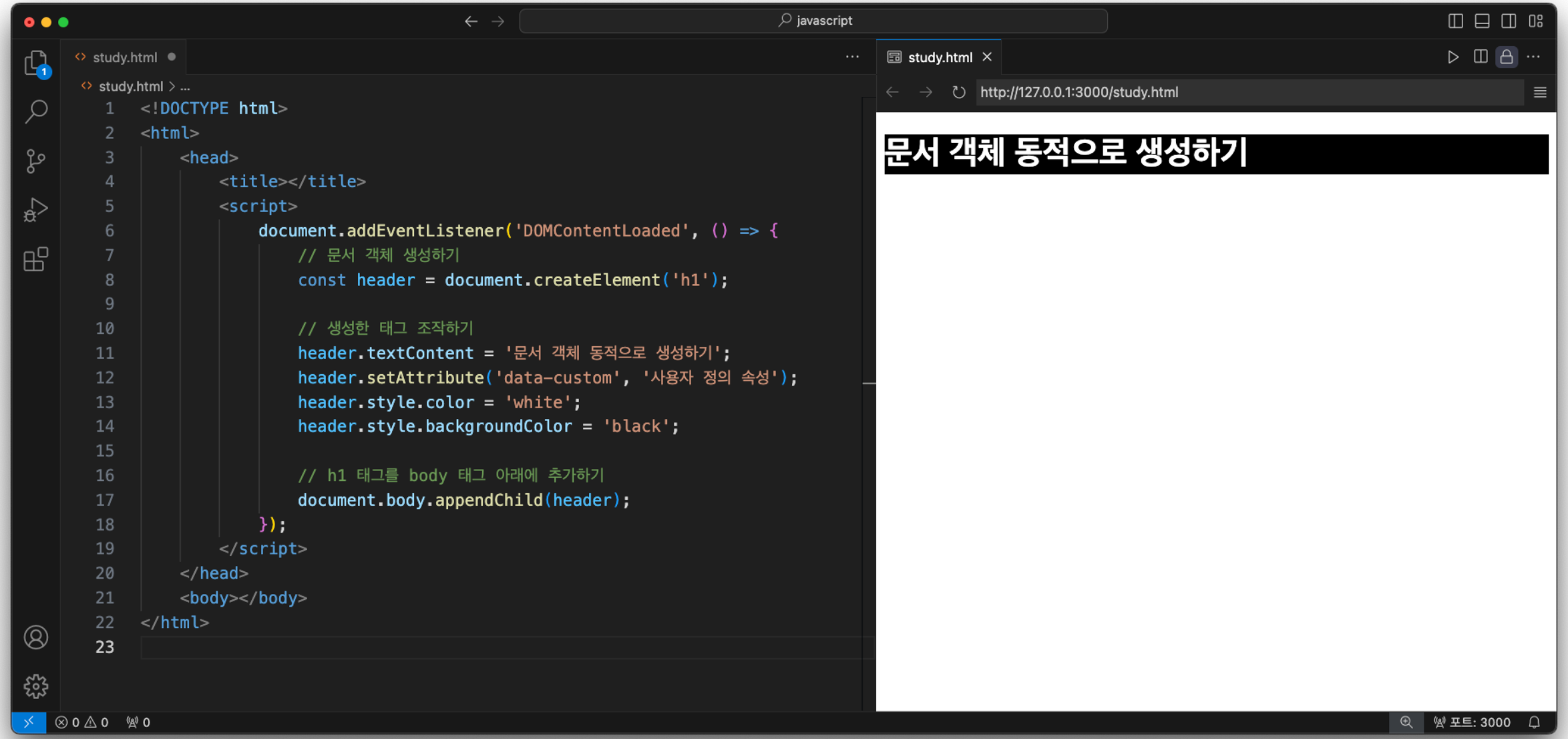


chap 7 스타일 조작하기



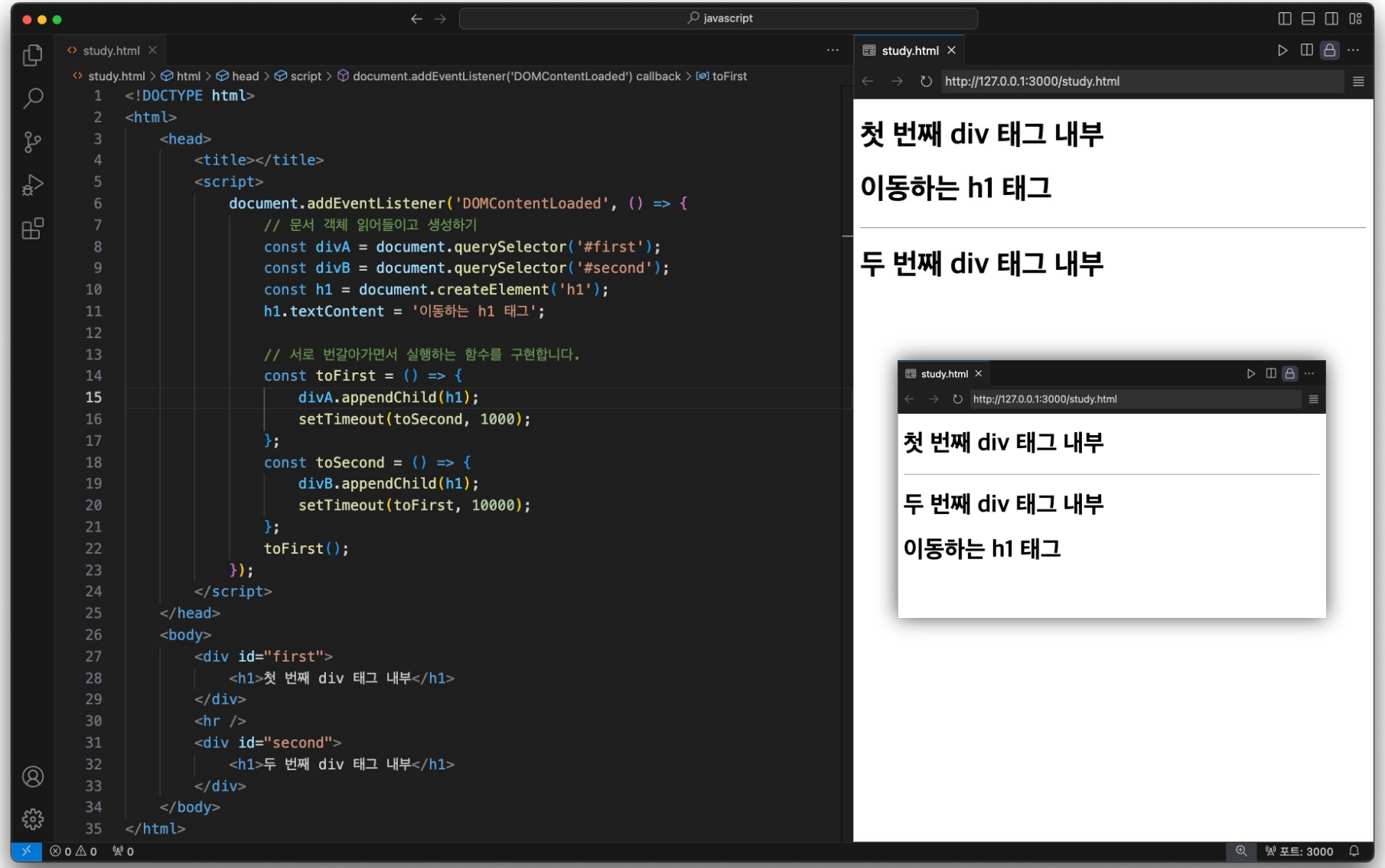
chap 7

문서 객체 생성



chap 7

문서 객체 이동



chap 7

문서 객체 제거하기

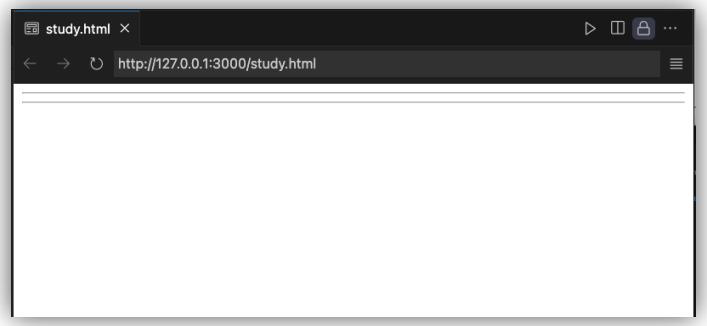
```html
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <script>
            document.addEventListener('DOMContentLoaded', () => {
                setTimeout(() => {
                    const h1 = document.querySelector('h1');

                    h1.parentNode.removeChild(h1);
                    // document.body.removeChild(h1)
                }, 3000);
            });
        </script>
    </head>
    <body>
        <hr />
        <h1>제거 대상 문서 객체</h1>
        <hr />
    </body>
</html>
```

study.html

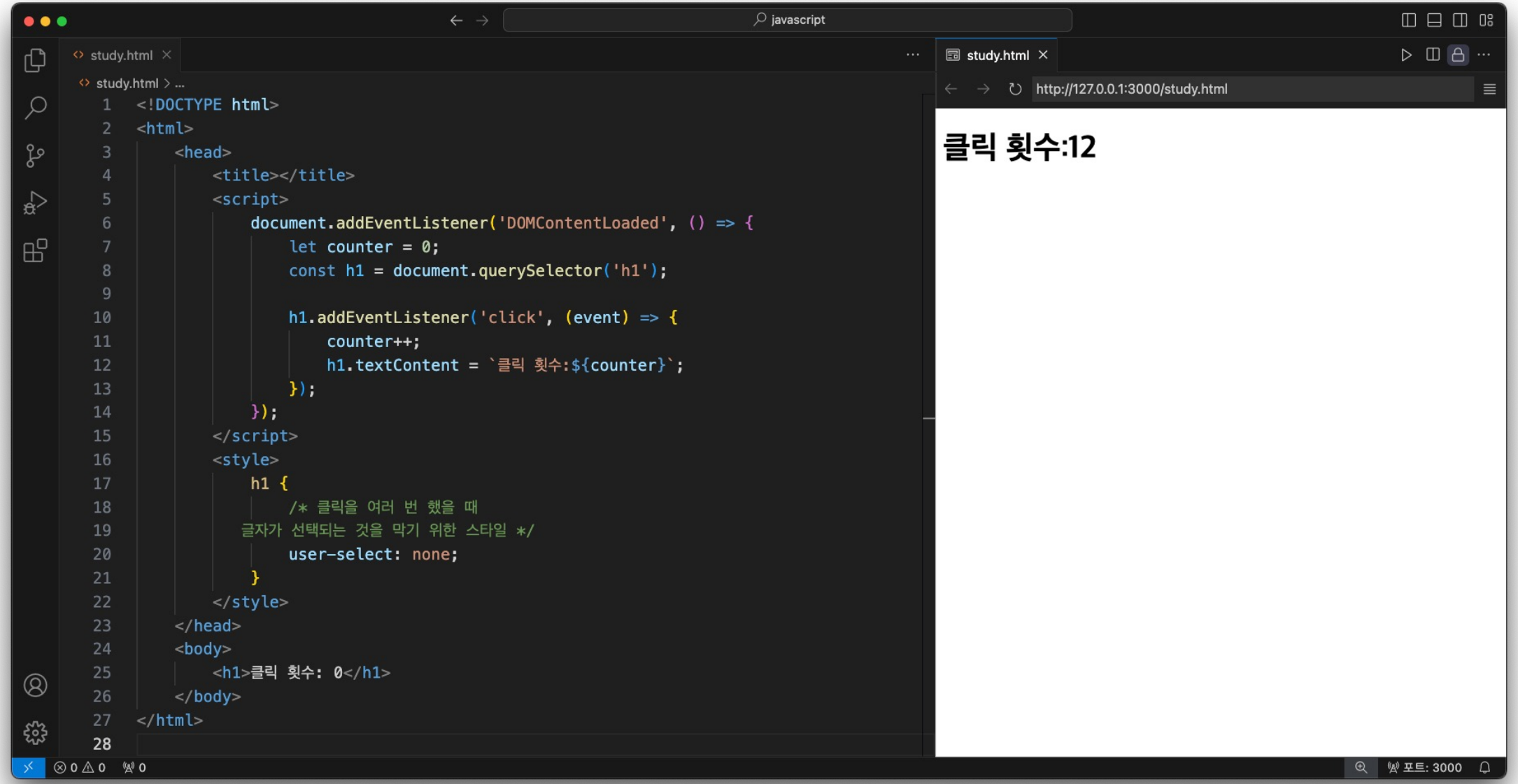
http://127.0.0.1:3000/study.html

제거 대상 문서 객체



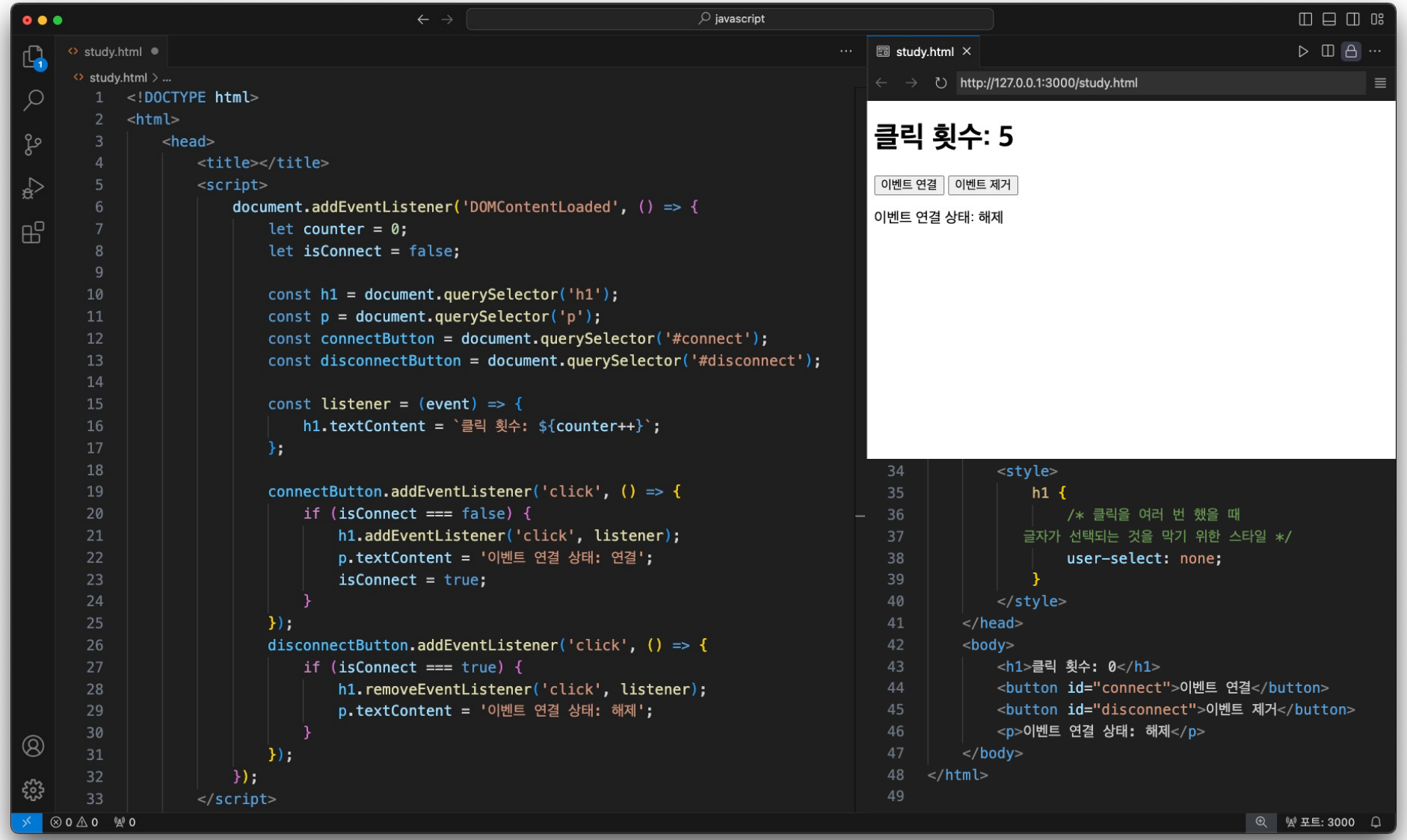
chap 7

이벤트 설정하기



chap 7

이벤트 설정하기




이벤트활용

chap 7

이벤트 모델

study.html

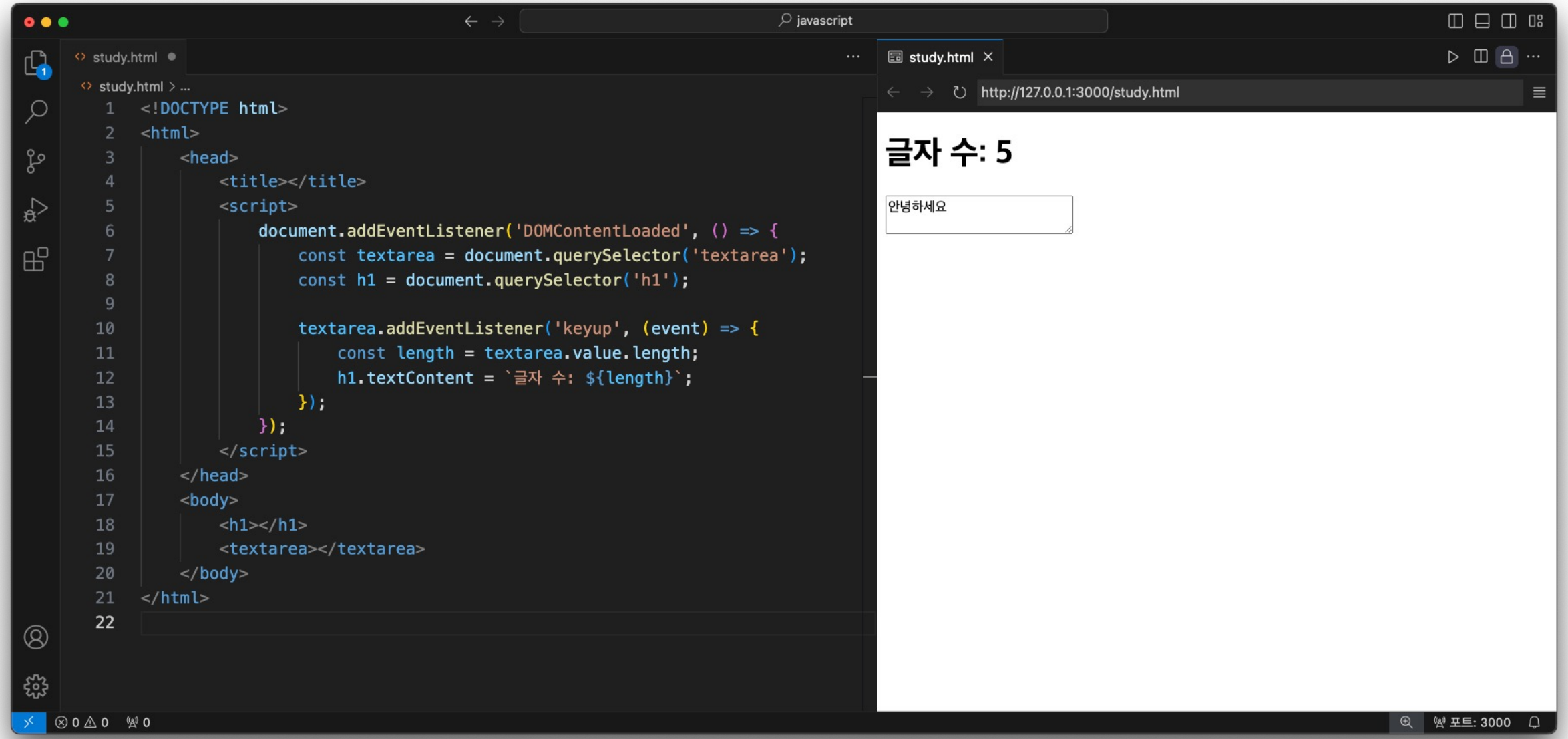
```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title></title>
5          <script>
6              document.body.addEventListener('keyup', () => {
7
8              })
9
10             document.body.onkeyup = (event) => {
11
12             }
13         </script>
14     </head>
15     <body>
16     </body>
17 </html>
18
```

