

02. 변수와 상수

변수와 상수에 대해서 알아보시다. 변수와 상수는, 특정 이름에 특정 값을 담을 때 사용합니다.

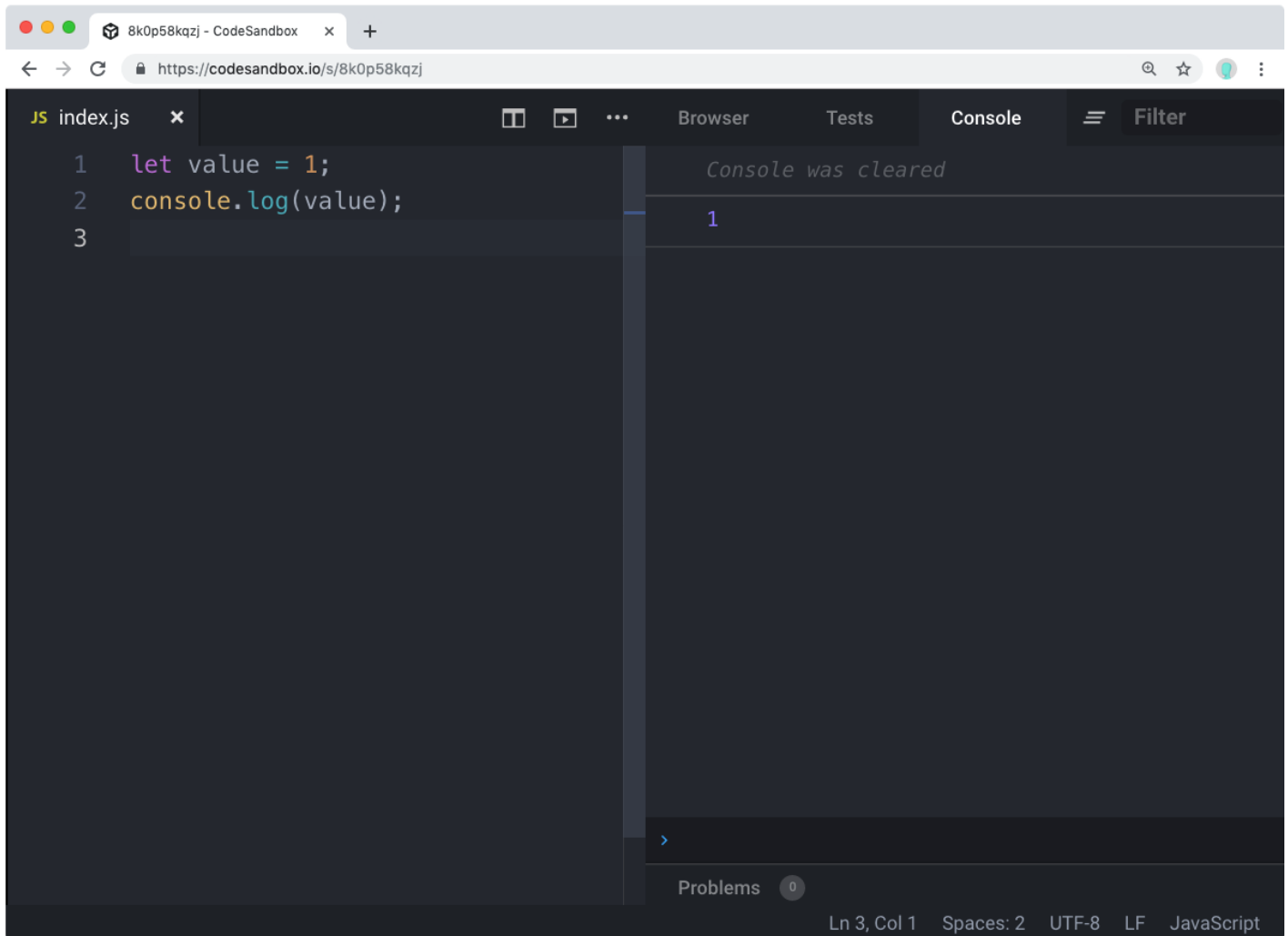
예를 들어서 우리가 `value` 라는 이름에 1 이라는 값을 넣는다고 가정해봅시다.

그러면, 코드를 이렇게 입력하면 됩니다.

```
let value = 1;
```

그러면, 앞으로 우리가 `value` 를 조회하면 `value` 는 1을 가르키게 됩니다. 예를 들어서 우리가 이전에 배웠던 `console.log` 를 통하여 `value` 값을 출력하도록 해보세요.

```
let value = 1;  
console.log(value);
```



1이라는 값이 우측에 나타날 것입니다.