study.html

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title></title>
5          <script>
6              const listener = (event) => {
7
8              }
9          </script>
10     </head>
11     <body onkeyup="listener(event)">
12     </body>
13 </html>
14
```

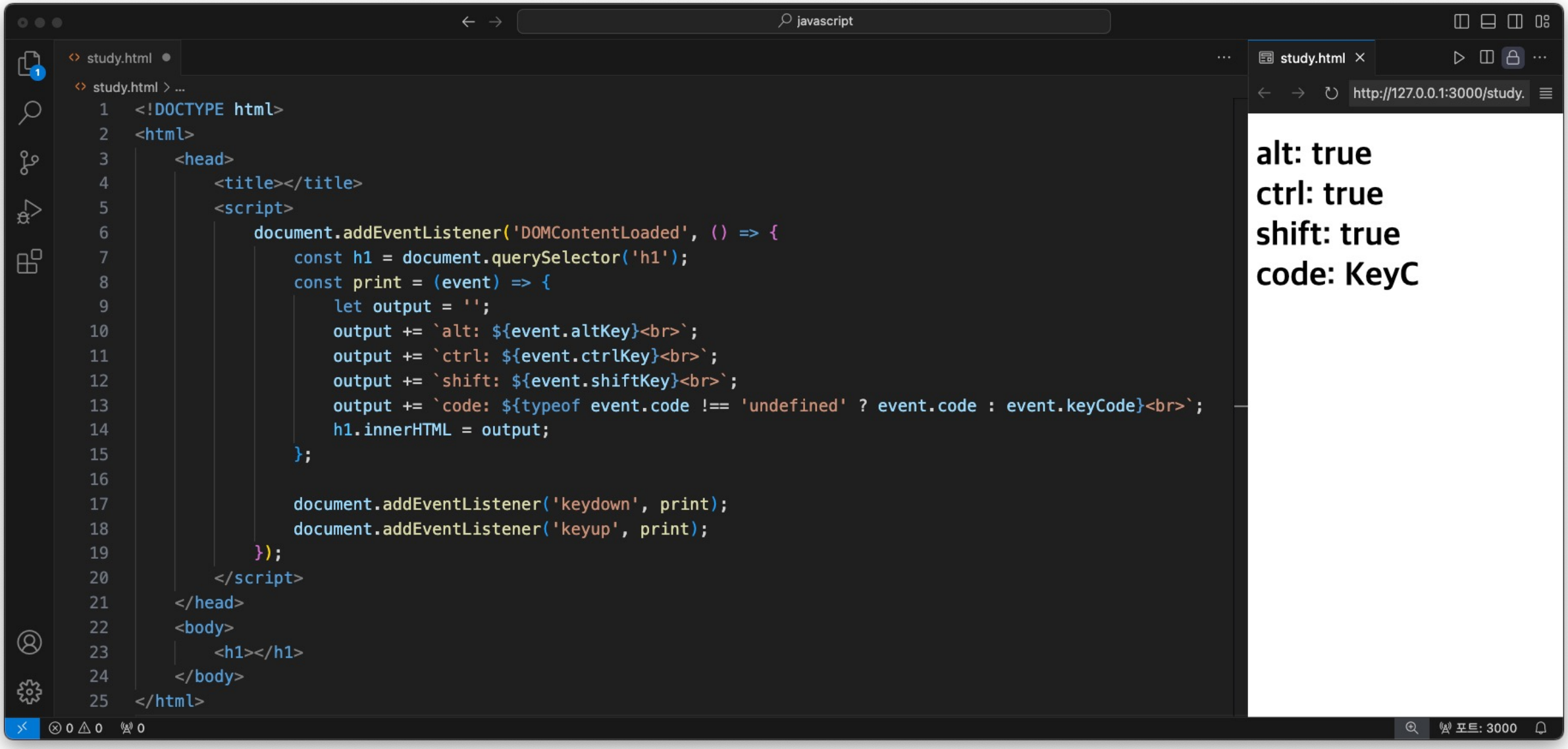
chap 7

키보드 이벤트



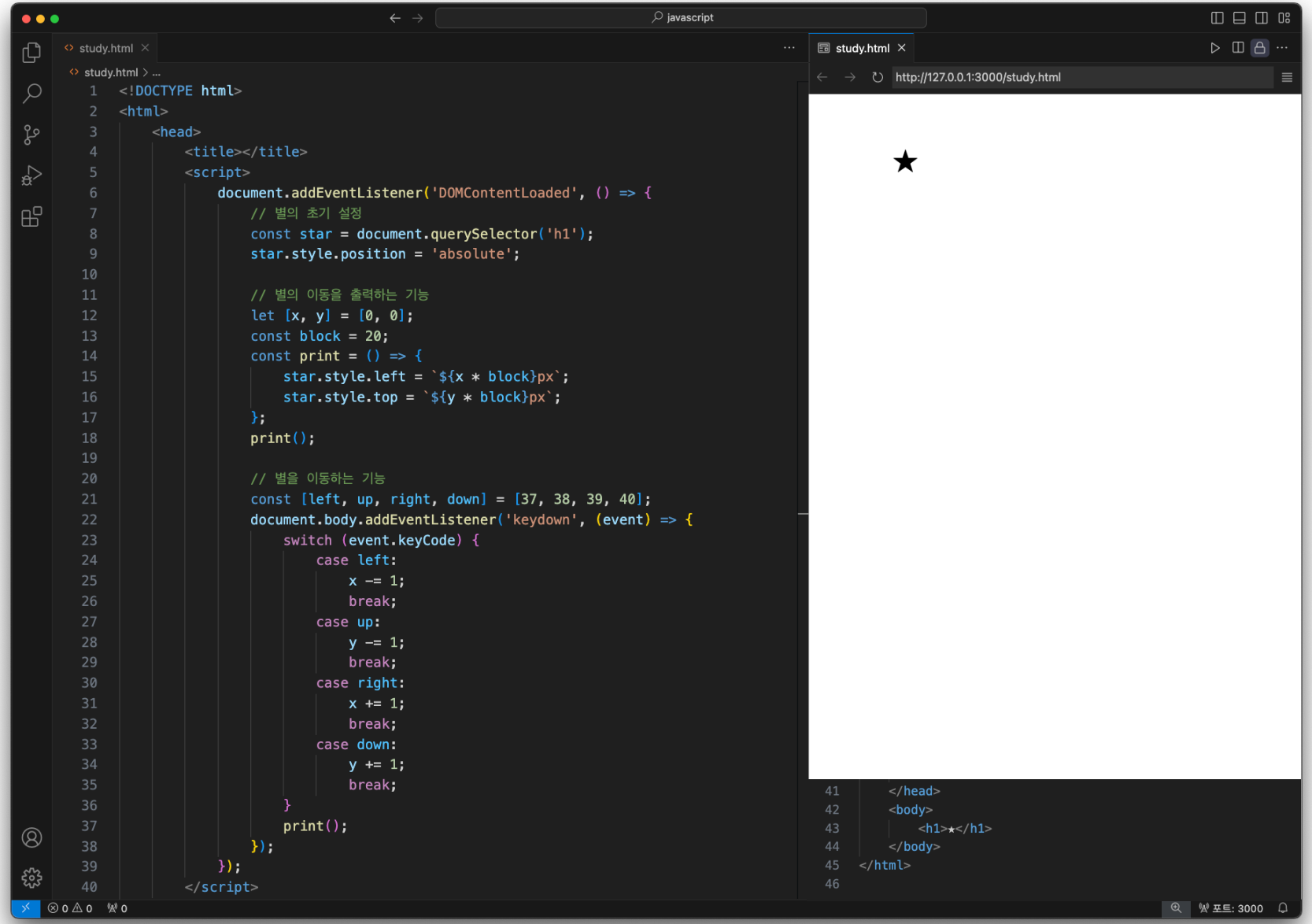
chap 7

키보드 이벤트 > 키보드 키 코드 사용하기



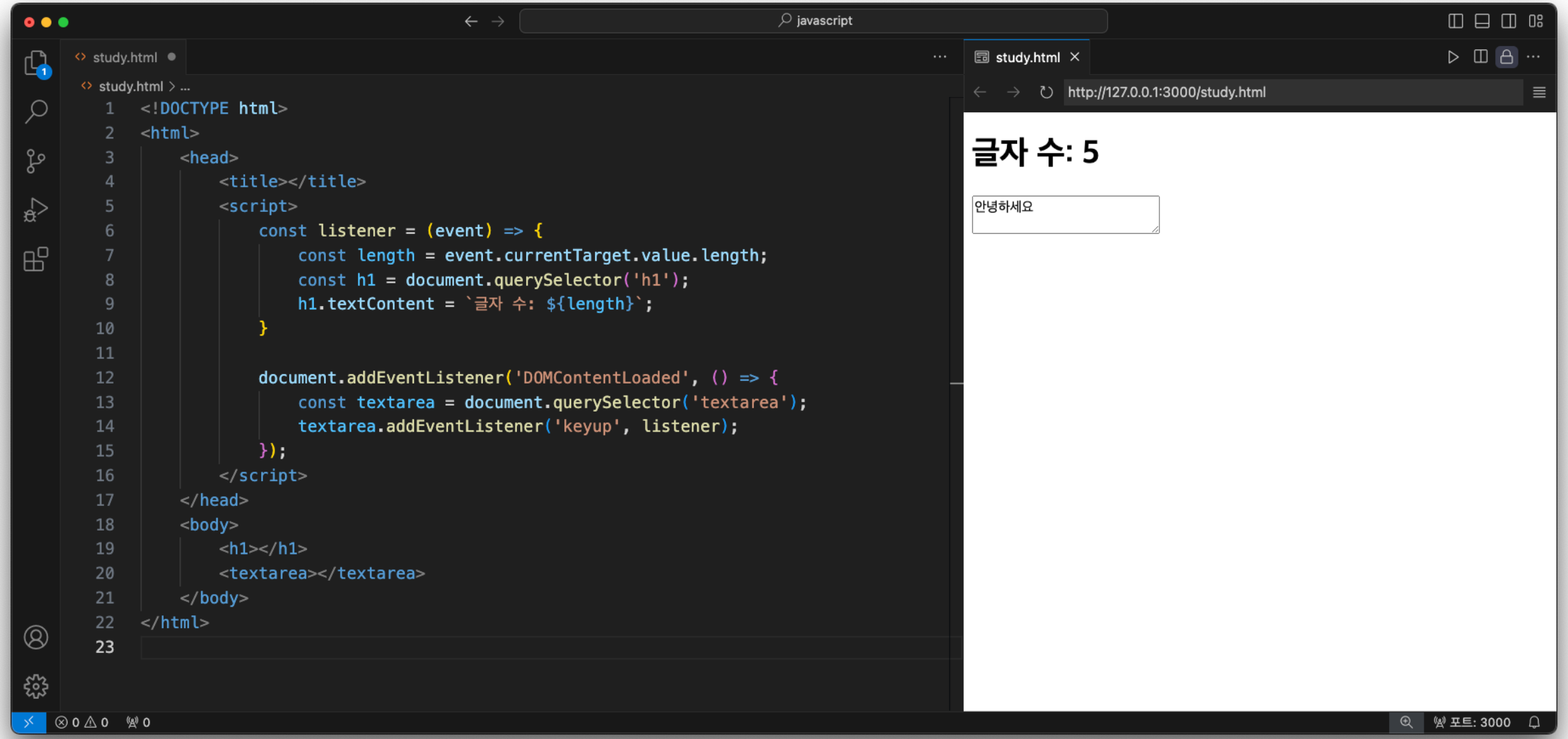
chap 7

키보드 이벤트 > 키보드 키 코드 사용하기



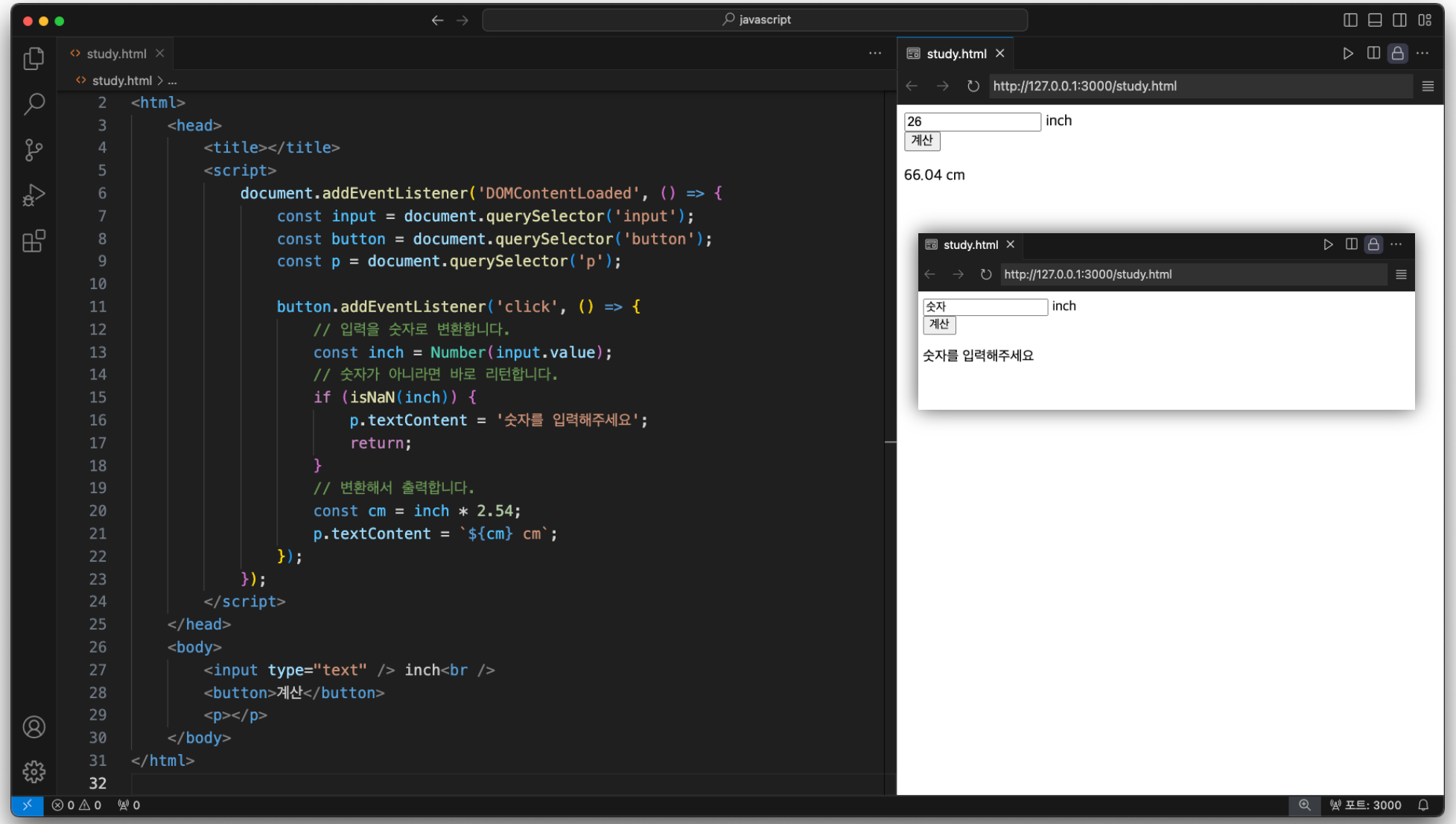
chap 7

이벤트 발생 객체



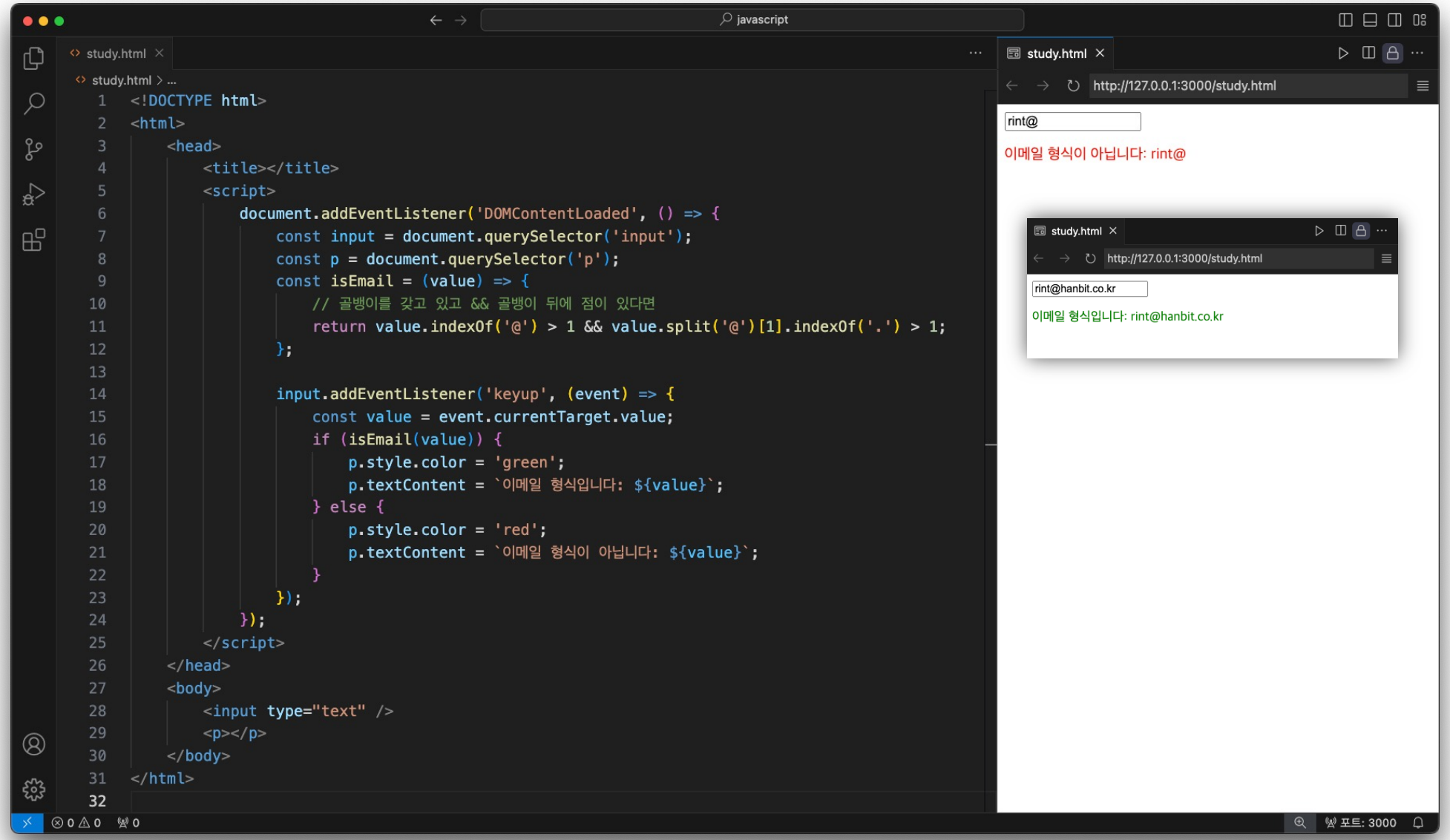
chap 7

글자 입력 양식 이벤트



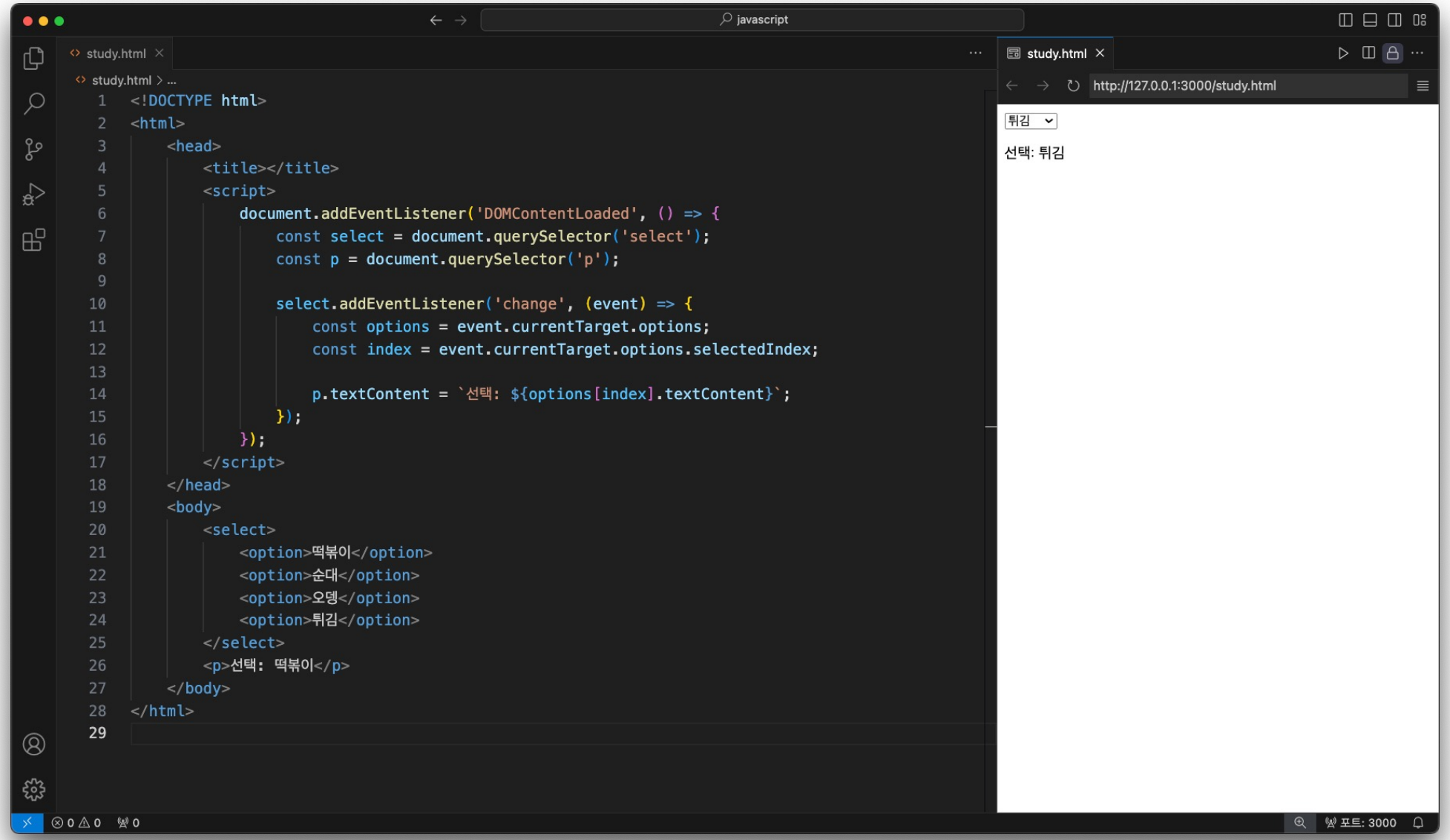
chap 7

글자 입력 양식 이벤트



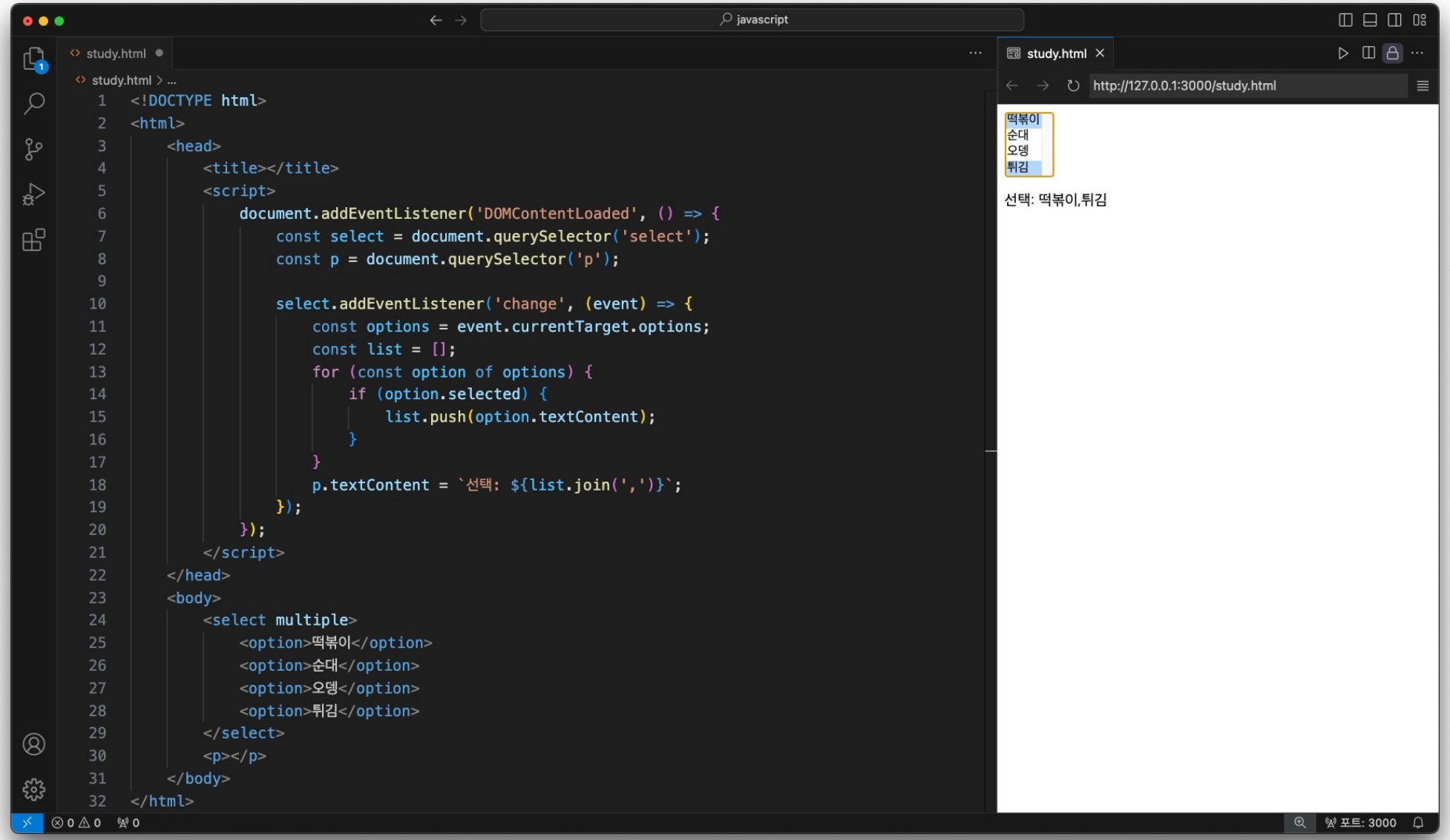
chap 7

글자 입력 양식 이벤트 > 드롭다운 목록 활용



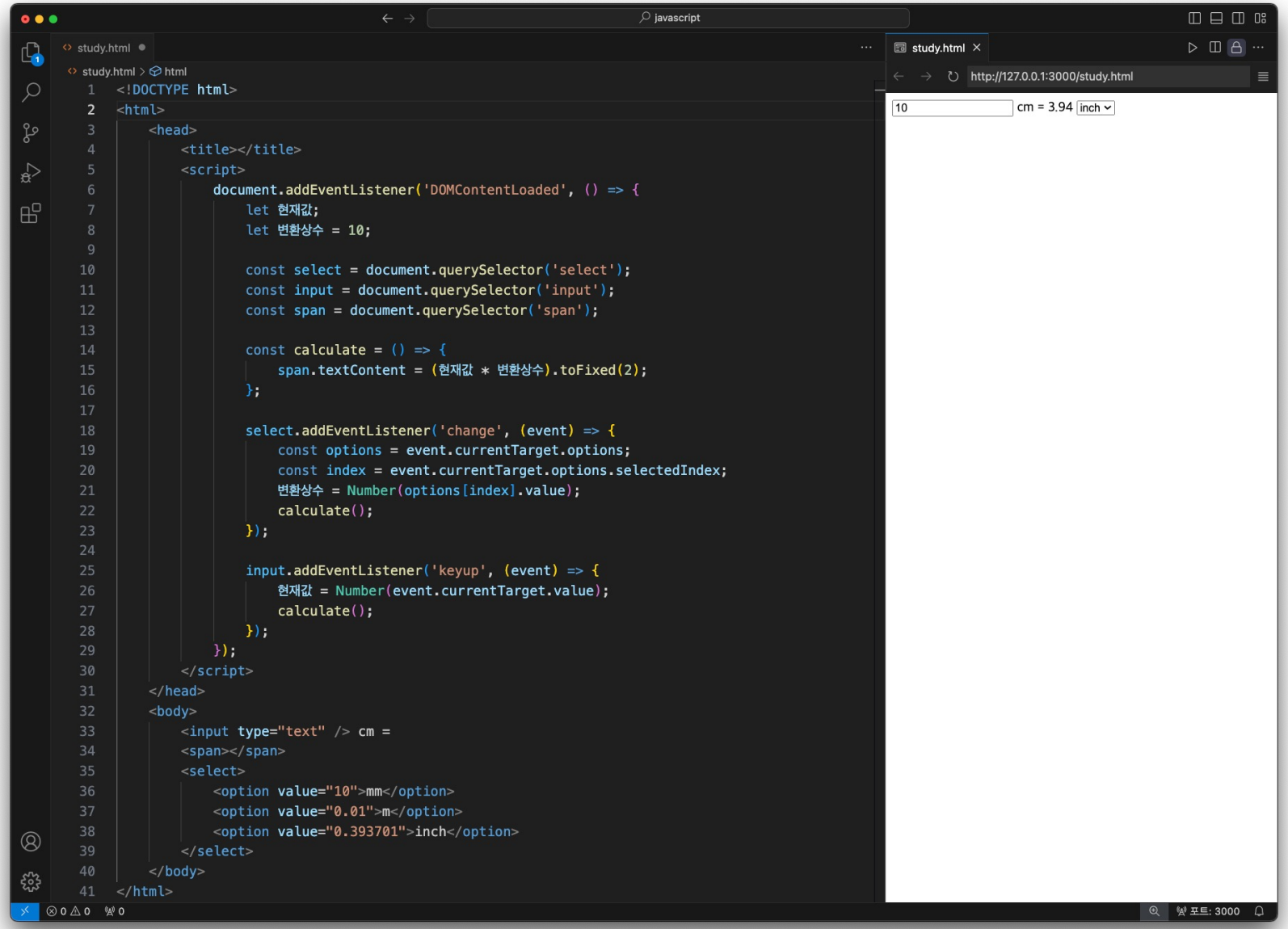
chap 7

글자 입력 양식 이벤트 > 드롭다운 목록 활용



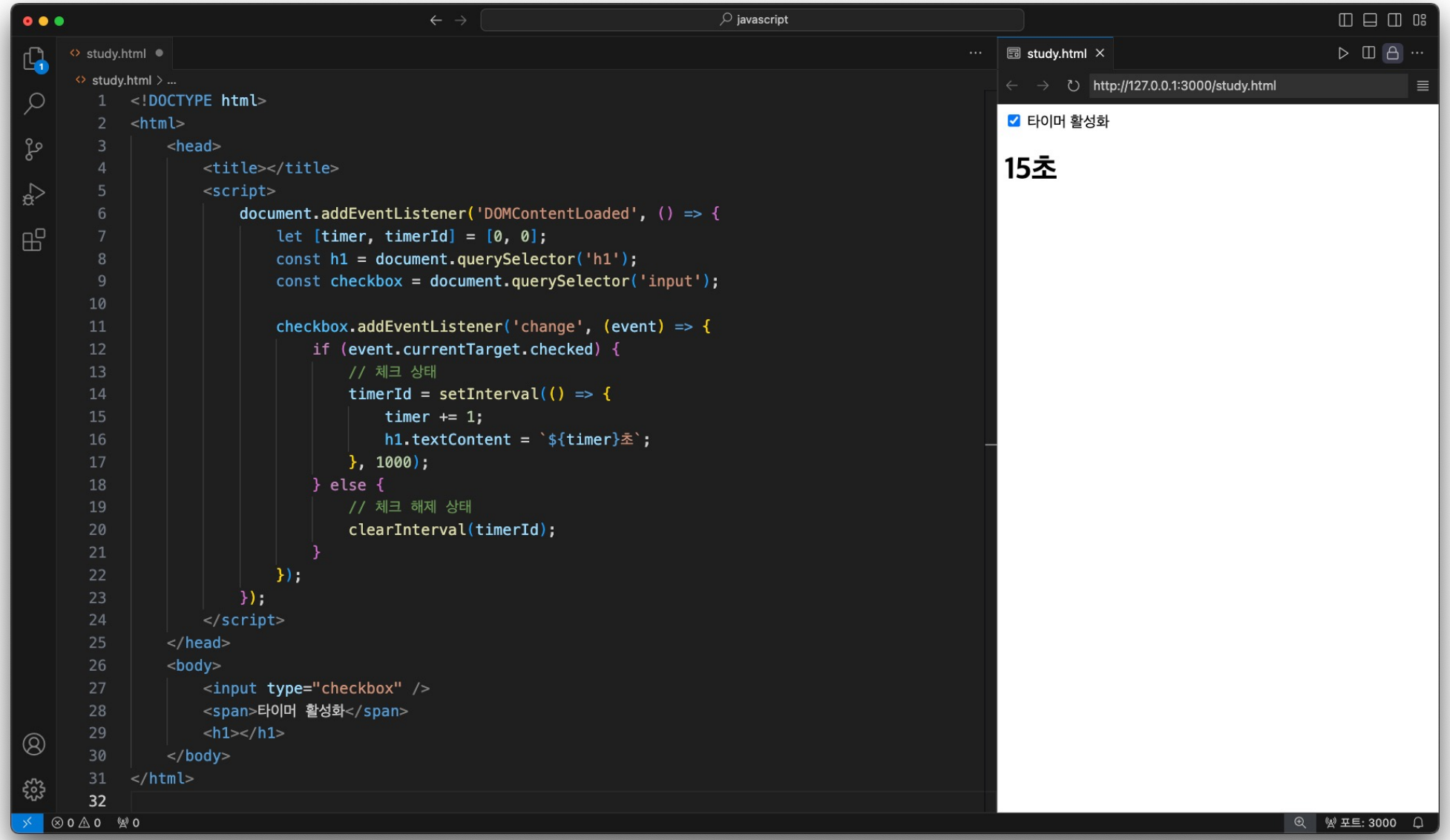
chap 7

글자 입력 양식 이벤트 > 드롭다운 목록 활용



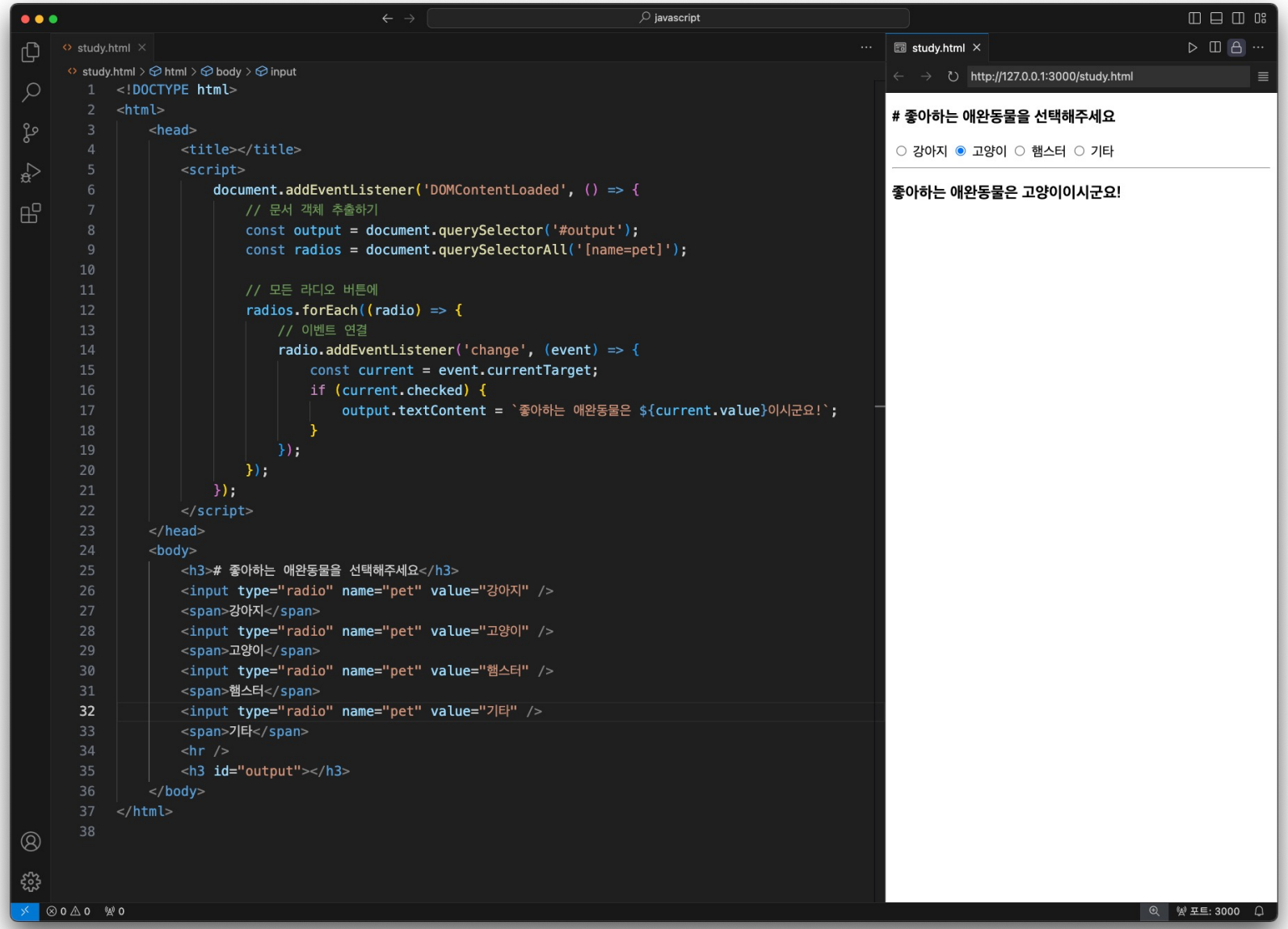
chap 7

글자 입력 양식 이벤트 > 체크 박스 활용



chap 7

글자 입력 양식 이벤트 > 라디오 버튼 활용



chap 7

기본 이벤트 막기

javascript

study.html

```
<!DOCTYPE html>
<html>
    <head>
        <title></title>
        <script>
            document.addEventListener('DOMContentLoaded', () => {
                const imgs = document.querySelectorAll('img');

                imgs.forEach((img) => {
                    img.addEventListener('contextmenu', (event) => {
                        event.preventDefault();
                    });
                });
            });
        </script>
    </head>
    <body>
        <img src="https://loremflickr.com/300/300" alt="" />
    </body>
</html>
```

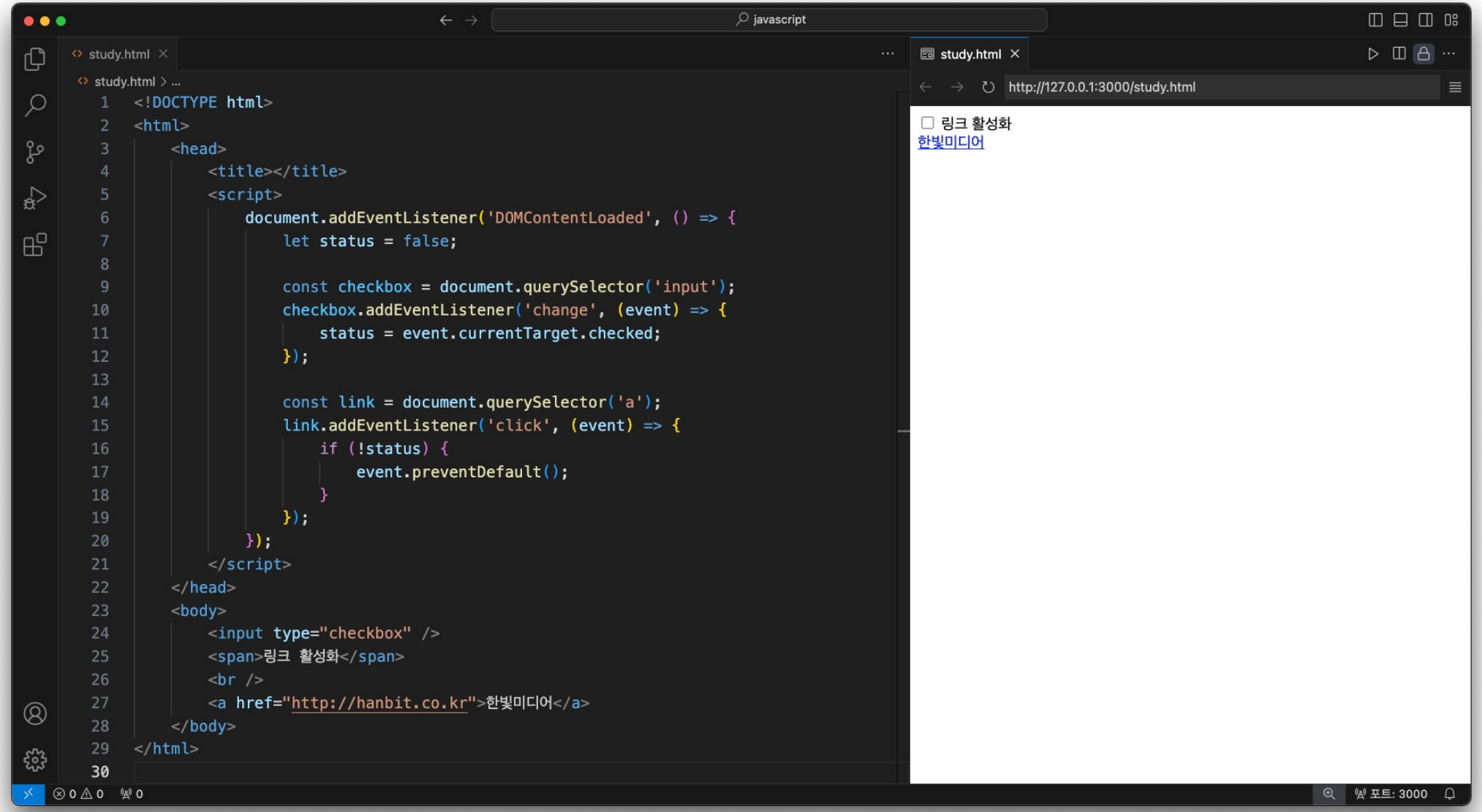
study.html

http://127.0.0.1:3000/study.html

포트: 3000

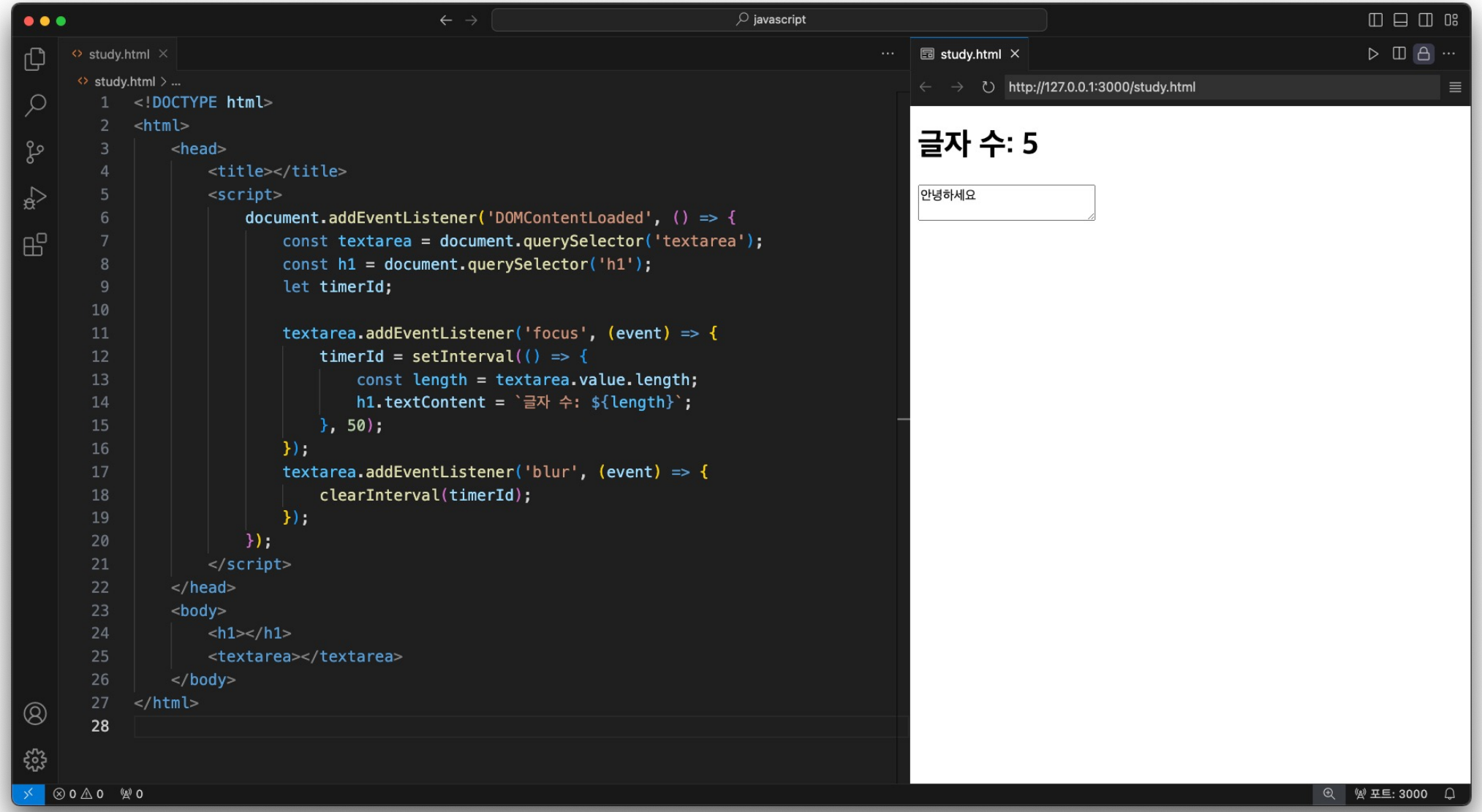
chap 7

기본 이벤트 막기

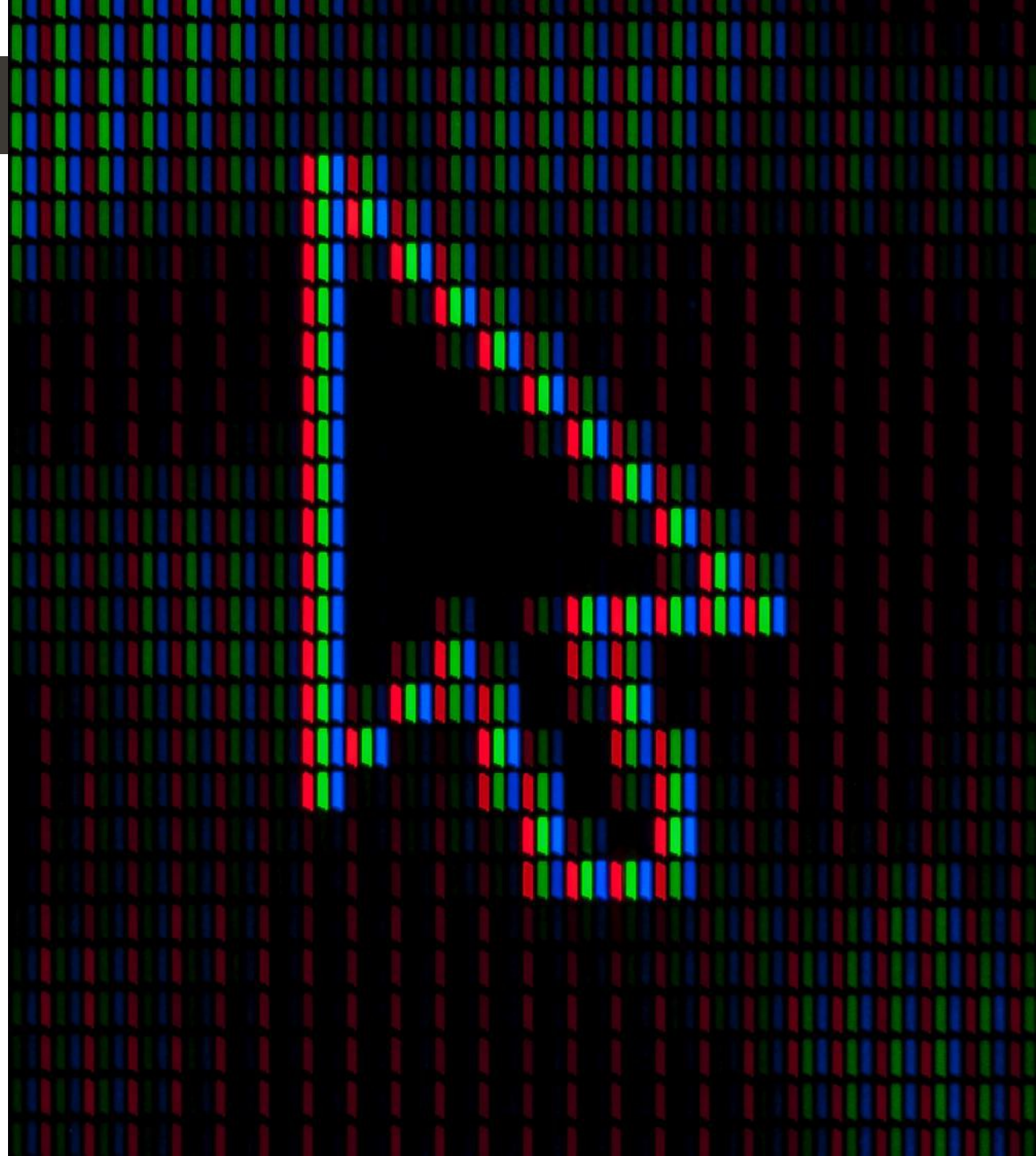


chap 7

타이머로 구현한 남은 글자 수 세기



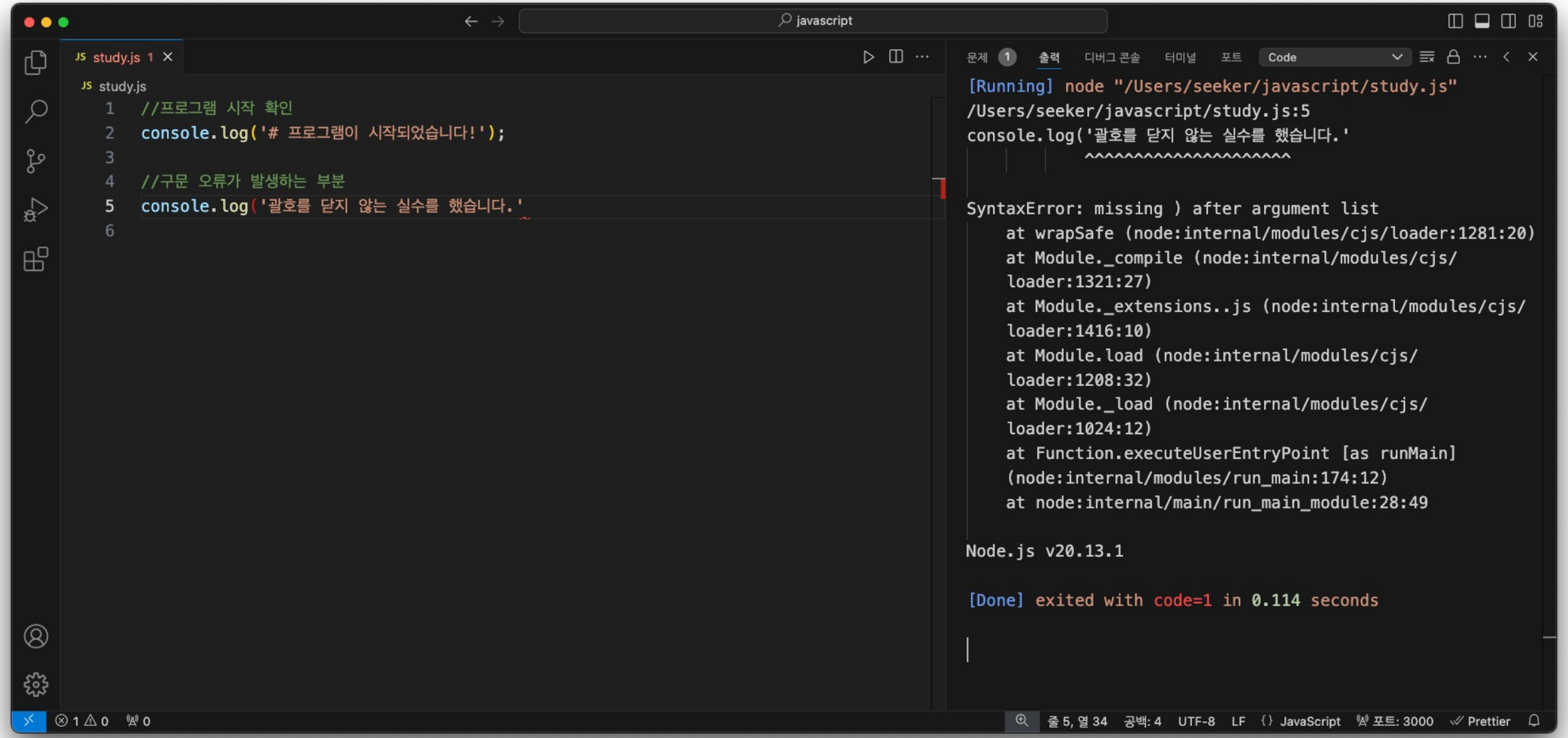
목차 a table of contents



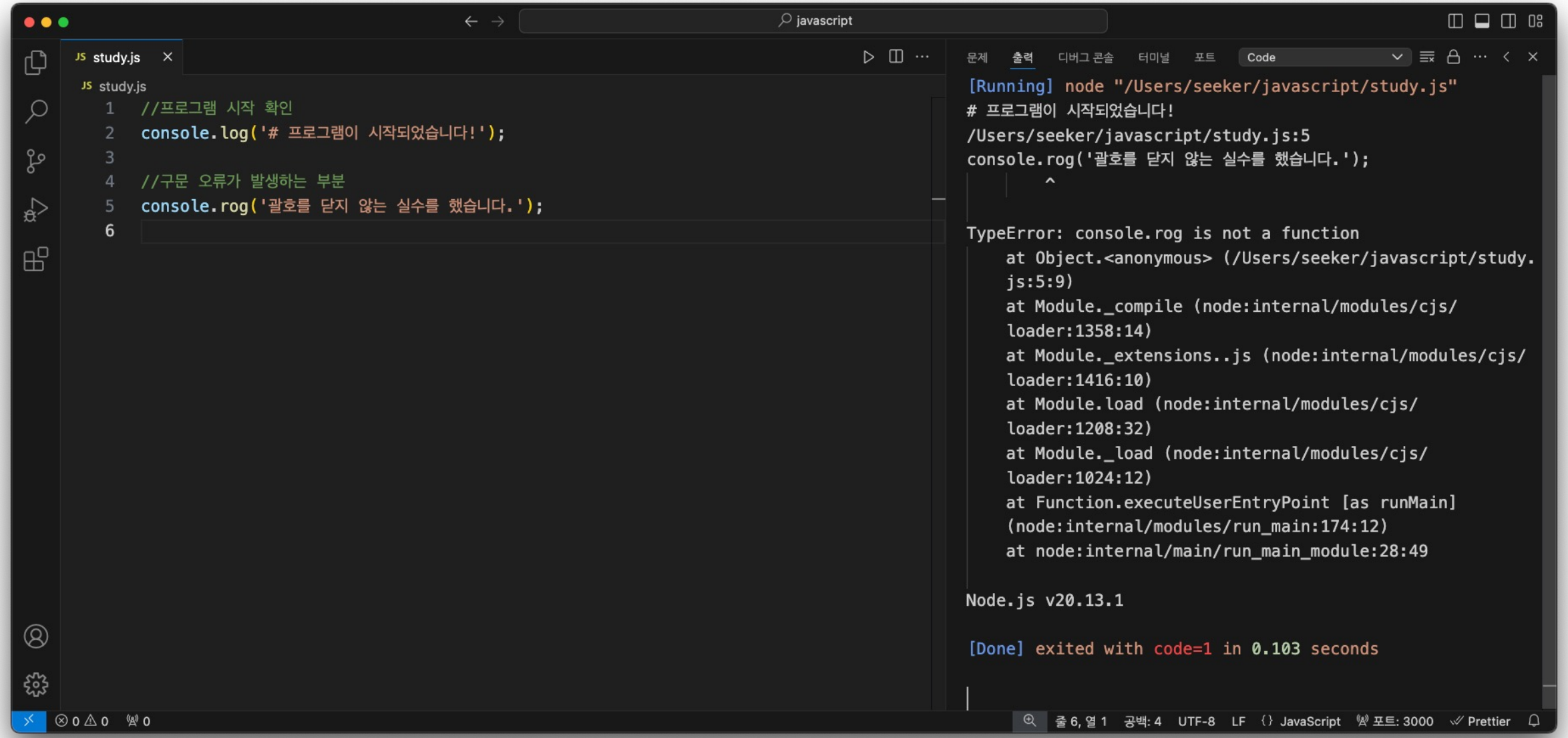
구문 오류와 예외

chap 8

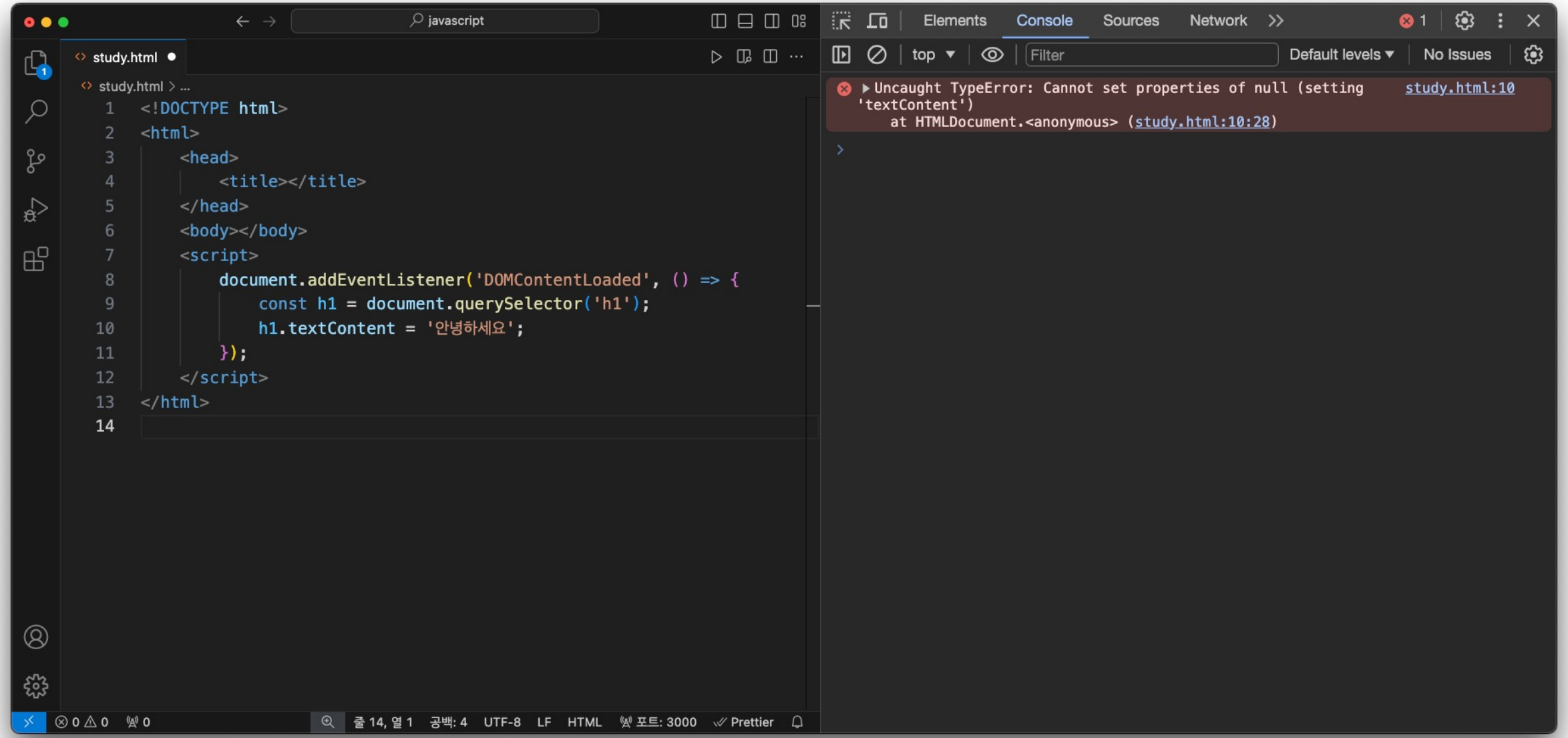
오류의 종류 > 구문 오류



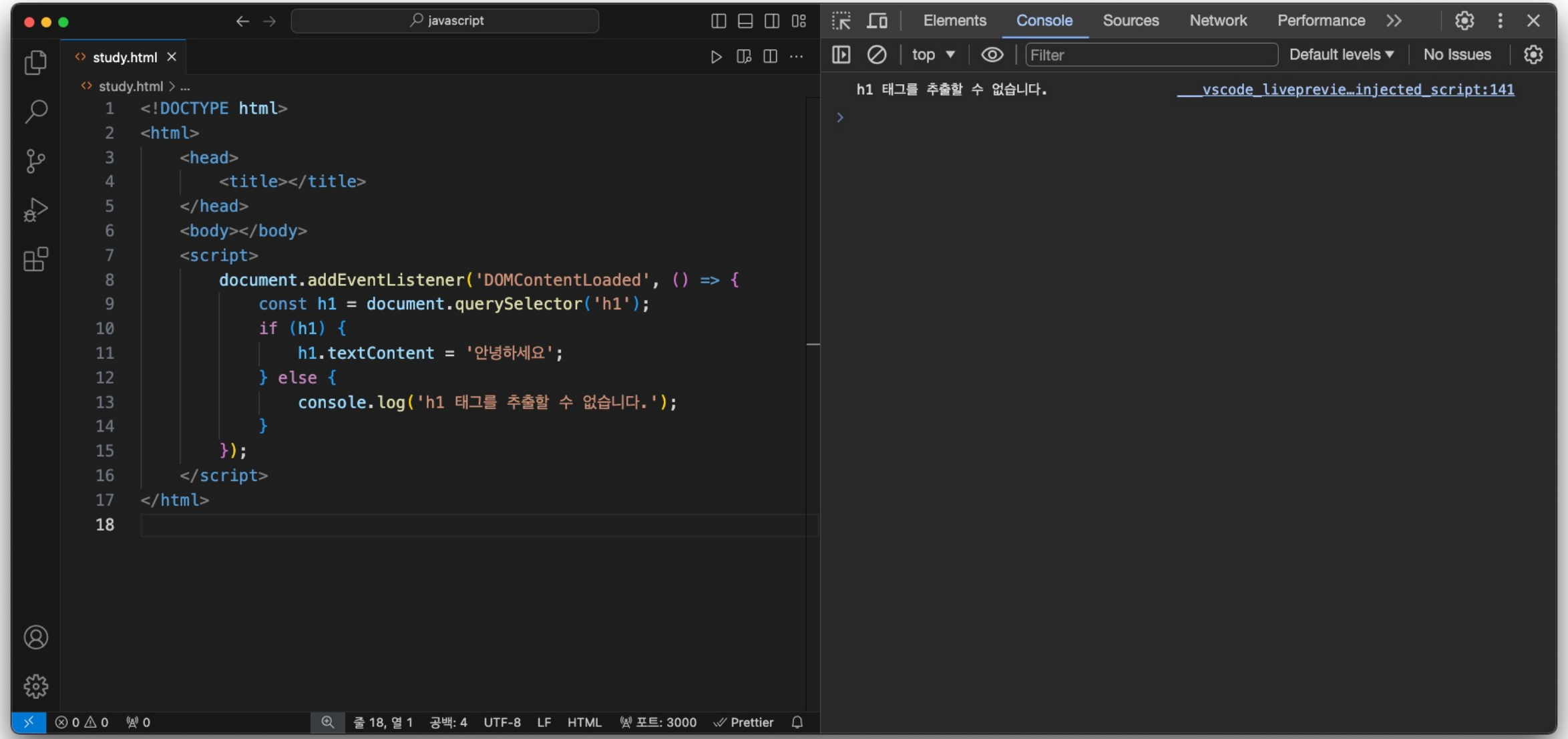
chap 8 오류의 종류 > 예외



chap 8 기본 예외 처리

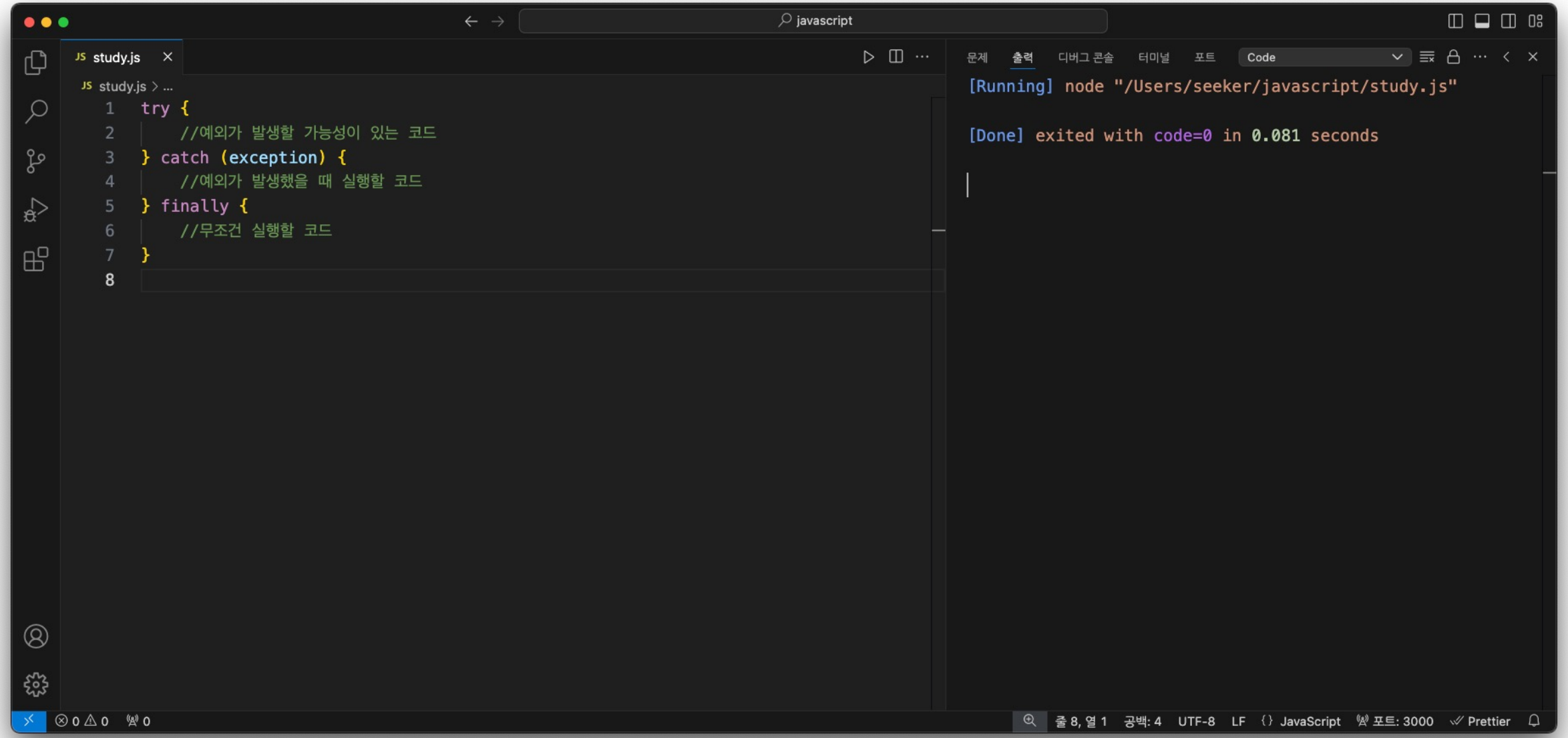


chap 8 기본 예외 처리

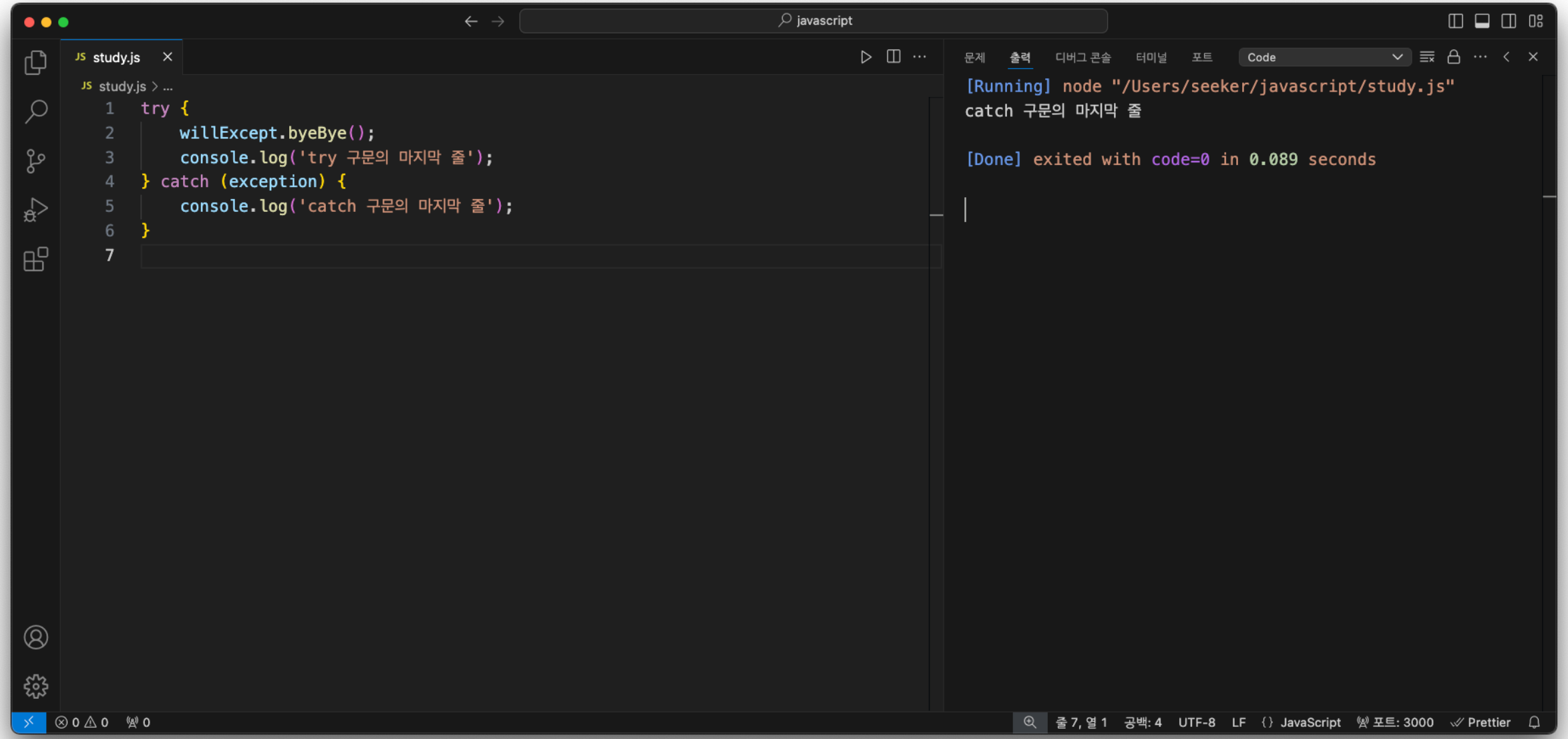


chap 8

고급 예외 처리

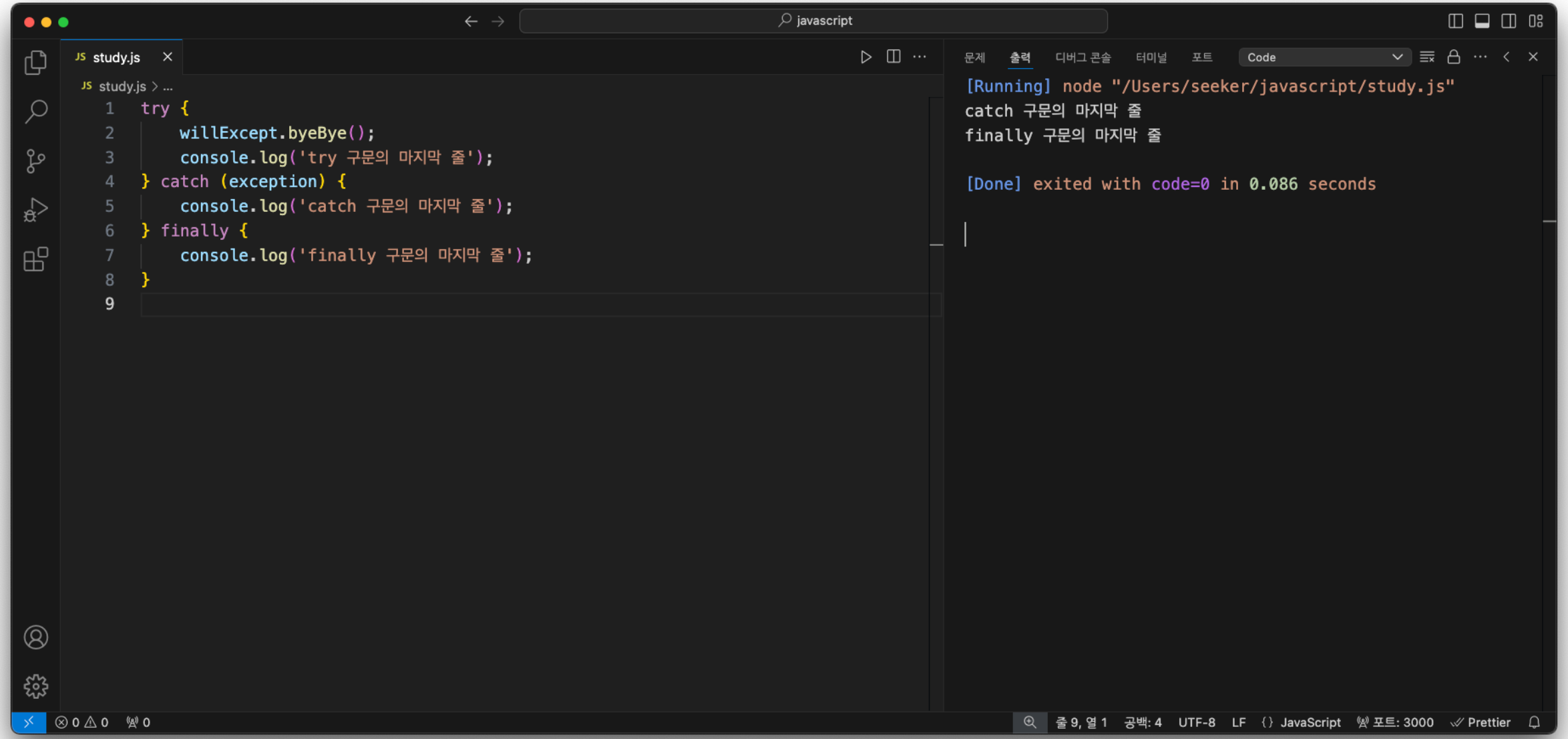


chap 8 고급 예외 처리



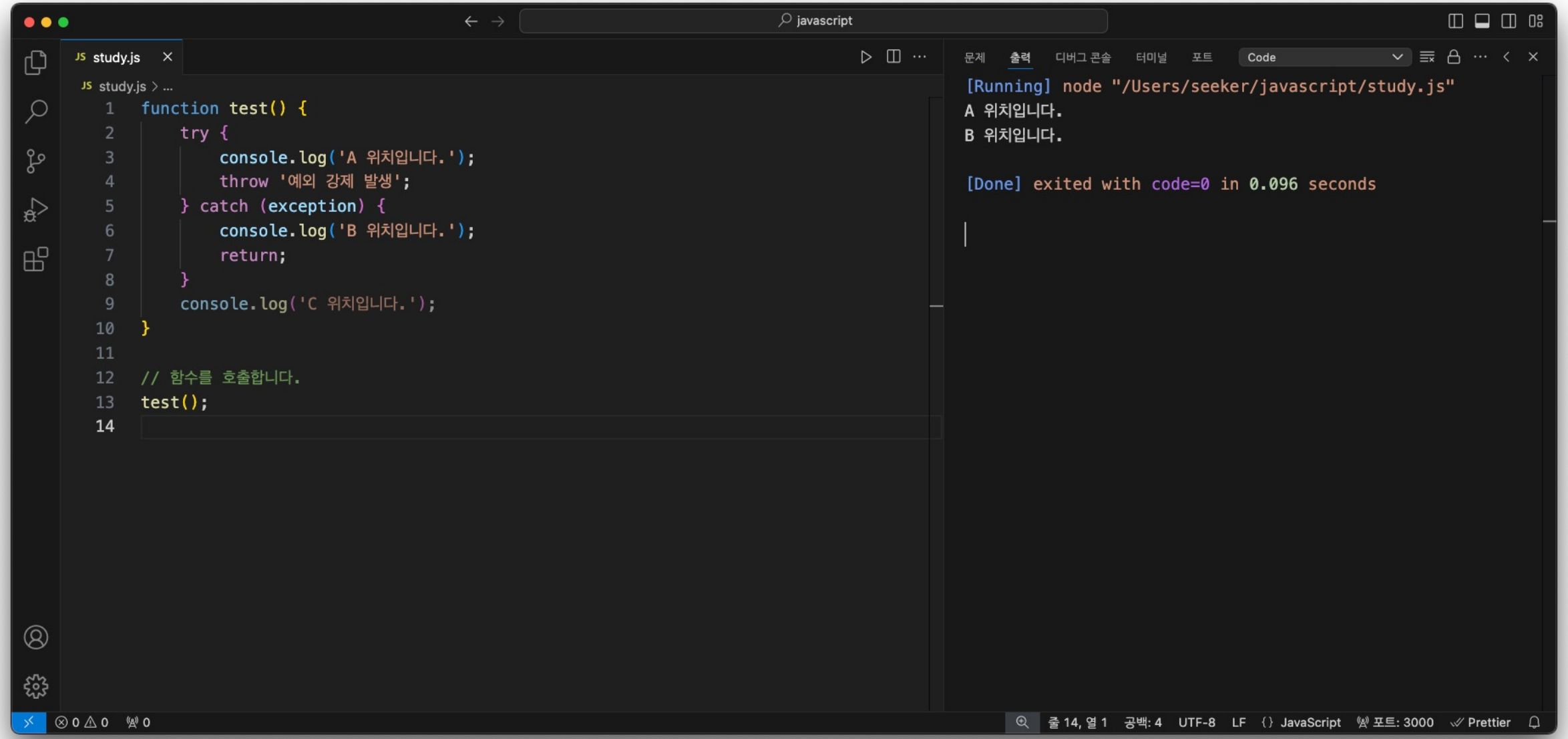
chap 8

고급 예외 처리



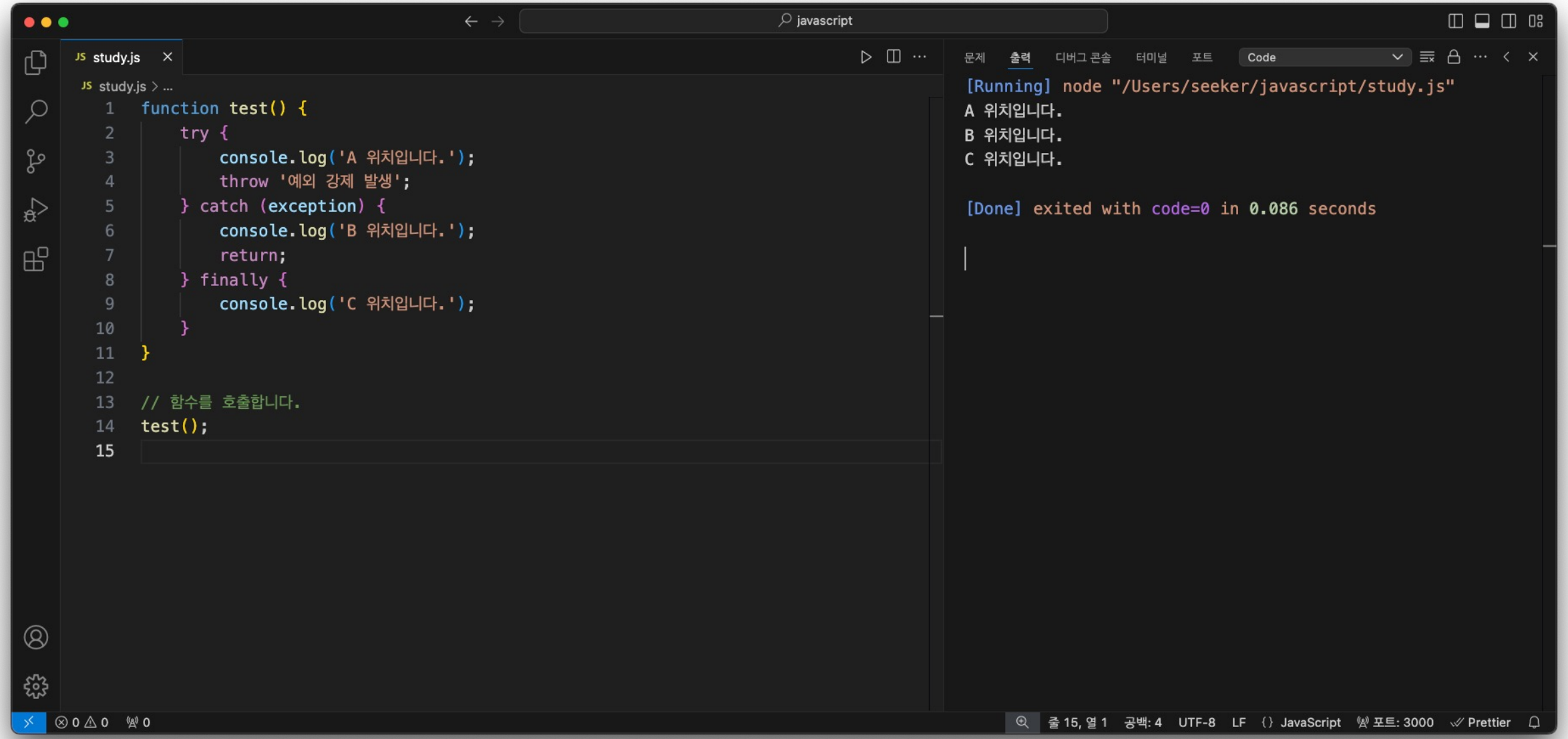
chap 8

finally 구문을 사용하는 이유



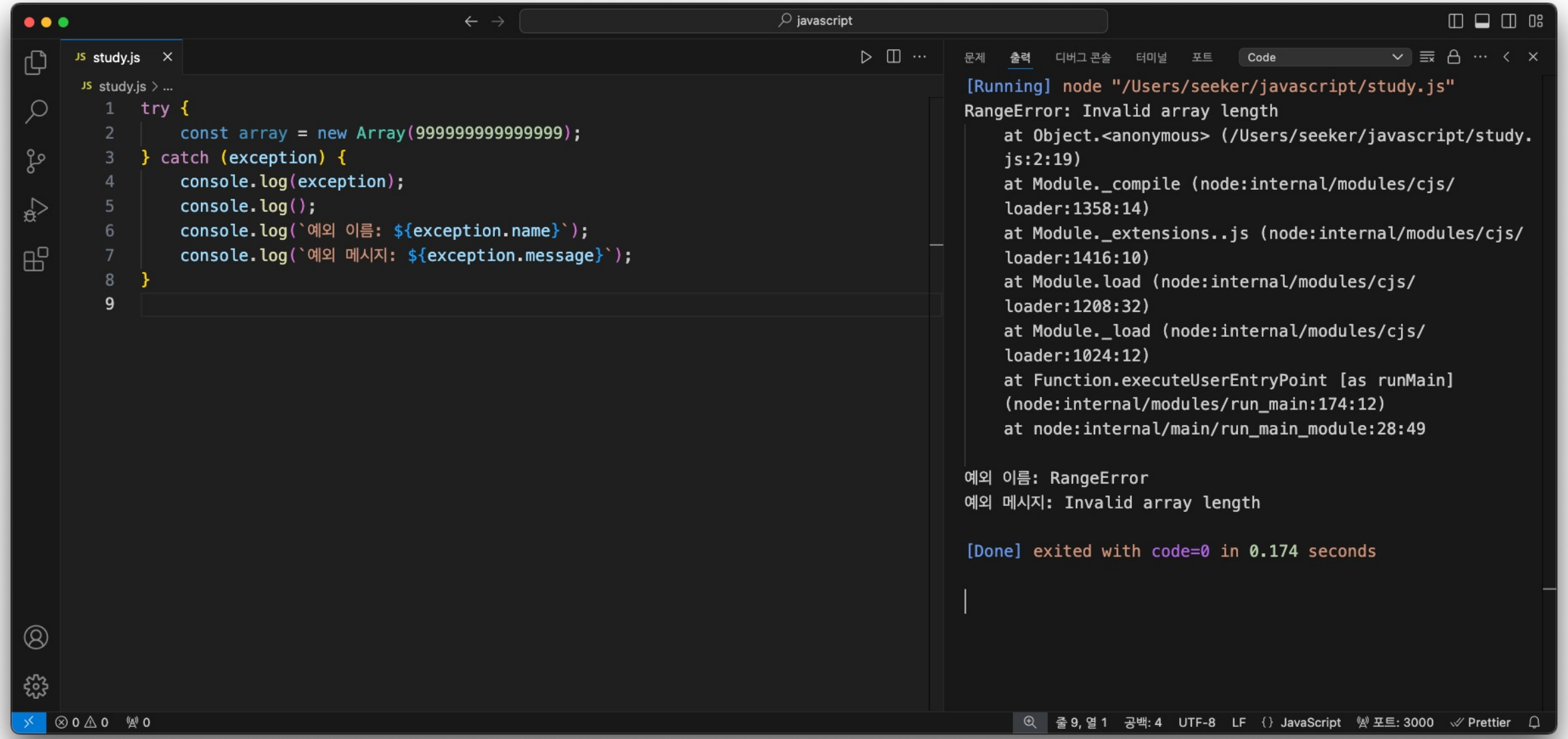
chap 8

finally 구문을 사용하는 이유



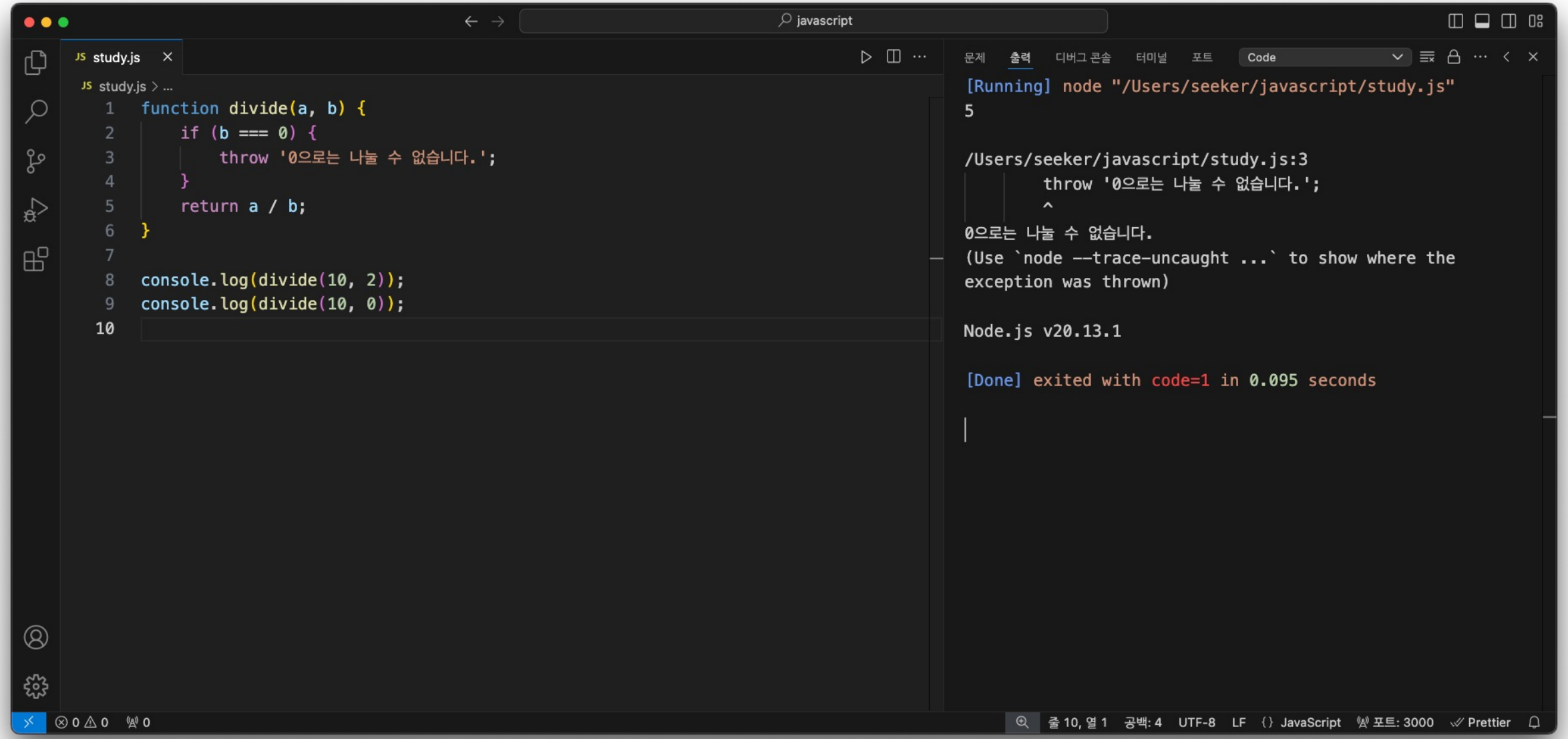
예외 처리 고급

chap 8 예외 객체


javascript
study.js
study.js > ...
1 try {
2 const array = new Array(999999999999999);
3 } catch (exception) {
4 console.log(exception);