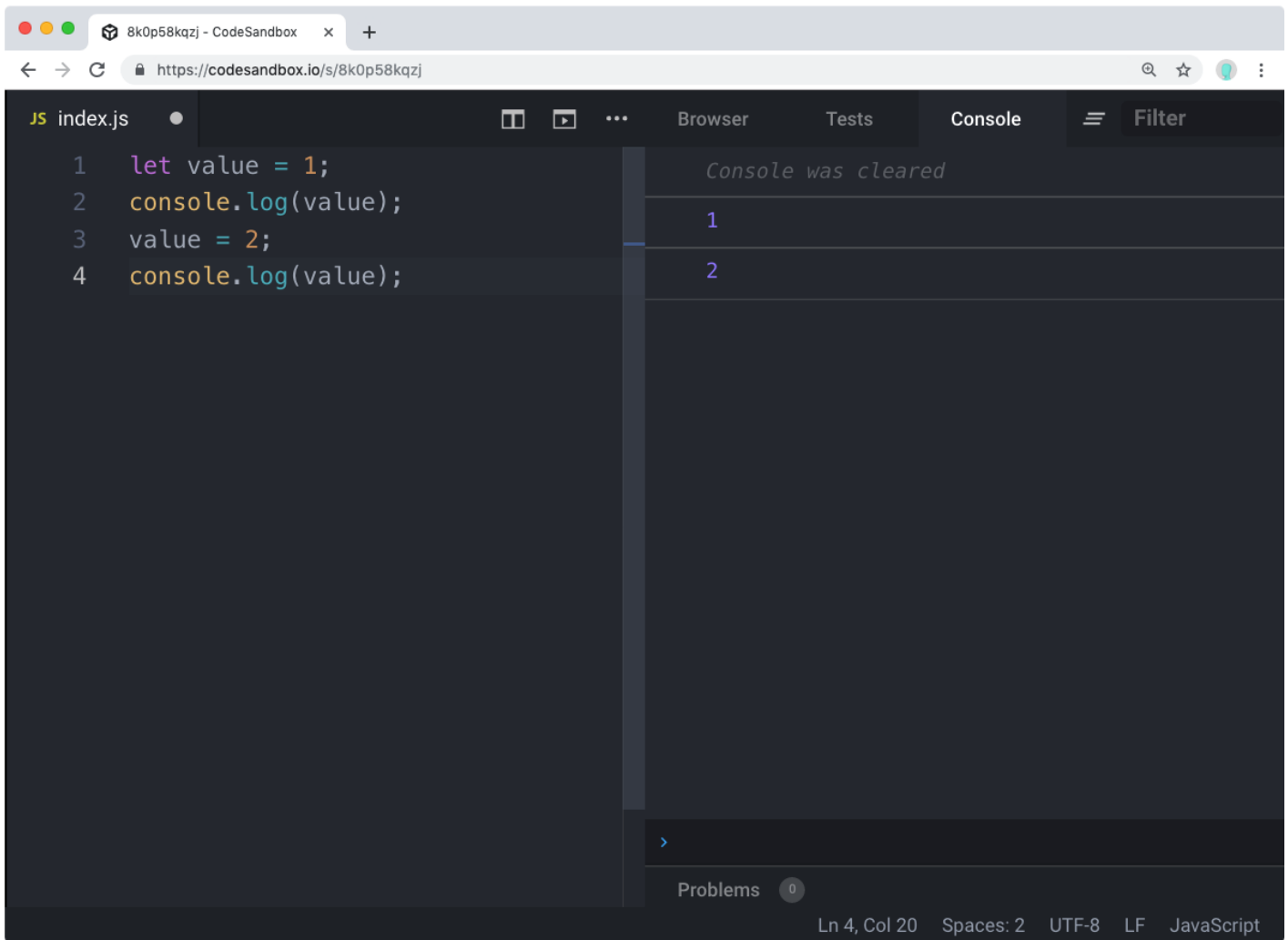
특정 이름에 특정 값을 설정하는 것. 우리는 이것을 **선언** 이라고 부릅니다. 쉽게 말하면 이제부터 `value` 는 1이야~ 라고 정해주는 것이죠.

값을 선언 할 때에는 두가지 종류가 있는데요, 하나는 변수이고, 하나는 상수입니다.