5 console.log();
6 console.log(`예외 이름: ${exception.name}`);
7 console.log(`예외 메시지: ${exception.message}`);
8 }
9
문제 출력 디버그 콘솔 터미널 포트 Code
[Running] node "/Users/seeker/javascript/study.js"
RangeError: Invalid array length
at Object.<anonymous> (/Users/seeker/javascript/study.js:2:19)
at Module._compile (node:internal/modules/cjs/loader:1358:14)
at Module._extensions..js (node:internal/modules/cjs/loader:1416:10)
at Module.load (node:internal/modules/cjs/loader:1208:32)
at Module._load (node:internal/modules/cjs/loader:1024:12)
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:174:12)
at node:internal/main/run_main_module:28:49
예외 이름: RangeError
예외 메시지: Invalid array length
[Done] exited with code=0 in 0.174 seconds
⊗ 0 ⚠ 0 0
줄 9, 열 1 공백: 4 UTF-8 LF { } JavaScript 포트: 3000 Prettier

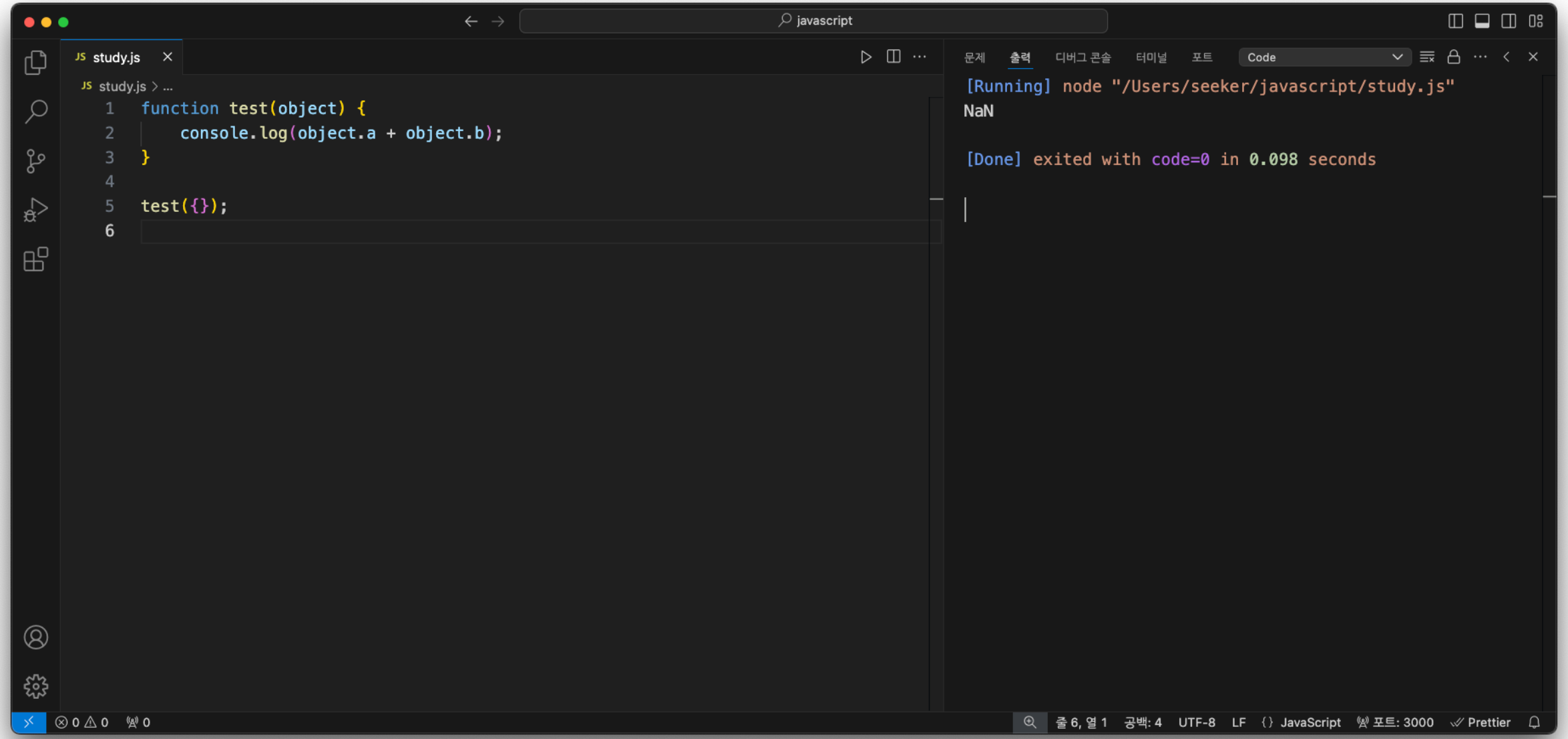
chap 8

예외 강제 발생



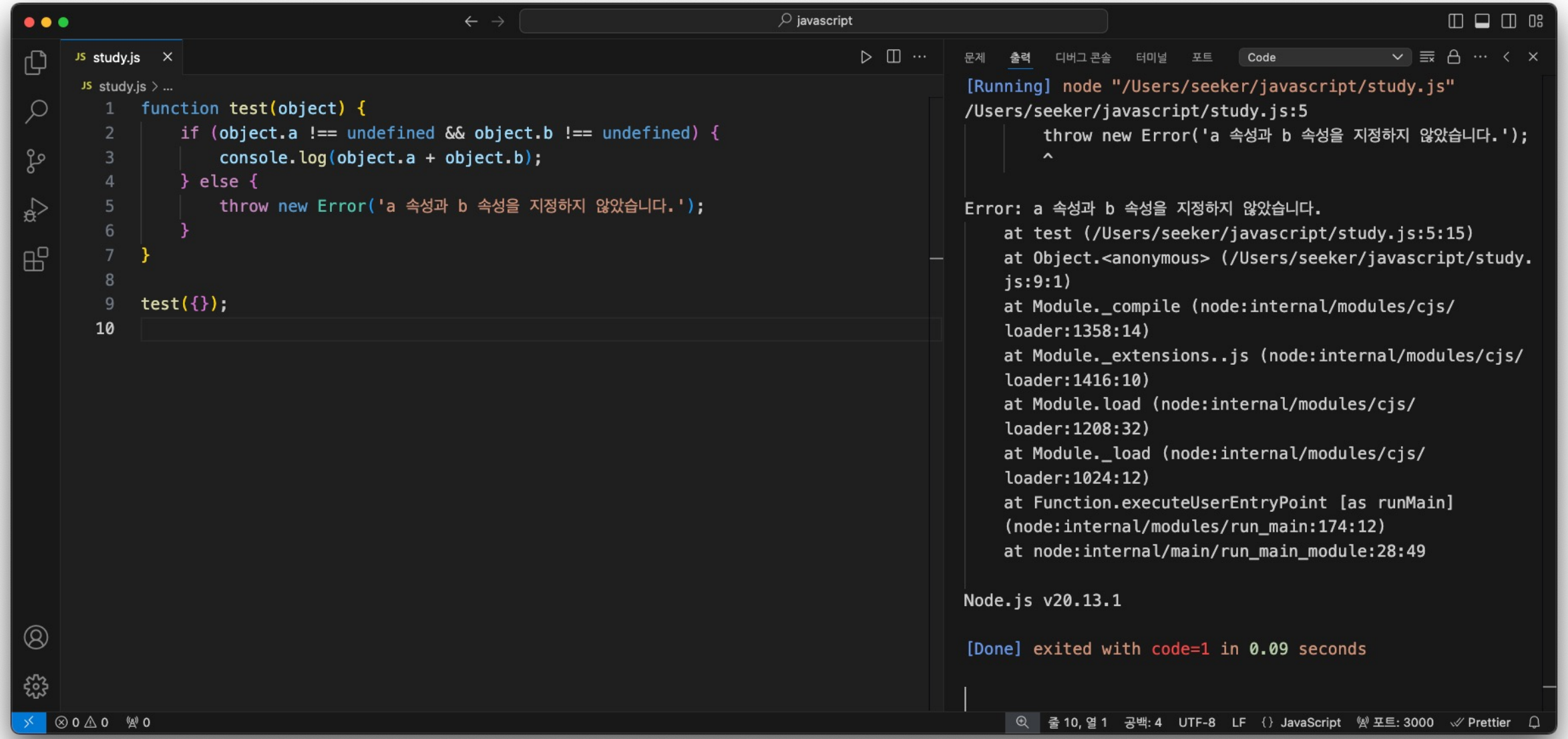
chap 8

예외 강제 발생

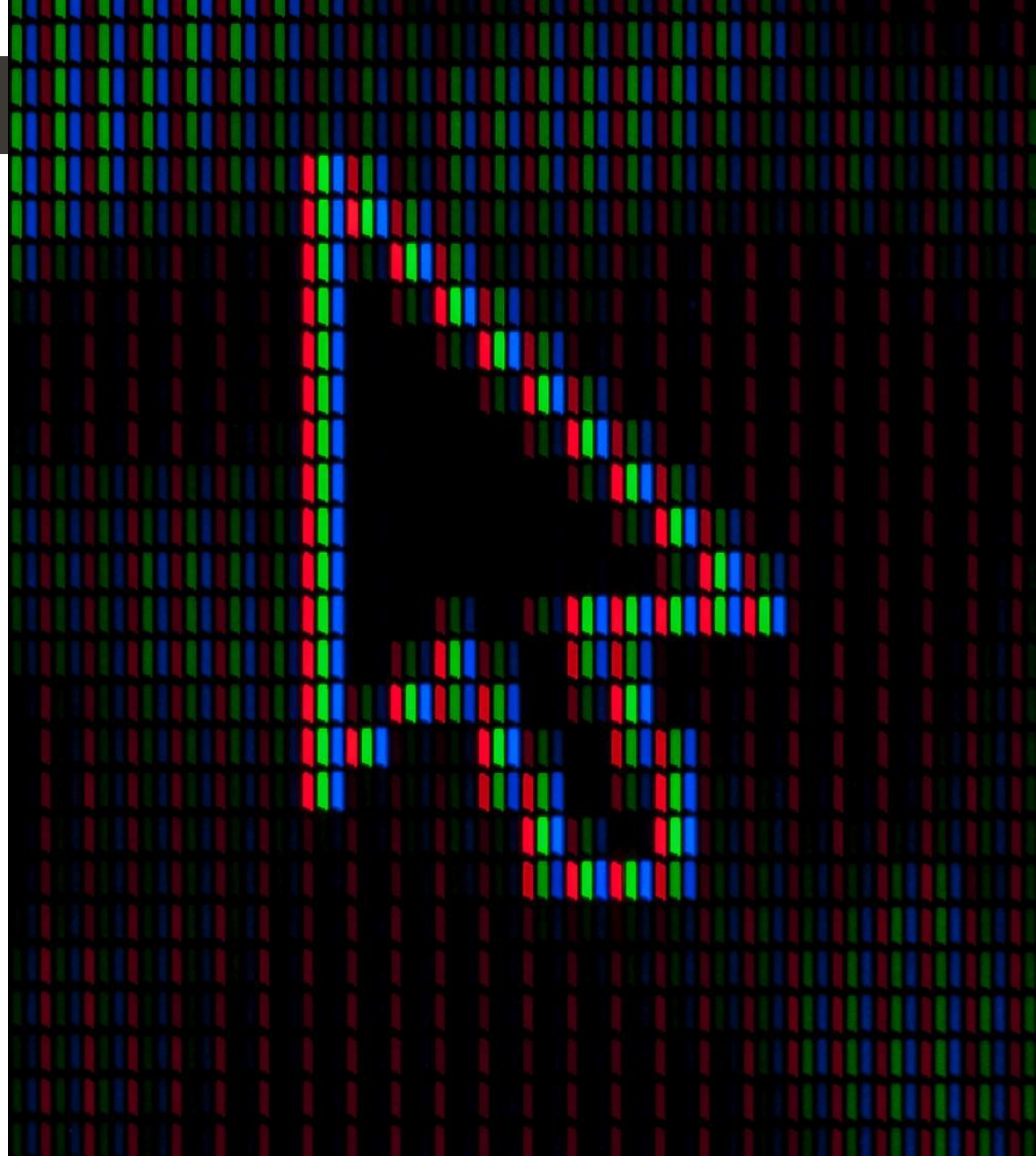


chap 8

예외 강제 발생



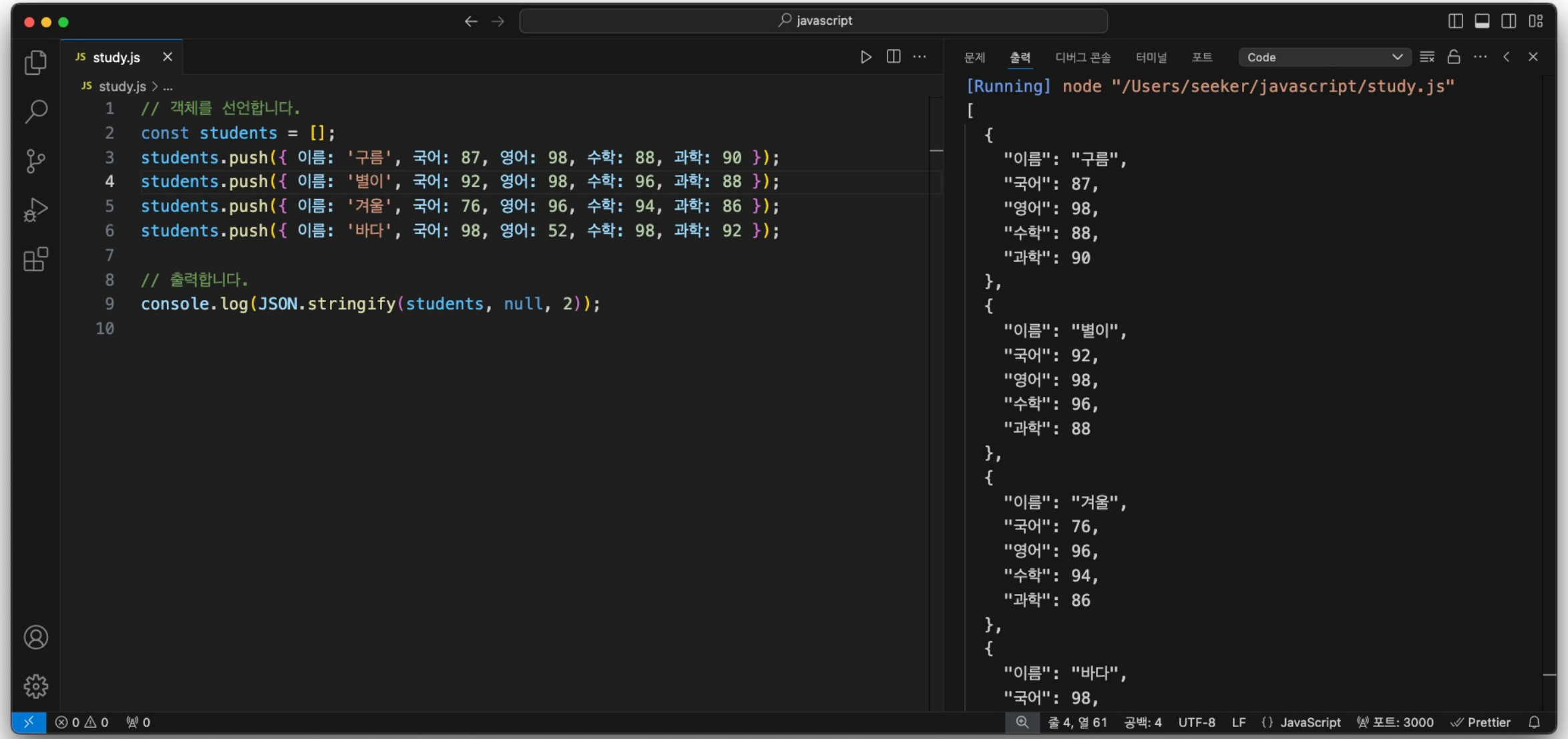
목차 a table of contents



기본기능

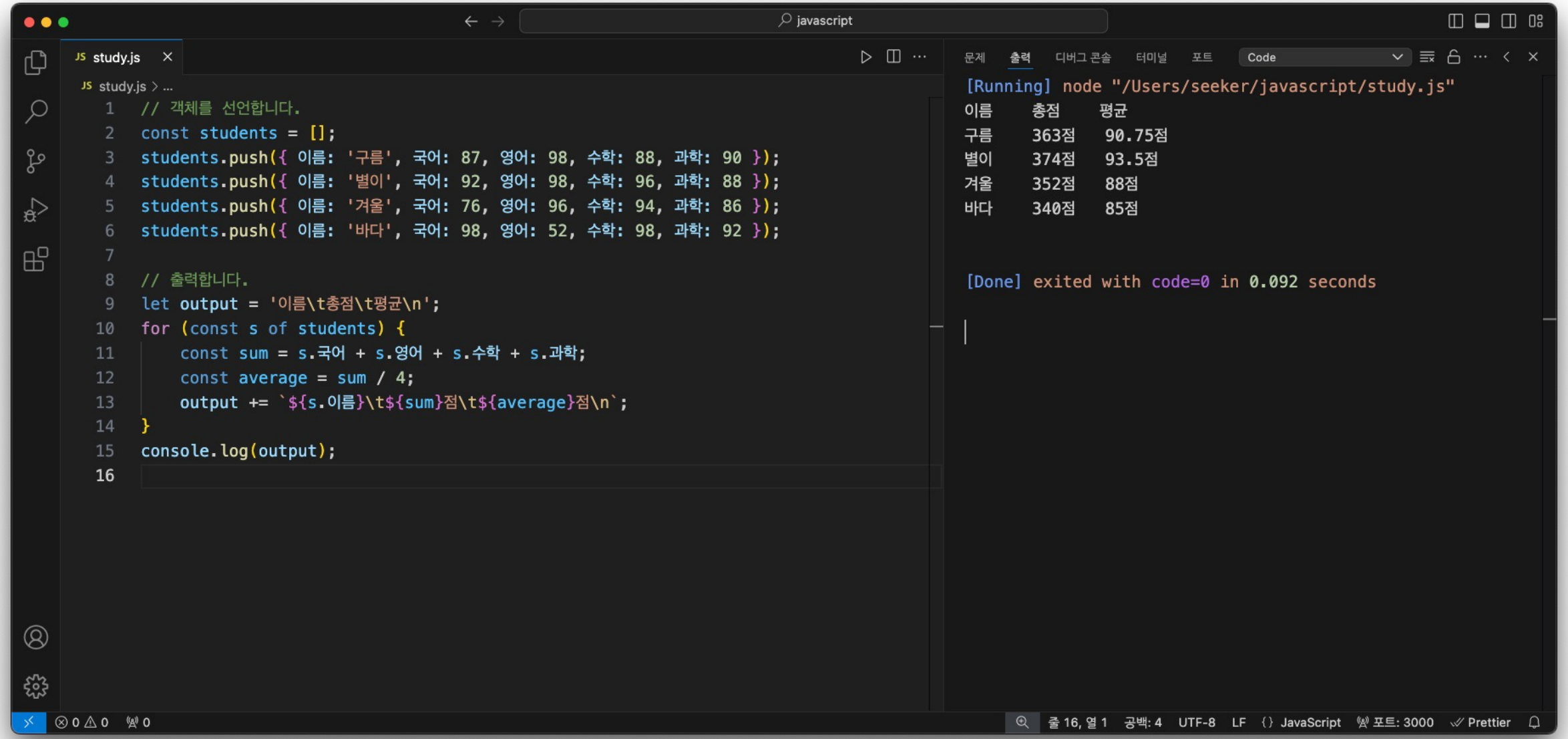
chap 9

같은 형태의 객체 만들기



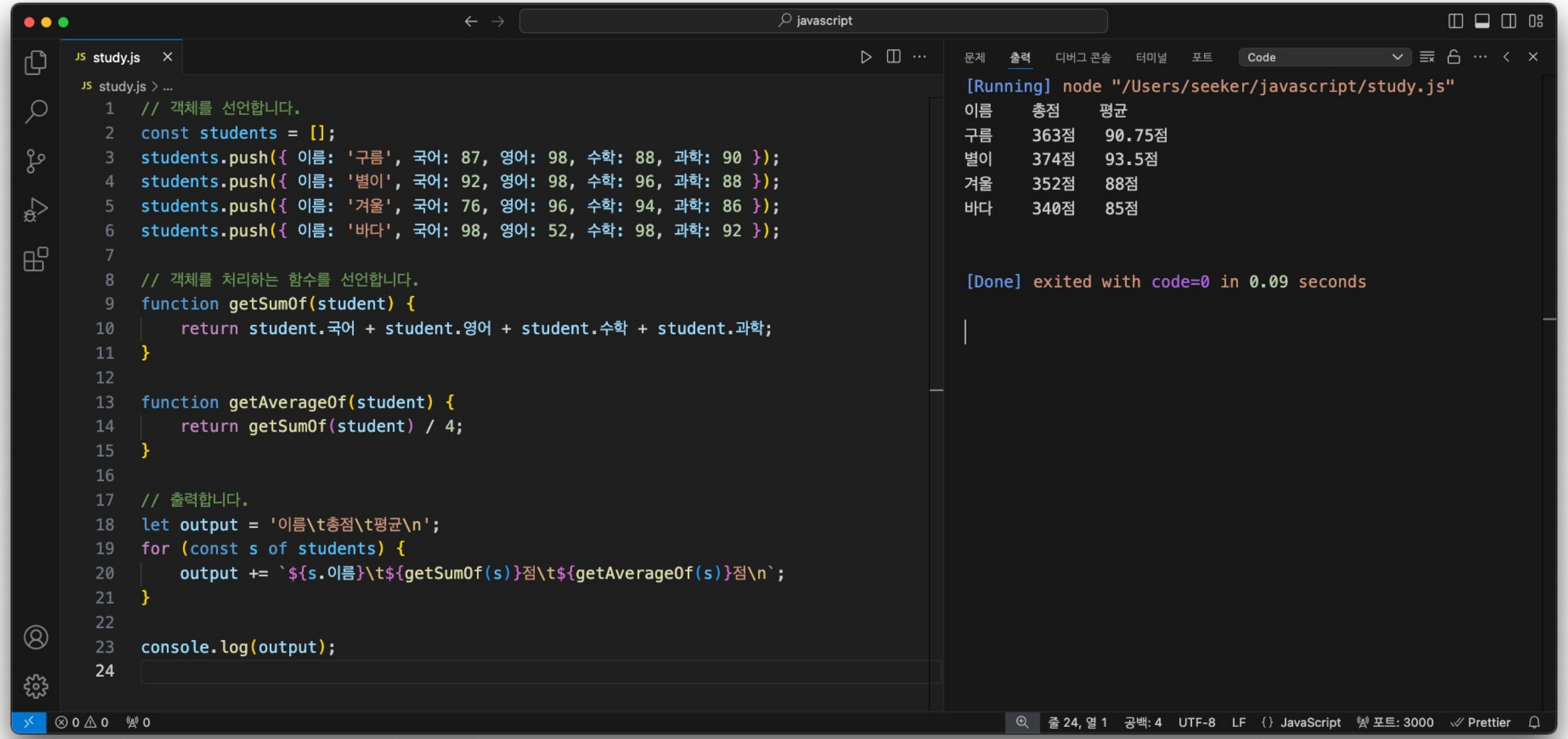
chap 9

같은 형태의 객체 만들기



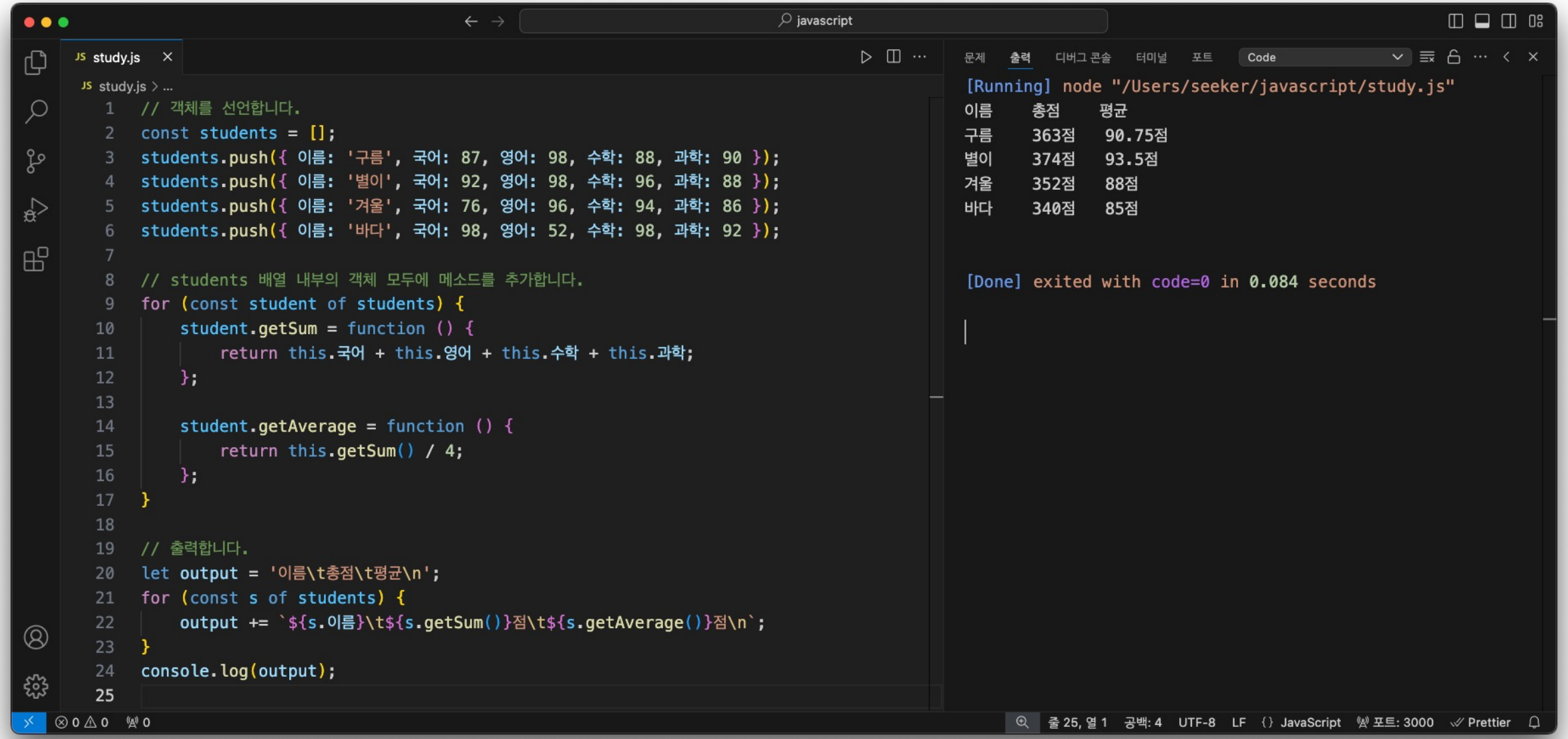
chap 9

객체를 처리하는 함수



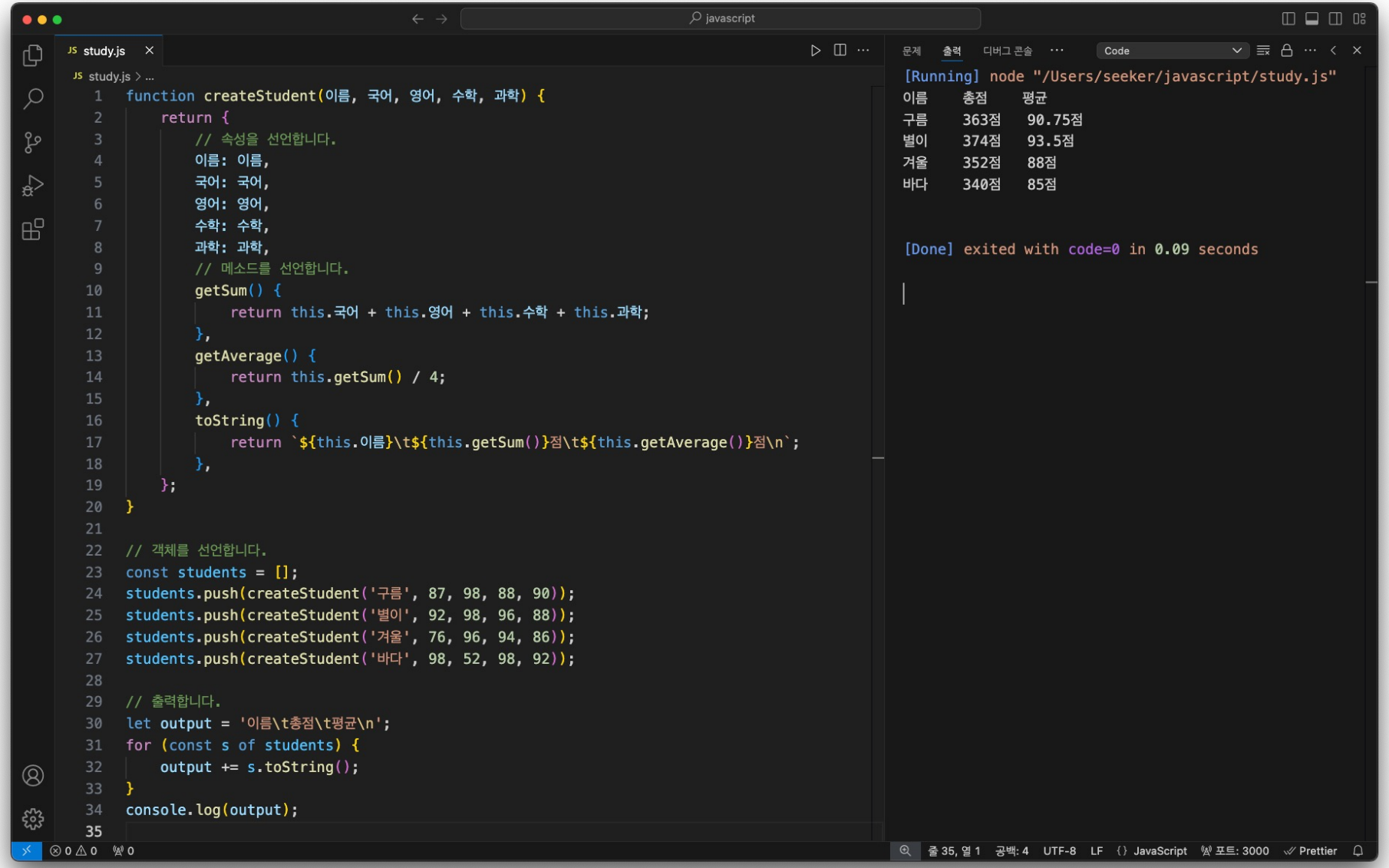
chap 9

객체의 기능을 메소드로 추가하기



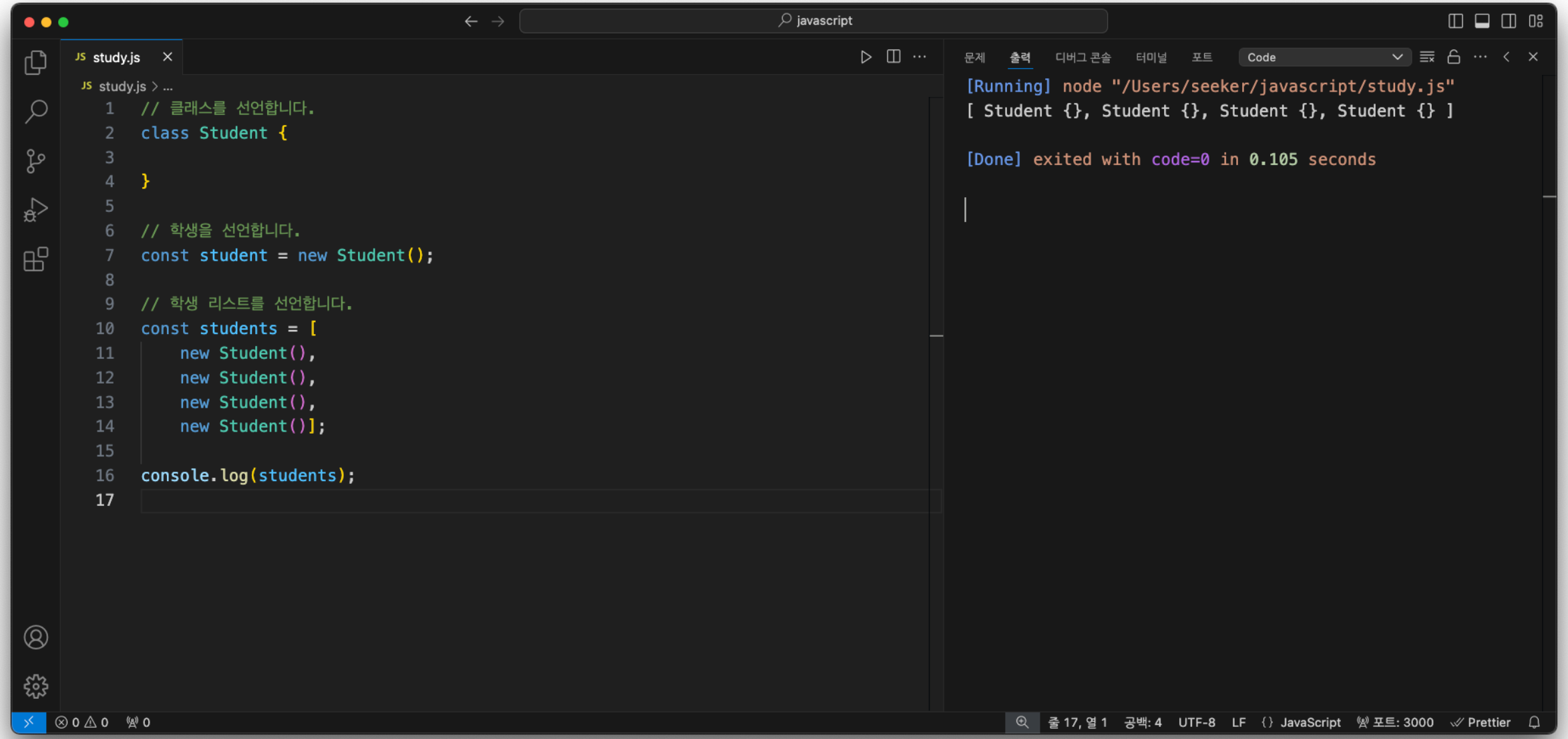
chap 9

객체의 기능을 메소드로 추가하기

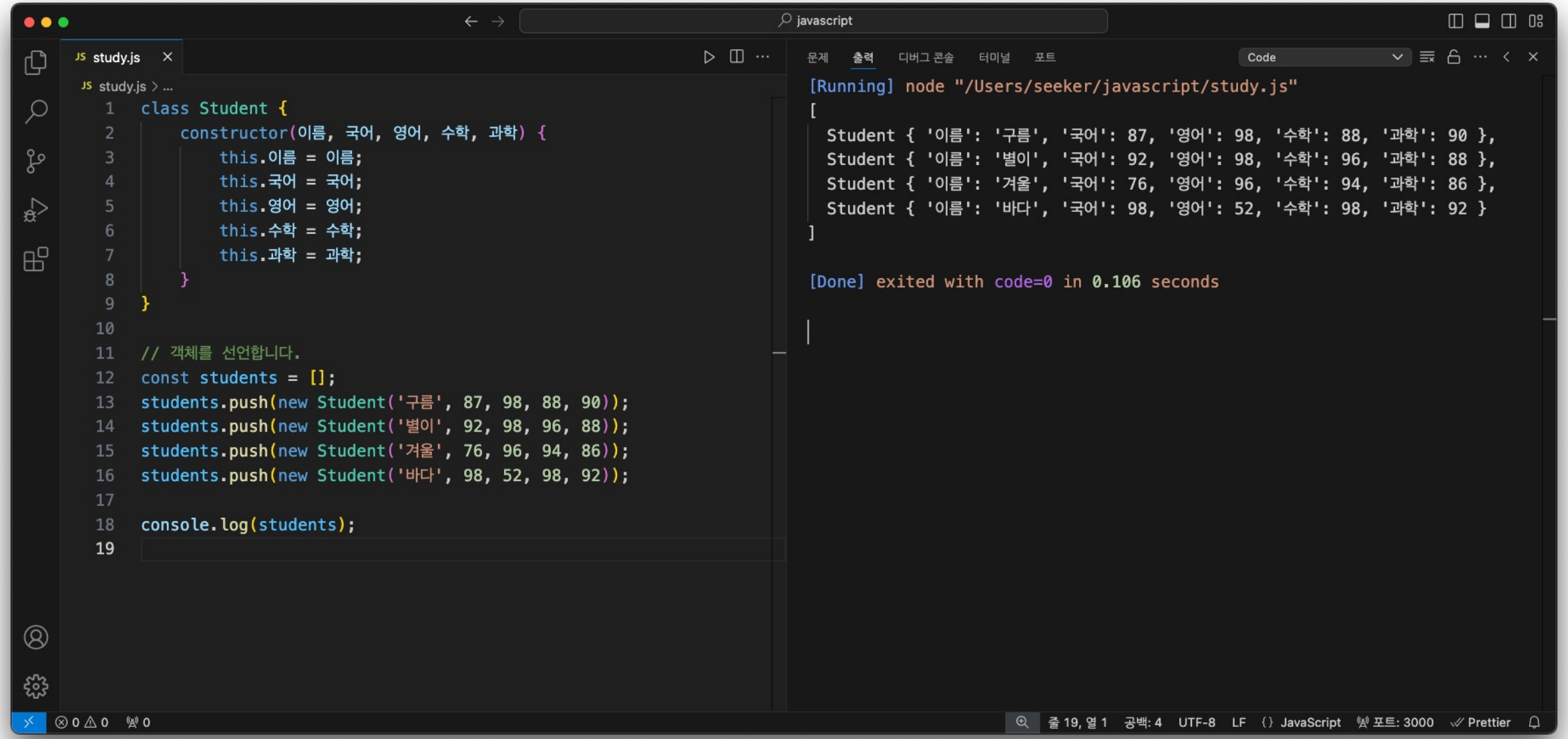


chap 9

클래스 선언하기

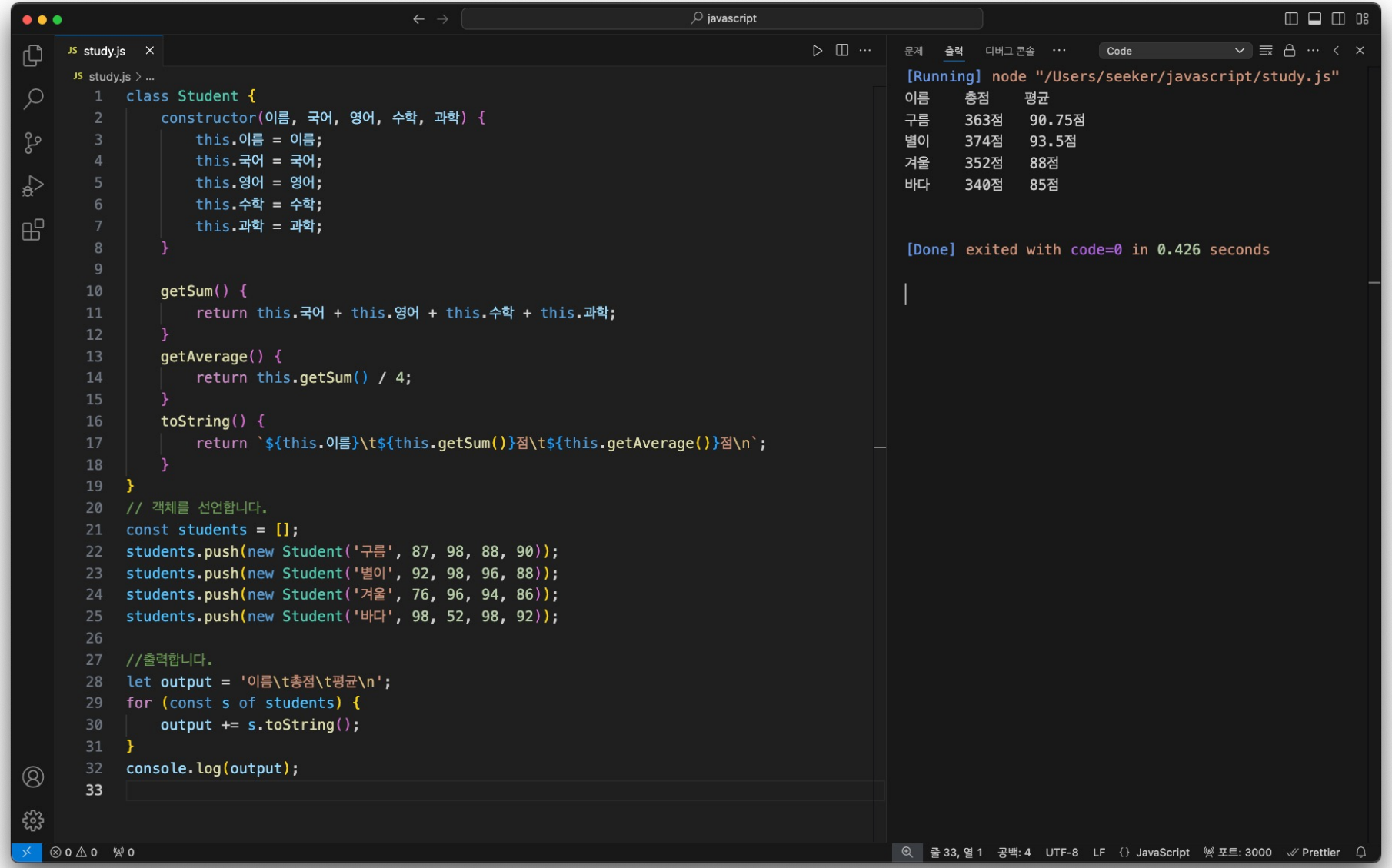


chap 9 생성자


javascript
study.js
study.js > ...
1 class Student {
2 constructor(이름, 국어, 영어, 수학, 과학) {
3 this.이름 = 이름;
4 this.국어 = 국어;
5 this.영어 = 영어;
6 this.수학 = 수학;
7 this.과학 = 과학;
8 }
9 }
10
11 // 객체를 선언합니다.
12 const students = [];
13 students.push(new Student('구름', 87, 98, 88, 90));
14 students.push(new Student('별이', 92, 98, 96, 88));
15 students.push(new Student('겨울', 76, 96, 94, 86));
16 students.push(new Student('바다', 98, 52, 98, 92));
17
18 console.log(students);
19
문제 출력 디버그 콘솔 터미널 포트
Code
[Running] node "/Users/seeker/javascript/study.js"
[
Student { '이름': '구름', '국어': 87, '영어': 98, '수학': 88, '과학': 90 },
Student { '이름': '별이', '국어': 92, '영어': 98, '수학': 96, '과학': 88 },
Student { '이름': '겨울', '국어': 76, '영어': 96, '수학': 94, '과학': 86 },
Student { '이름': '바다', '국어': 98, '영어': 52, '수학': 98, '과학': 92 }
]
[Done] exited with code=0 in 0.106 seconds
⊗ 0 ⚠ 0 0
줄 19, 열 1 공백: 4 UTF-8 LF { } JavaScript 포트: 3000 Prettier

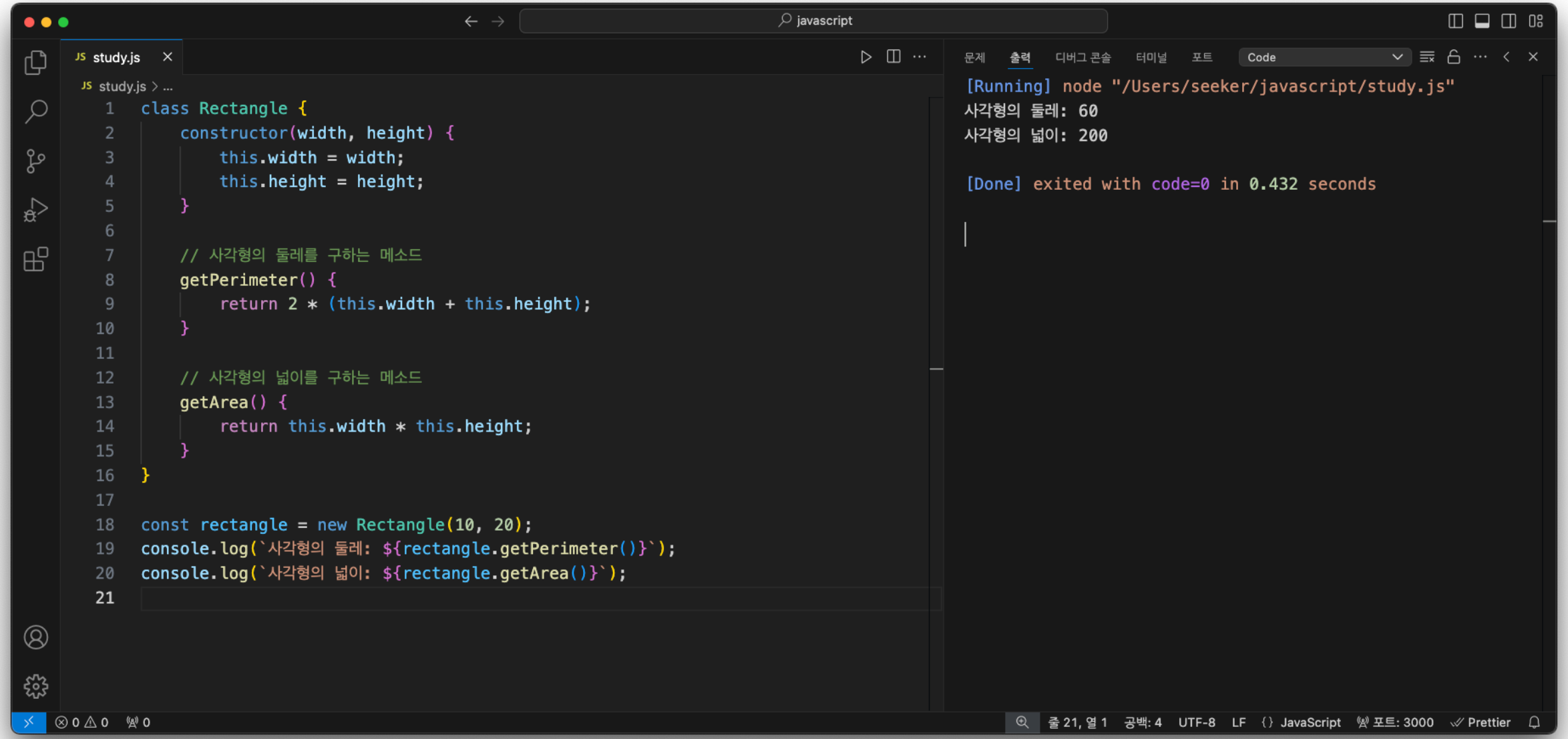
chap 9

메소드

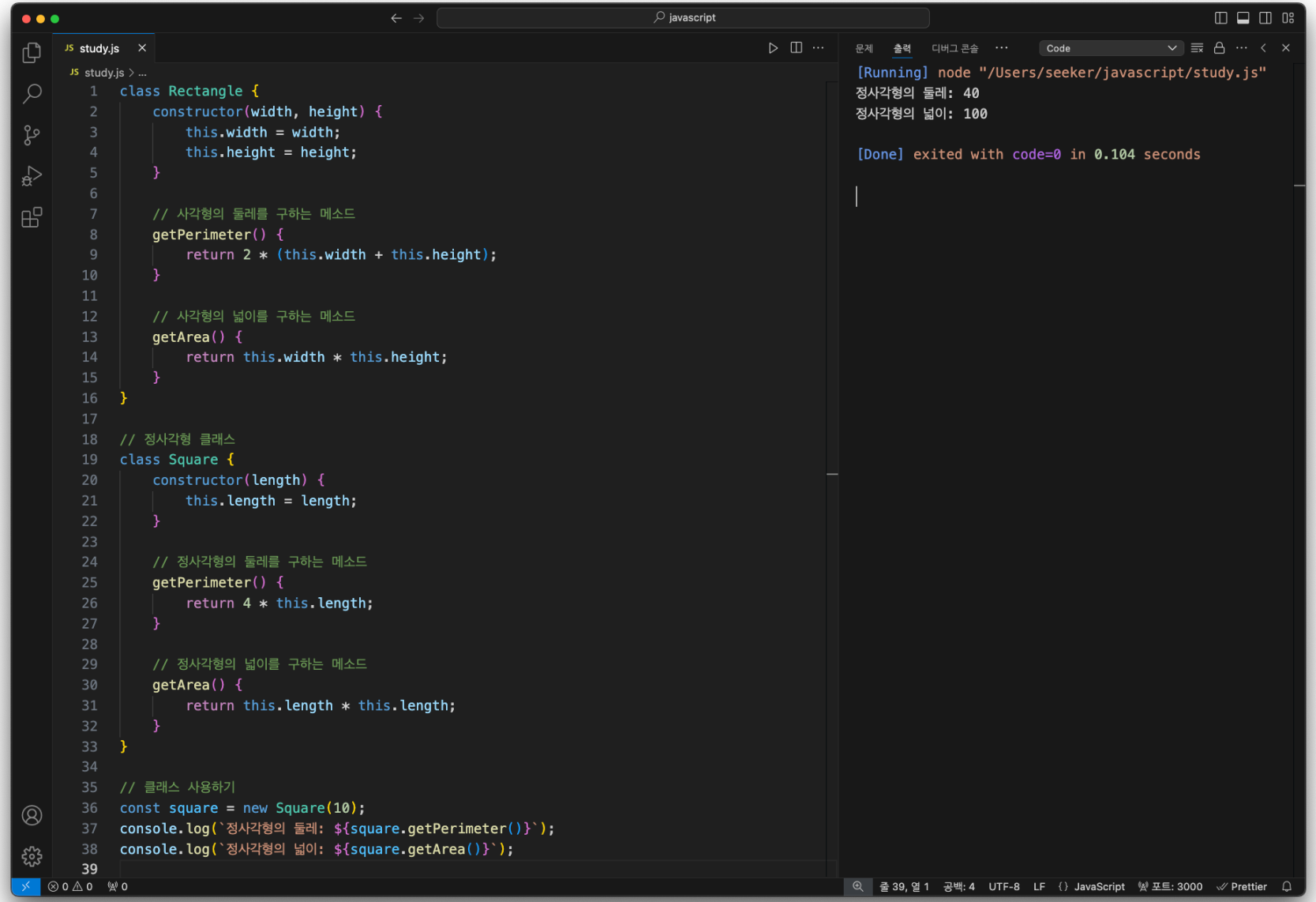

javascript
study.js
study.js > ...
class Student {
constructor(이름, 국어, 영어, 수학, 과학) {
this.이름 = 이름;
this.국어 = 국어;
this.영어 = 영어;
this.수학 = 수학;
this.과학 = 과학;
}
getSum() {
return this.국어 + this.영어 + this.수학 + this.과학;
}
getAverage() {
return this.getSum() / 4;
}
toString() {
return `${this.이름}\t${this.getSum()}점\t${this.getAverage()}점\n`;
}
}
// 객체를 선언합니다.
const students = [];
students.push(new Student('구름', 87, 98, 88, 90));
students.push(new Student('별이', 92, 98, 96, 88));
students.push(new Student('겨울', 76, 96, 94, 86));
students.push(new Student('바다', 98, 52, 98, 92));
//출력합니다.
let output = '이름\t총점\t평균\n';
for (const s of students) {
output += s.toString();
}
console.log(output);
문제
출력
디버그 콘솔
Code
[Running] node "/Users/seeker/javascript/study.js"
이름 총점 평균
구름 363점 90.75점