변수

변수는, 바뀔수 있는 값을 말합니다. 한번 값을 선언하고 나서 바꿀 수 있습니다.

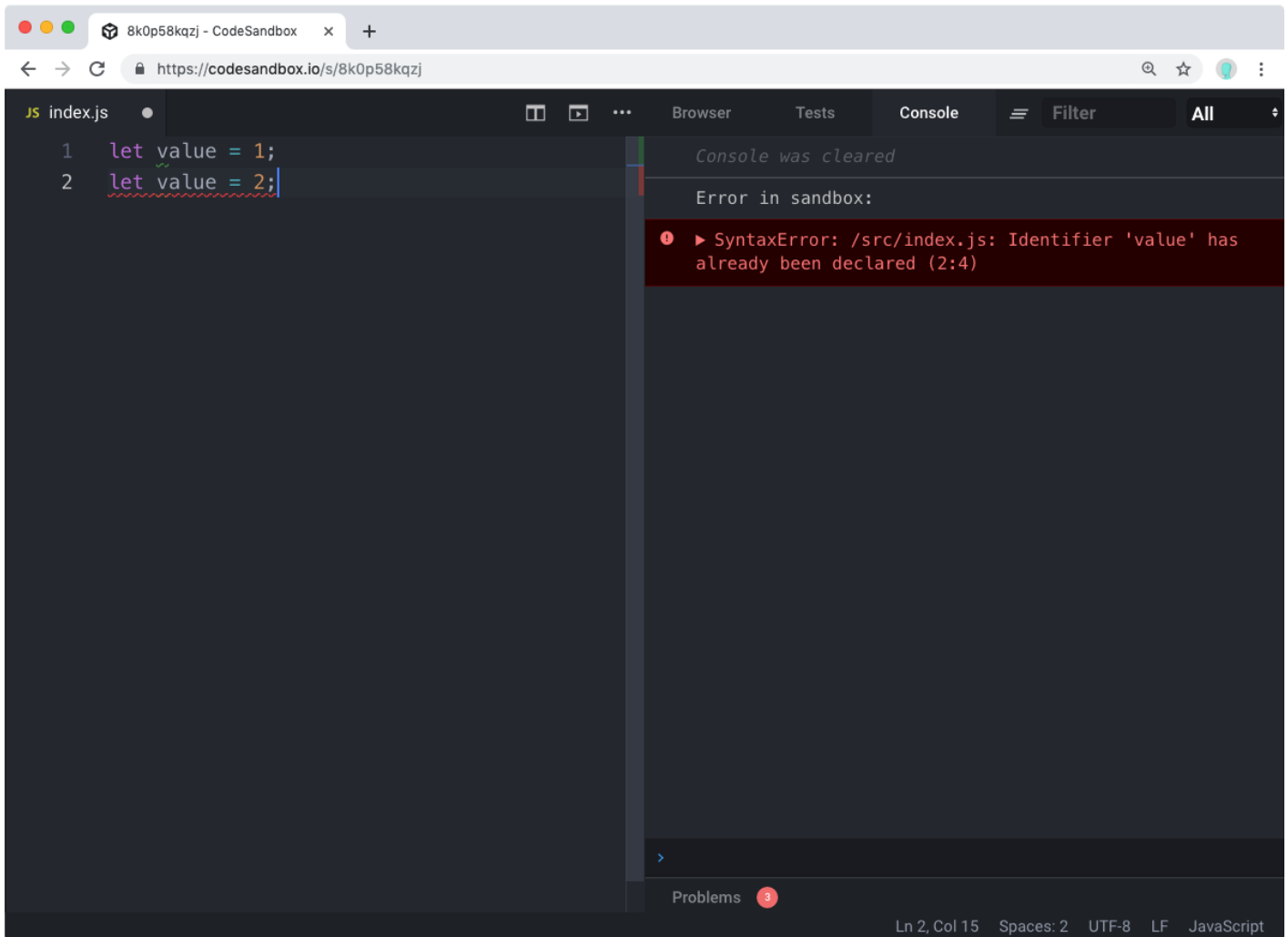
```
let value = 1;  
console.log(value);  
value = 2;  
console.log(value);
```



변수를 선언 할 때에는 이렇게 `let` 이라는 키워드를 사용합니다. 사용 하실 때 주의 할 점은 한번 선언했으면 똑같은 이름으로 선언하지 못합니다.

이런 코드는 오류가 발생합니다.

```
let value = 1;  
let value = 2;
```



단, 다른 블록 범위 내에서는 똑같은 이름으로 사용이 가능하긴 한데요, 이에 대해서는 나중에 알아보겠습니다.