별이 374점 93.5점
겨울 352점 88점
바다 340점 85점
[Done] exited with code=0 in 0.426 seconds
줄 33, 열 1
공백: 4
UTF-8
LF
JavaScript
포트: 3000
Prettier

고급 기능

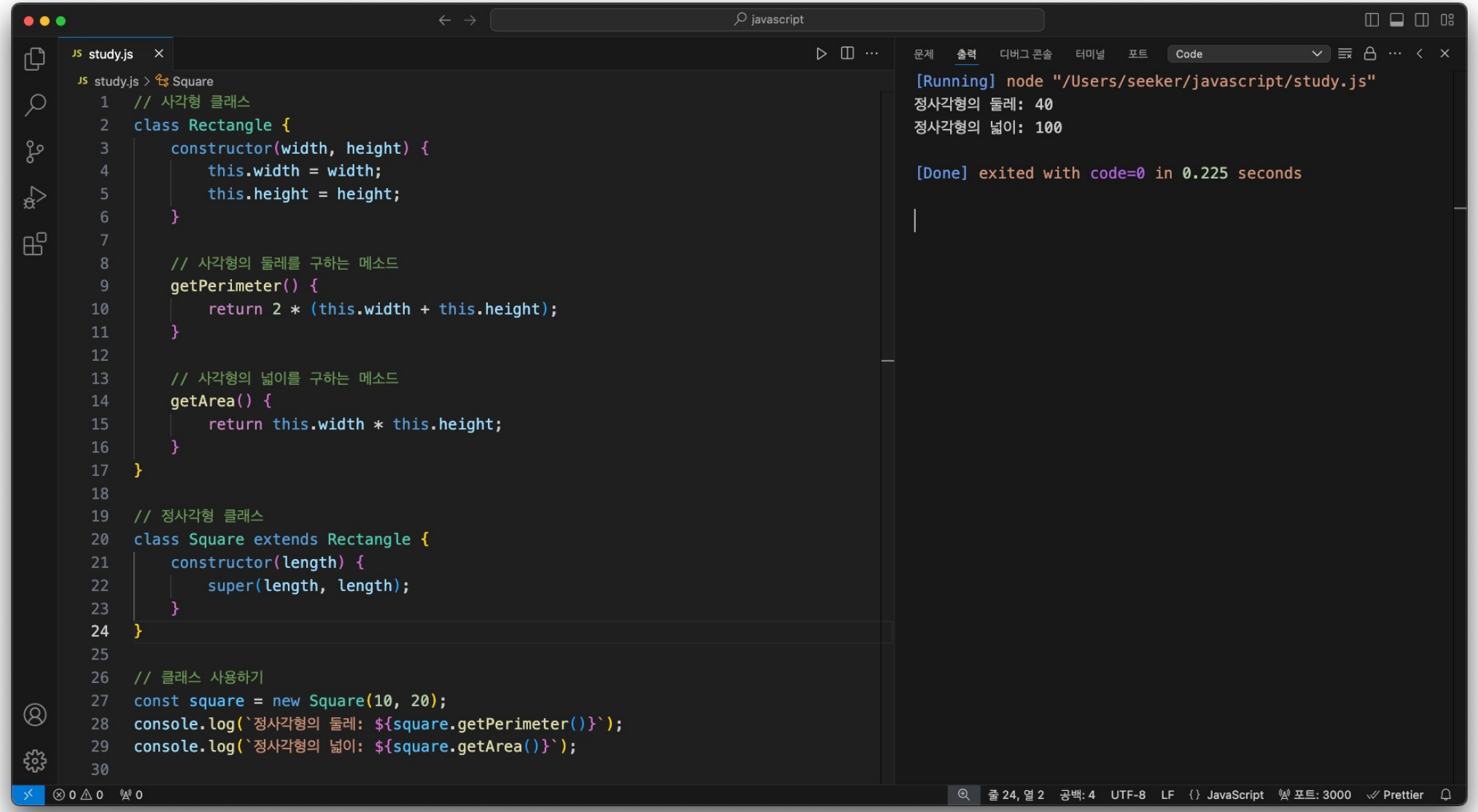
chap 9 상속


javascript
study.js
study.js > ...
1 class Rectangle {
2 constructor(width, height) {
3 this.width = width;
4 this.height = height;
5 }
6
7 // 사각형의 둘레를 구하는 메소드
8 getPerimeter() {
9 return 2 * (this.width + this.height);
10 }
11
12 // 사각형의 넓이를 구하는 메소드
13 getArea() {
14 return this.width * this.height;
15 }
16 }
17
18 const rectangle = new Rectangle(10, 20);
19 console.log(`사각형의 둘레: ${rectangle.getPerimeter()}`);
20 console.log(`사각형의 넓이: ${rectangle.getArea()}`);
21
문제 출력 디버그 콘솔 터미널 포트 Code
[Running] node "/Users/seeker/javascript/study.js"
사각형의 둘레: 60
사각형의 넓이: 200
[Done] exited with code=0 in 0.432 seconds
줄 21, 열 1 공백: 4 UTF-8 LF JavaScript 포트: 3000 Prettier

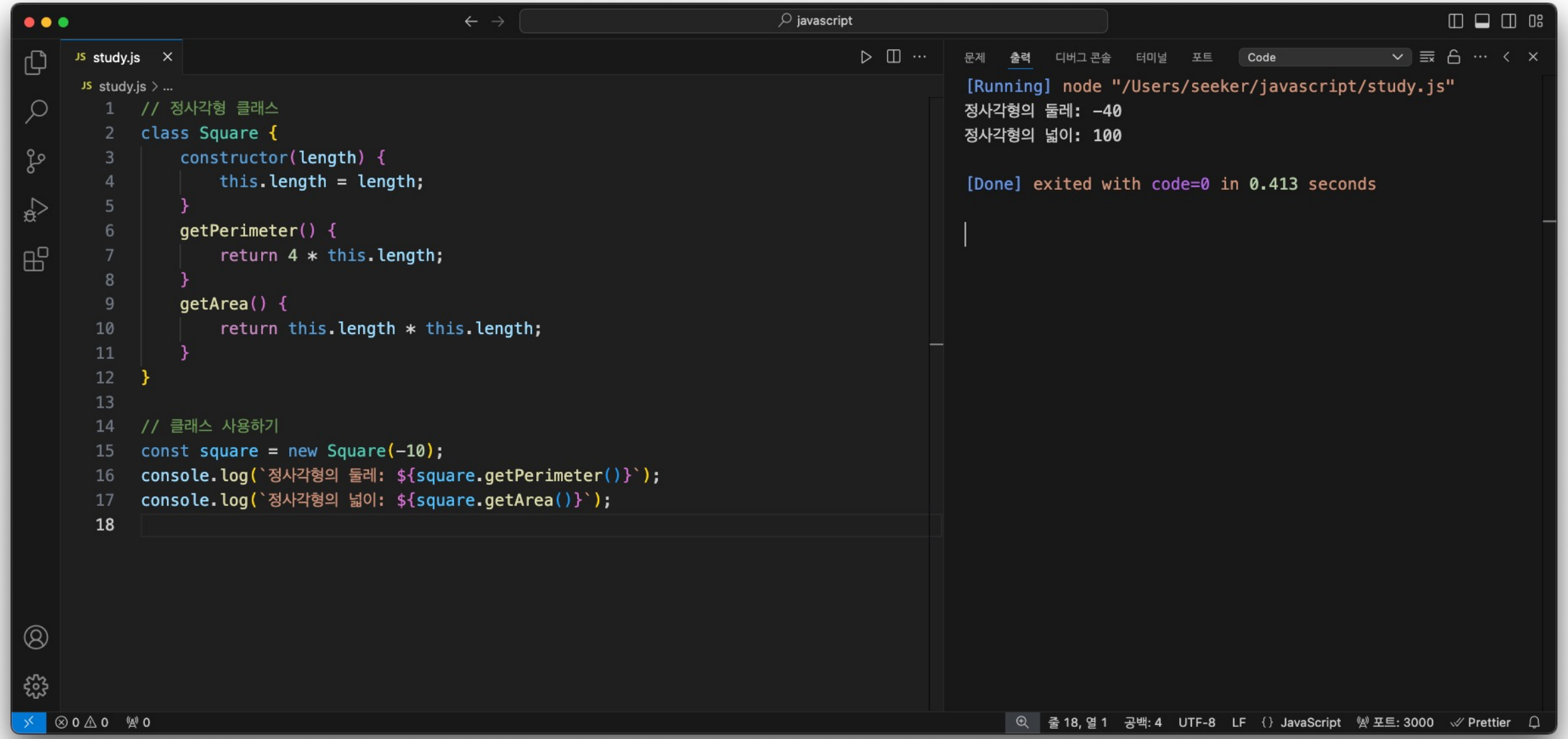
chap 9 상속


javascript
study.js
study.js > ...
class Rectangle {
constructor(width, height) {
this.width = width;
this.height = height;
}
// 사각형의 둘레를 구하는 메소드
getPerimeter() {
return 2 * (this.width + this.height);
}
// 사각형의 넓이를 구하는 메소드
getArea() {
return this.width * this.height;
}
}
// 정사각형 클래스
class Square {
constructor(length) {
this.length = length;
}
// 정사각형의 둘레를 구하는 메소드
getPerimeter() {
return 4 * this.length;
}
// 정사각형의 넓이를 구하는 메소드
getArea() {
return this.length * this.length;
}
}
// 클래스 사용하기
const square = new Square(10);
console.log(`정사각형의 둘레: ${square.getPerimeter()}`);
console.log(`정사각형의 넓이: ${square.getArea()}`);
문제
출력
디버그 콘솔
Code
[Running] node "/Users/seeker/javascript/study.js"
정사각형의 둘레: 40
정사각형의 넓이: 100
[Done] exited with code=0 in 0.104 seconds
줄 39, 열 1
공백: 4
UTF-8
LF
JavaScript
포트: 3000
Prettier

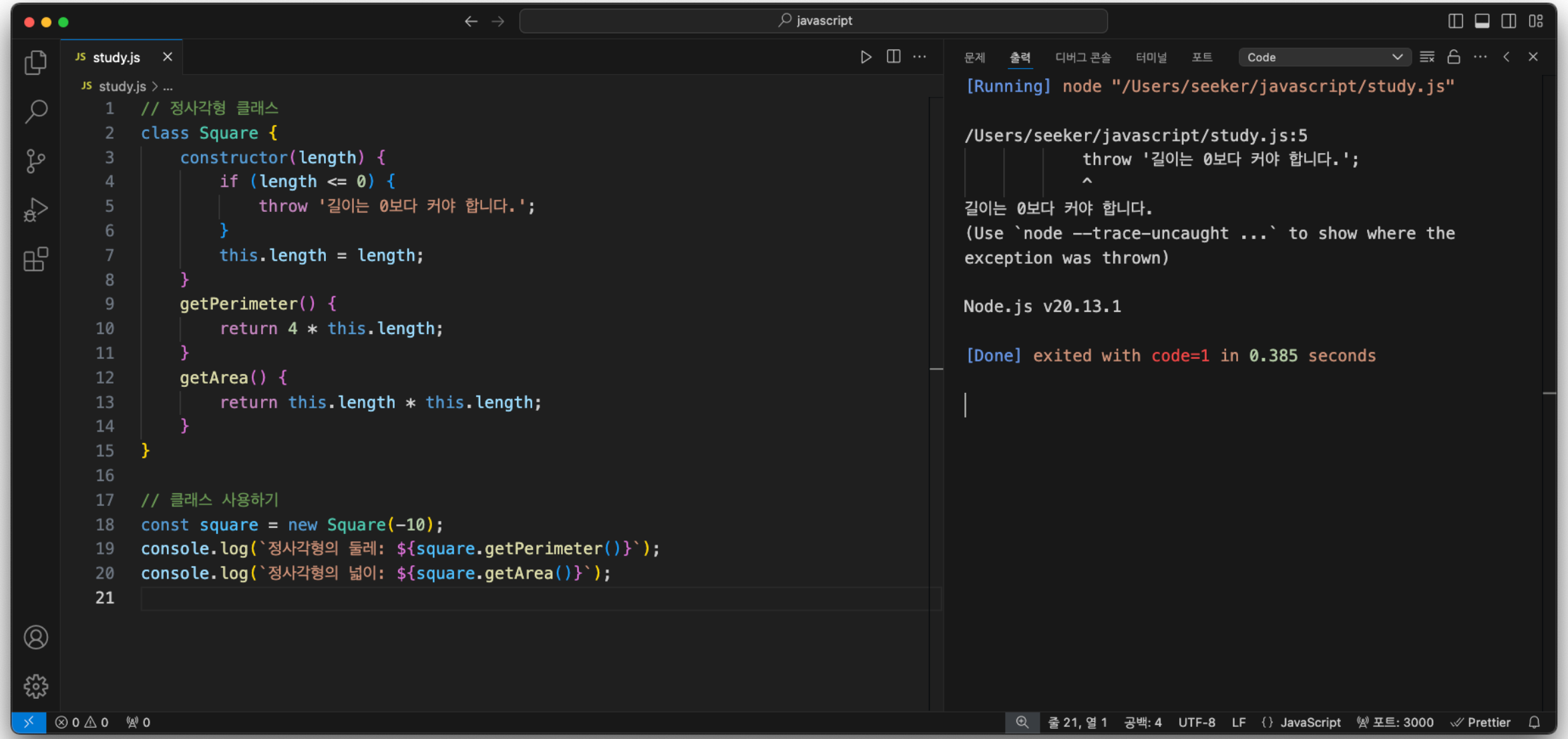
chap 9 상속


javascript
study.js
study.js > Square
// 사각형 클래스
class Rectangle {
constructor(width, height) {
this.width = width;
this.height = height;
}
// 사각형의 둘레를 구하는 메소드
getPerimeter() {
return 2 * (this.width + this.height);
}
// 사각형의 넓이를 구하는 메소드
getArea() {
return this.width * this.height;
}
}
// 정사각형 클래스
class Square extends Rectangle {
constructor(length) {
super(length, length);
}
}
// 클래스 사용하기
const square = new Square(10, 20);
console.log(`정사각형의 둘레: ${square.getPerimeter()}`);
console.log(`정사각형의 넓이: ${square.getArea()}`);
문제
출력
디버그 콘솔
터미널
포트
Code
[Running] node "/Users/seeker/javascript/study.js"
정사각형의 둘레: 40
정사각형의 넓이: 100
[Done] exited with code=0 in 0.225 seconds
줄 24, 열 2
공백: 4
UTF-8
LF
JavaScript
포트: 3000
Prettier

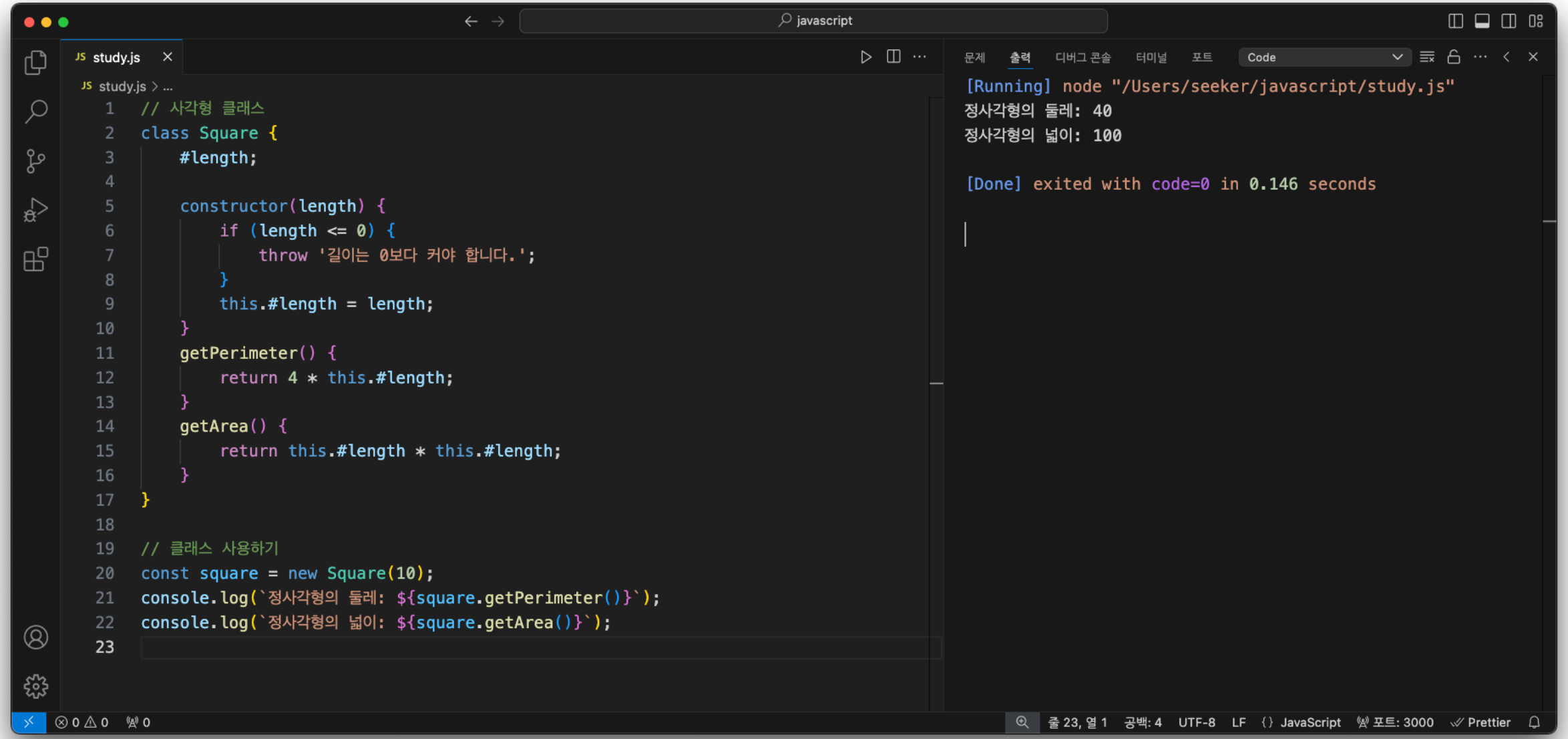
chap 9 private 속성과 메소드



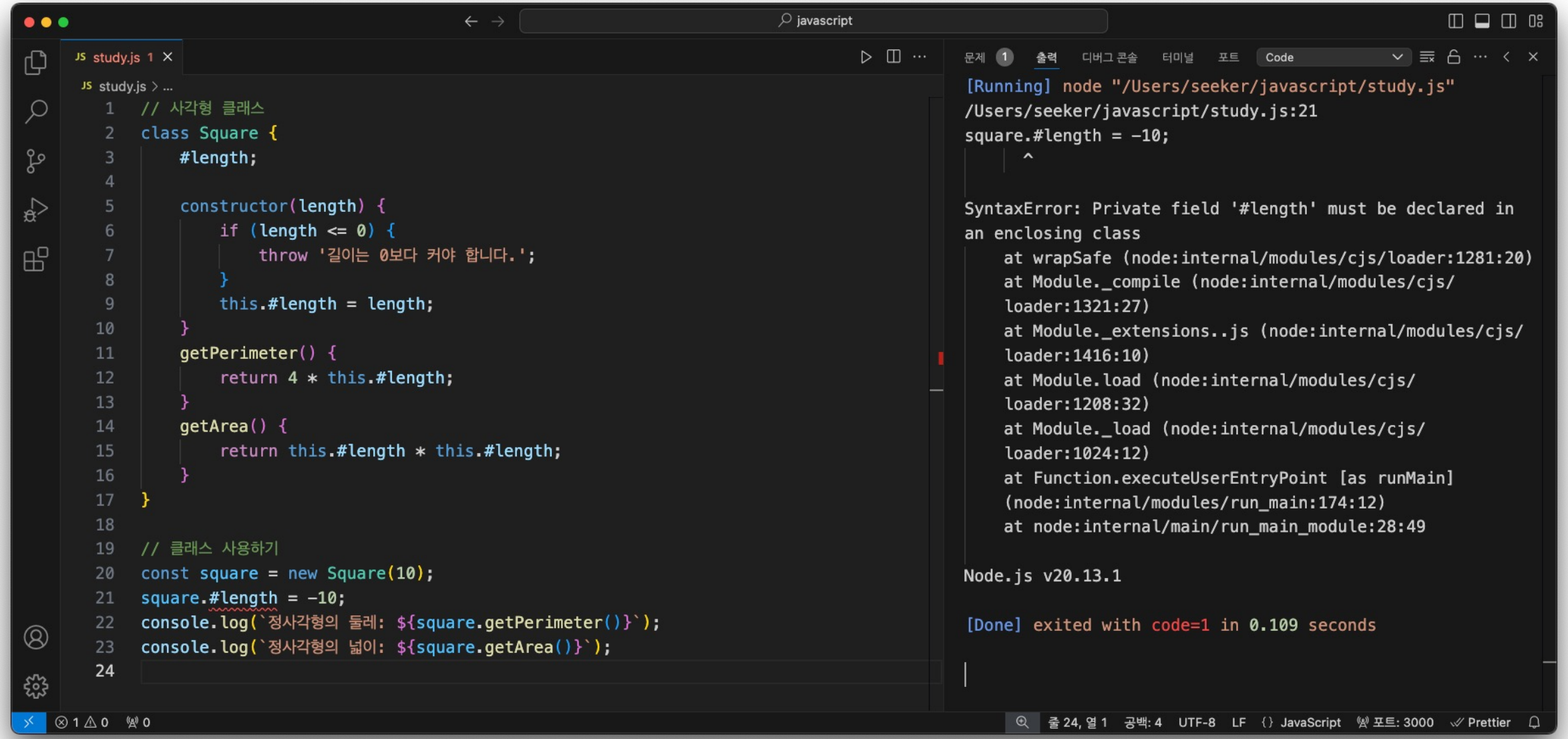
chap 9 private 속성과 메소드



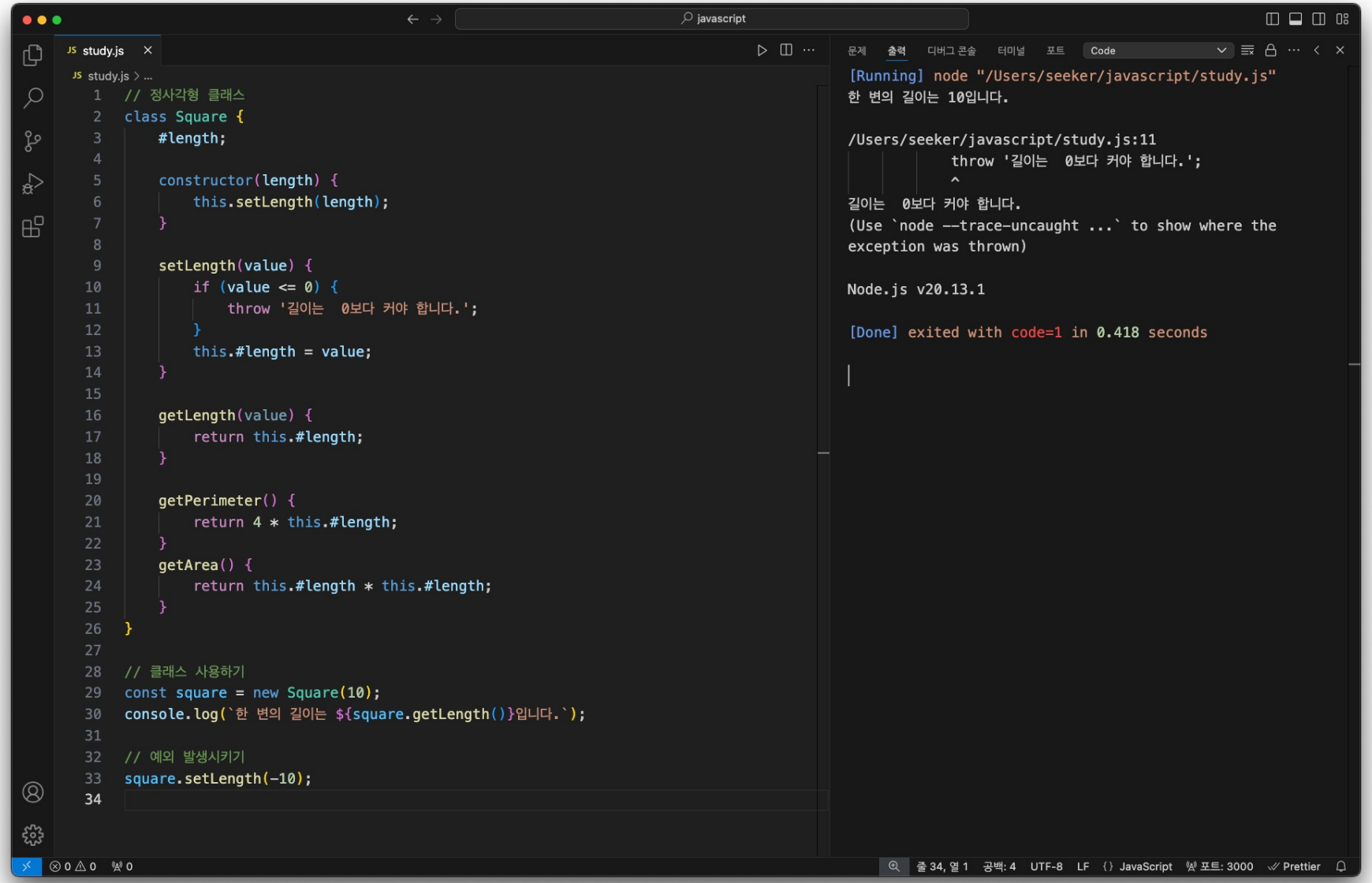
chap 9 private 속성과 메소드



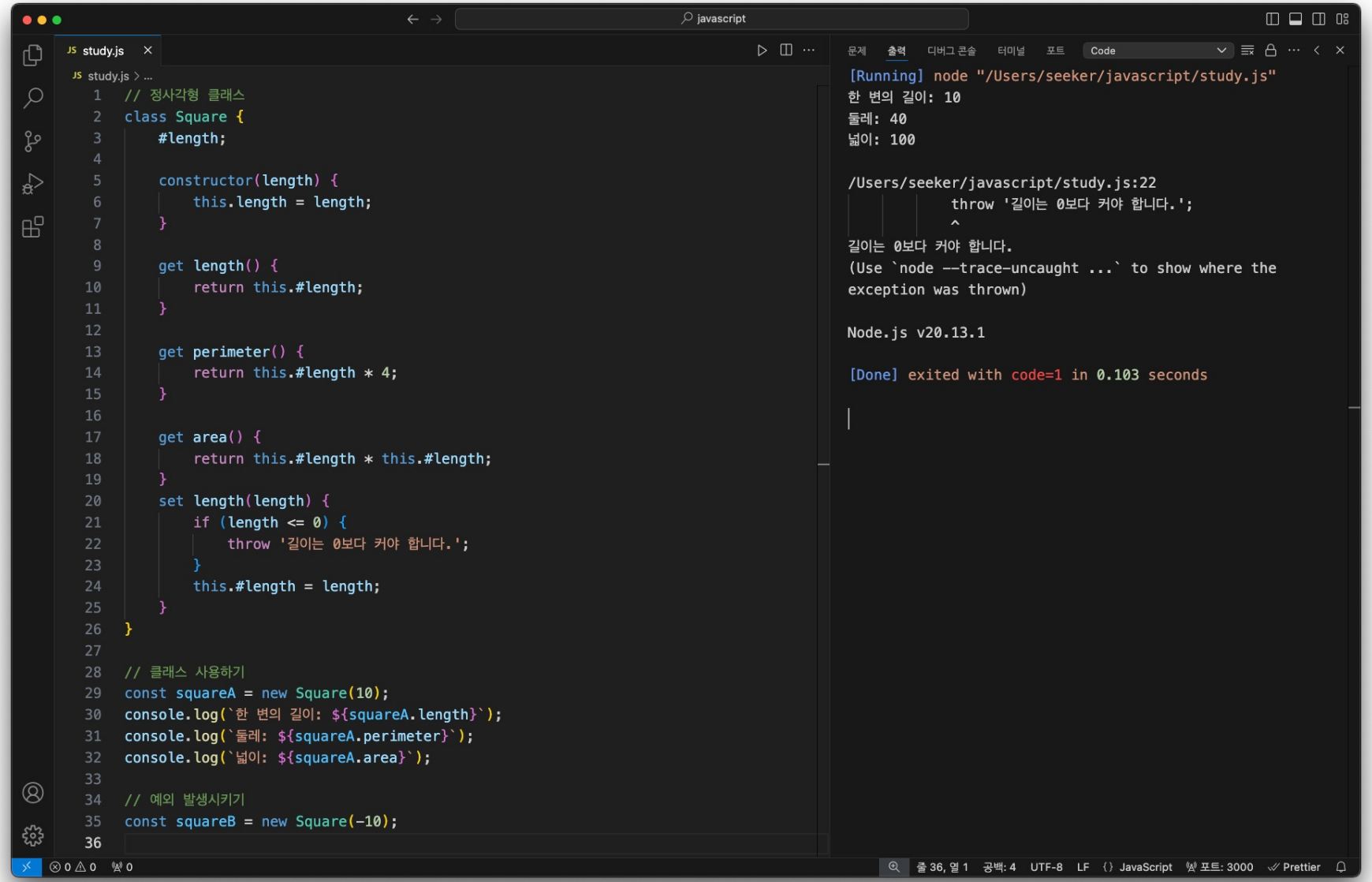
chap 9 private 속성과 메소드



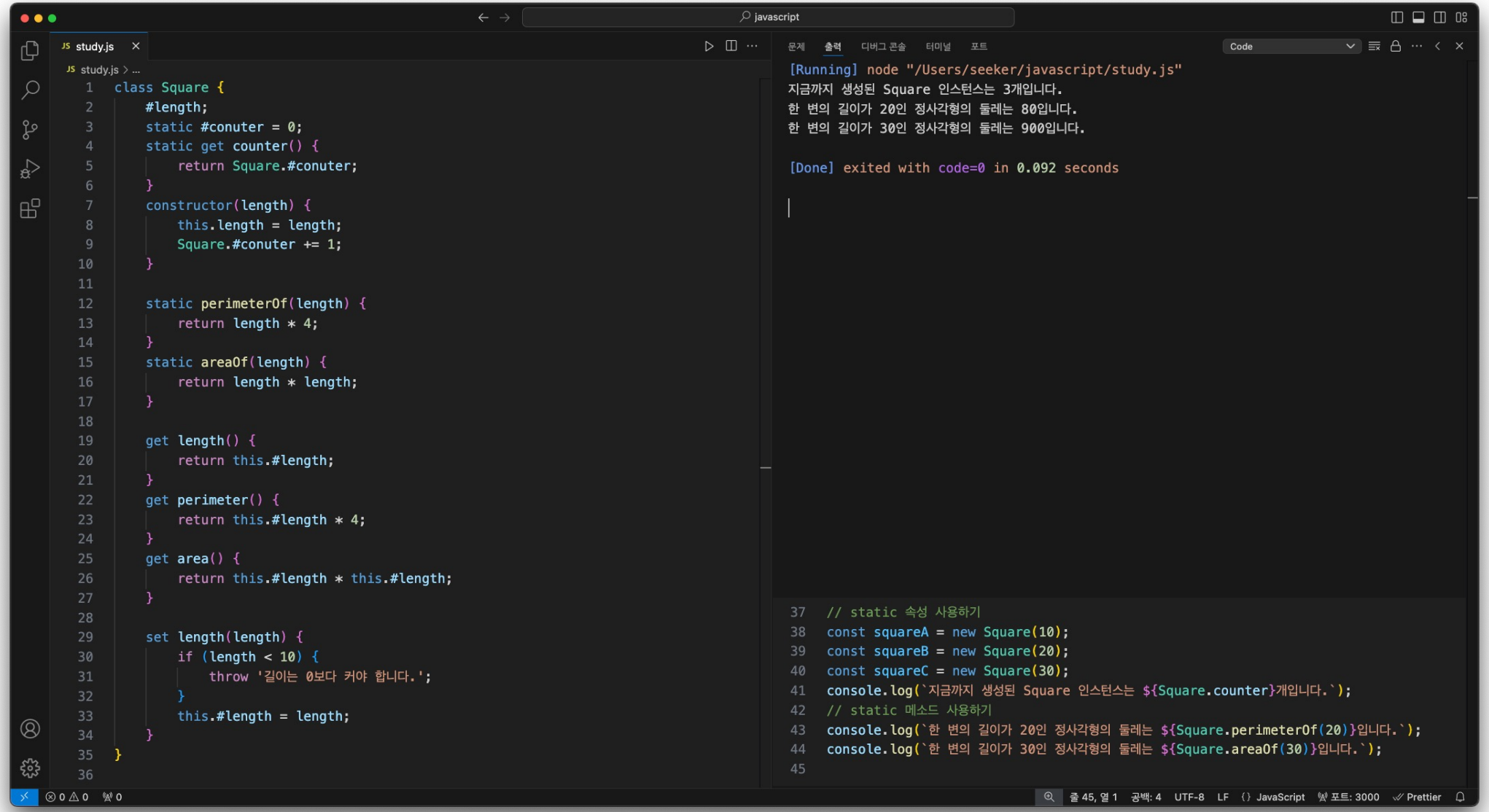
chap 9 getter와 setter



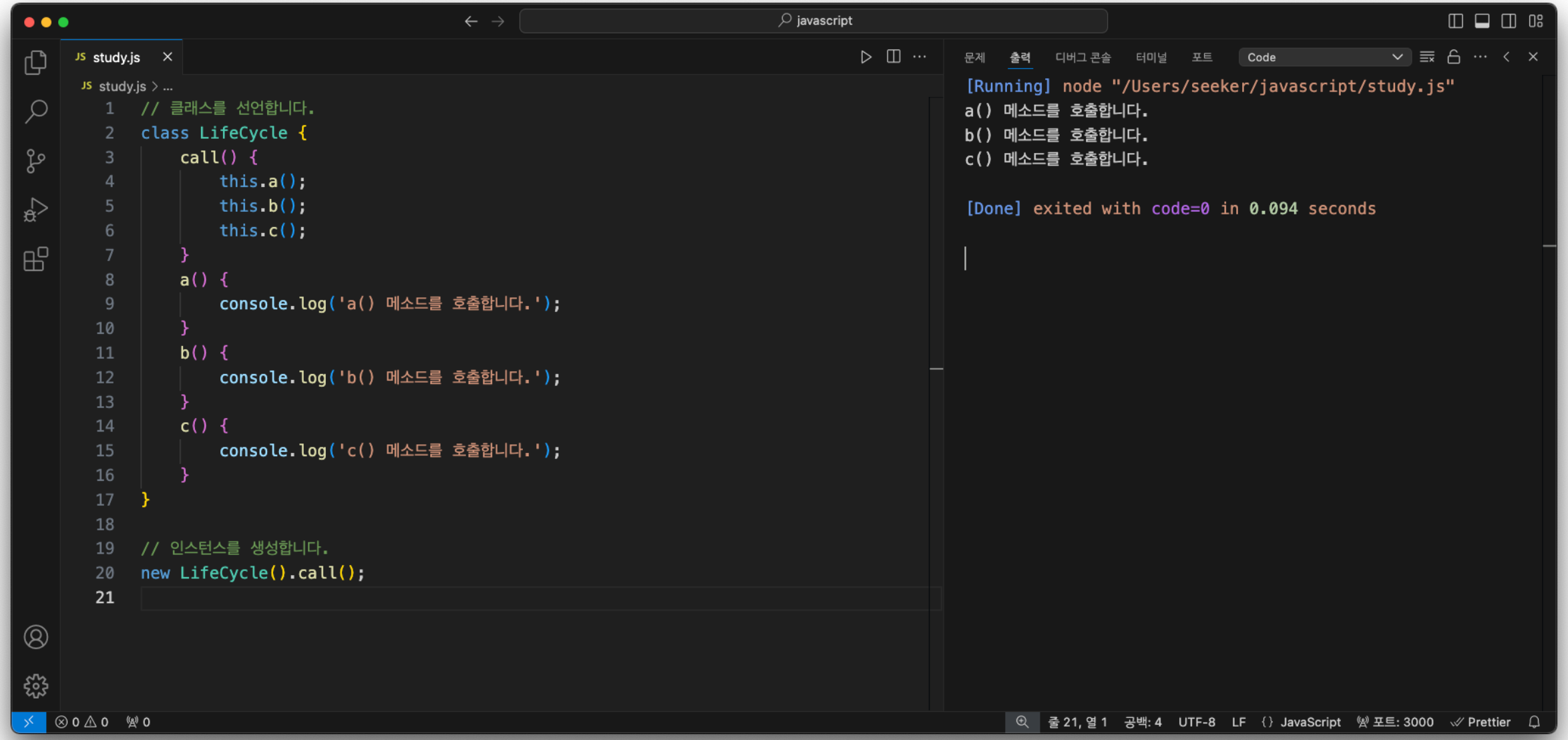
chap 9 getter와 setter



chap 9 static 속성과 메소드

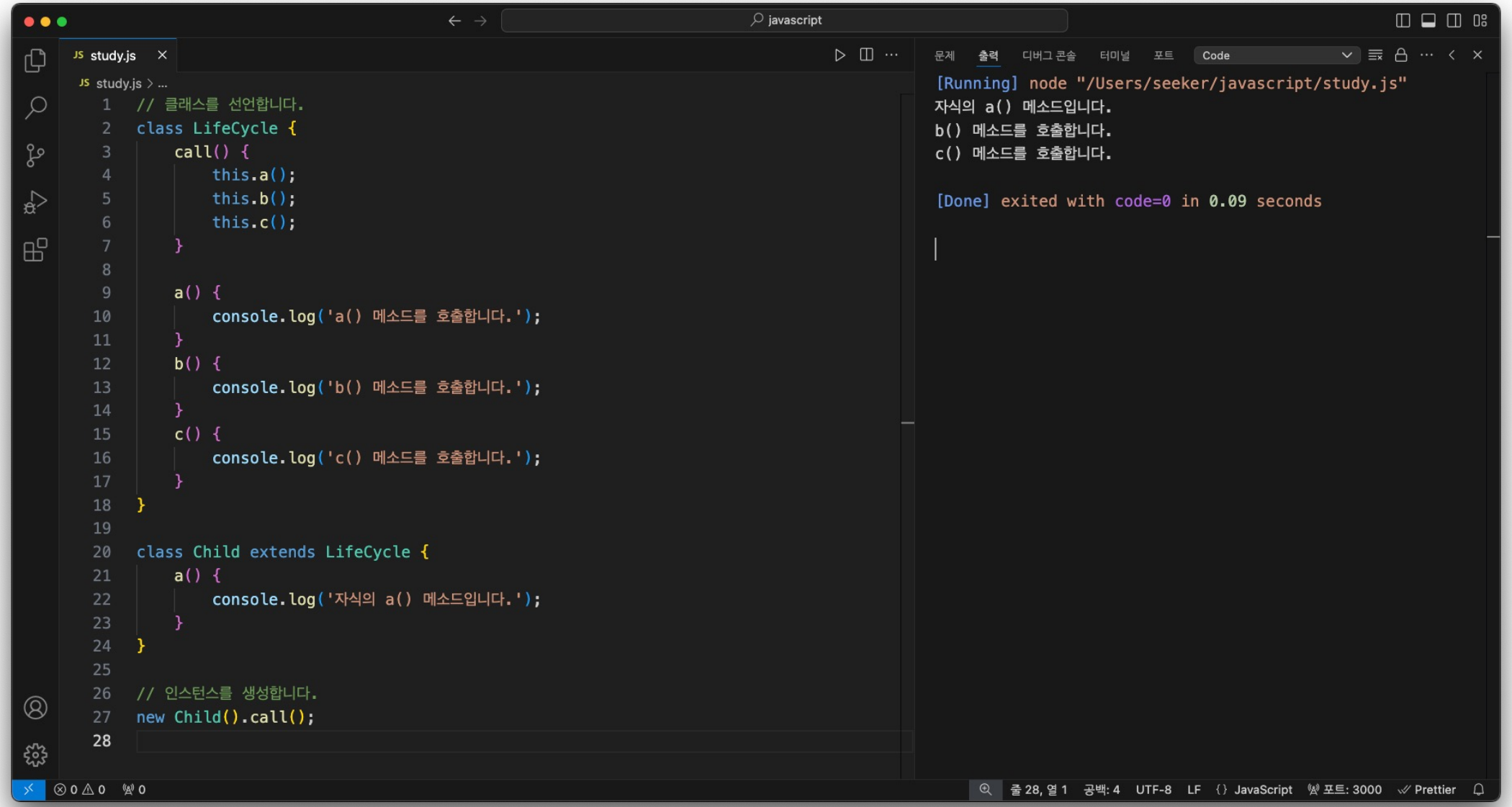


chap 9 오버라이드


javascript
study.js
study.js > ...
1 // 클래스를 선언합니다.
2 class LifeCycle {
3 call() {
4 this.a();
5 this.b();
6 this.c();
7 }
8 a() {
9 console.log('a() 메소드를 호출합니다.');
10 }
11 b() {
12 console.log('b() 메소드를 호출합니다.');
13 }
14 c() {
15 console.log('c() 메소드를 호출합니다.');
16 }
17 }
18
19 // 인스턴스를 생성합니다.
20 new LifeCycle().call();
21
문제 출력 디버그 콘솔 터미널 포트 Code
[Running] node "/Users/seeker/javascript/study.js"
a() 메소드를 호출합니다.
b() 메소드를 호출합니다.
c() 메소드를 호출합니다.
[Done] exited with code=0 in 0.094 seconds
⊗ 0 ⚠ 0 0
줄 21, 열 1 공백: 4 UTF-8 LF { } JavaScript 포트: 3000 Prettier

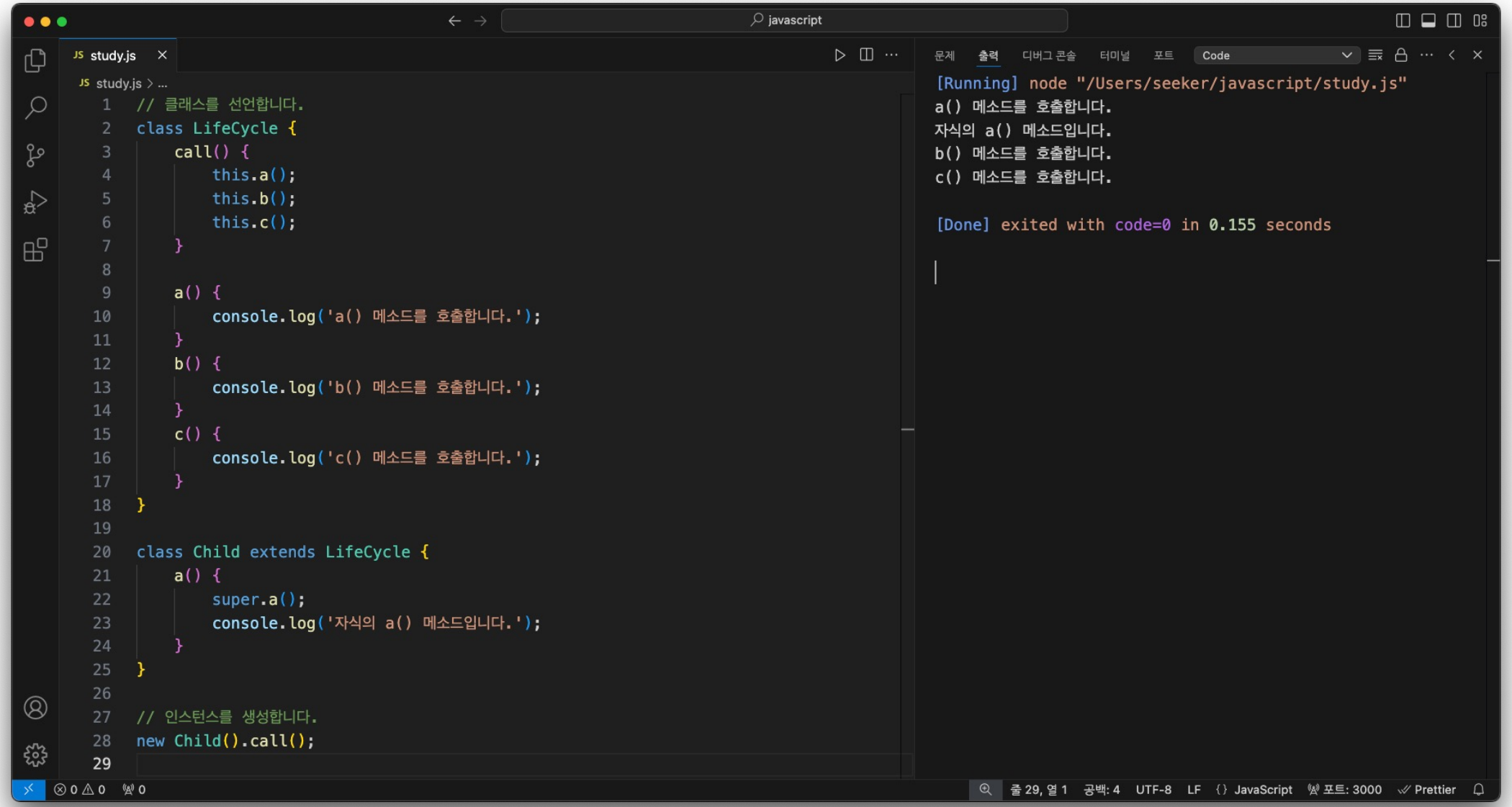
chap 9

오버라이드


javascript
study.js
study.js > ...
// 클래스를 선언합니다.
class LifeCycle {
call() {
this.a();
this.b();
this.c();
}
a() {
console.log('a() 메소드를 호출합니다.');
}
b() {
console.log('b() 메소드를 호출합니다.');
}
c() {
console.log('c() 메소드를 호출합니다.');
}
}
class Child extends LifeCycle {
a() {
console.log('자식의 a() 메소드입니다.');
}
}
// 인스턴스를 생성합니다.
new Child().call();
문제
출력
디버그 콘솔
터미널
포트
Code
[Running] node "/Users/seeker/javascript/study.js"
자식의 a() 메소드입니다.
b() 메소드를 호출합니다.
c() 메소드를 호출합니다.
[Done] exited with code=0 in 0.09 seconds
줄 28, 열 1
공백: 4
UTF-8
LF
JavaScript
포트: 3000
Prettier

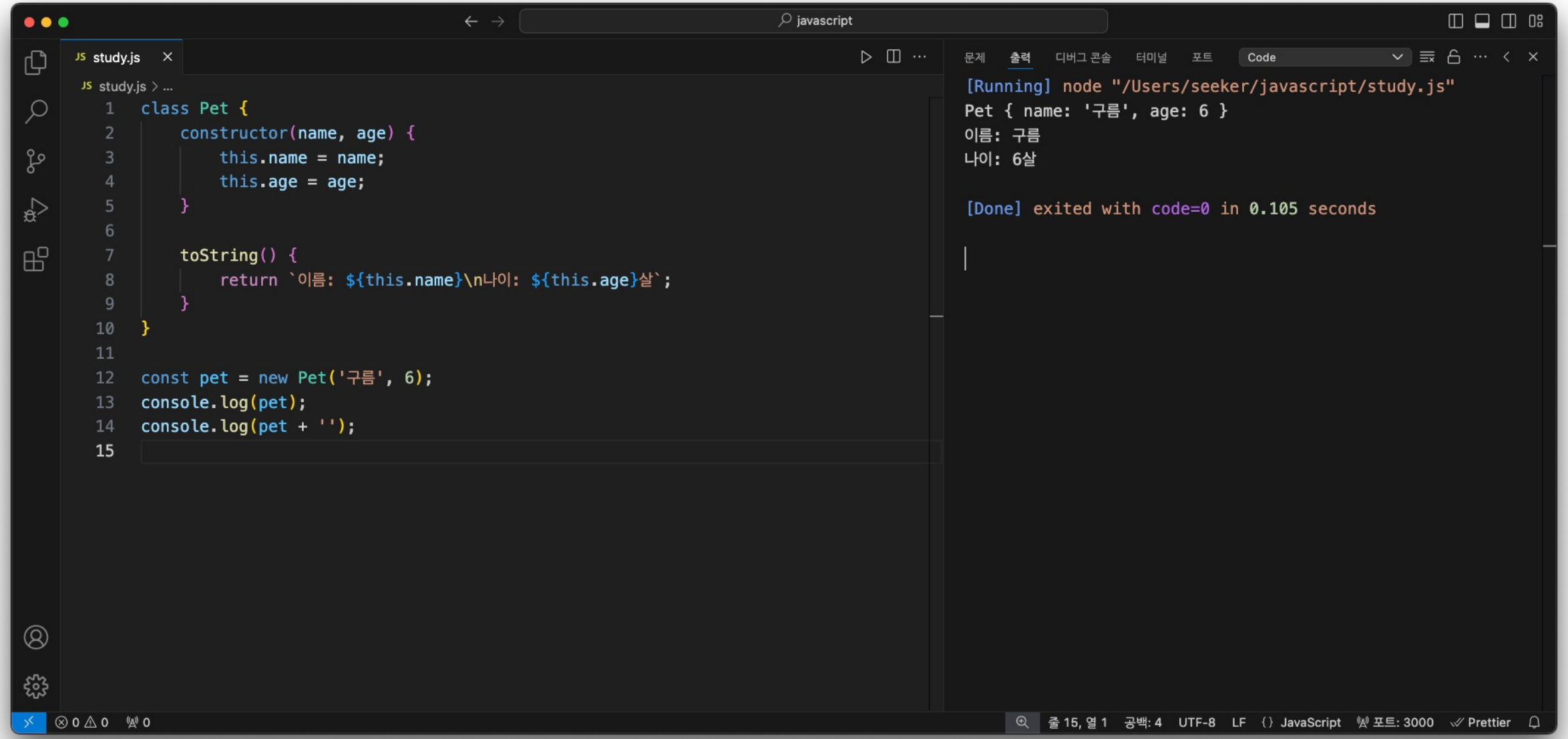
chap 9

오버라이드


javascript
study.js
study.js > ...
1 // 클래스를 선언합니다.
2 class LifeCycle {
3 call() {
4 this.a();
5 this.b();
6 this.c();
7 }
8
9 a() {
10 console.log('a() 메소드를 호출합니다.');
11 }
12 b() {
13 console.log('b() 메소드를 호출합니다.');
14 }
15 c() {
16 console.log('c() 메소드를 호출합니다.');
17 }
18 }
19
20 class Child extends LifeCycle {
21 a() {
22 super.a();
23 console.log('자식의 a() 메소드입니다.');
24 }
25 }
26
27 // 인스턴스를 생성합니다.
28 new Child().call();
29
문제 출력 디버그 콘솔 터미널 포트 Code
[Running] node "/Users/seeker/javascript/study.js"
a() 메소드를 호출합니다.
자식의 a() 메소드입니다.
b() 메소드를 호출합니다.
c() 메소드를 호출합니다.
[Done] exited with code=0 in 0.155 seconds
⊗ 0 ⚠ 0 0
줄 29, 열 1 공백: 4 UTF-8 LF {} JavaScript 포트: 3000 Prettier

chap 9

오버라이드 > 예제



감사합니다