상수

상수는, 한번 선언하고 값이 바뀌지 않는 값을 의미합니다. 즉, 값이 고정적이죠. 상수를 선언 할 때에는 다음과 같이 선언합니다.

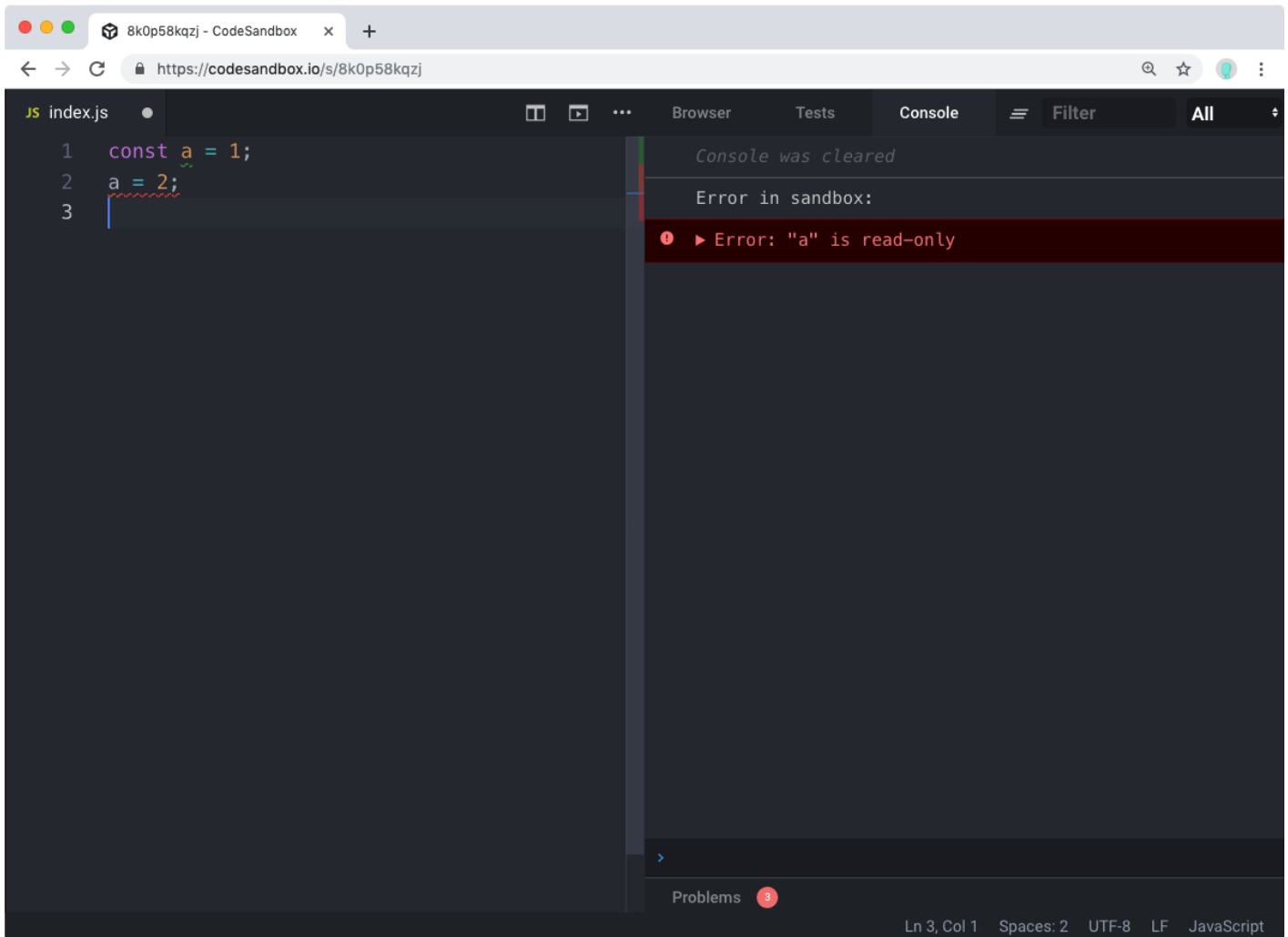
```
const a = 1;
```

이렇게, 상수를 선언 할 때에는 `const` 키워드를 사용합니다.

상수를 선언하고 나면, 값을 바꿀 수 없습니다.

한번 다음 코드를 입력해보세요.

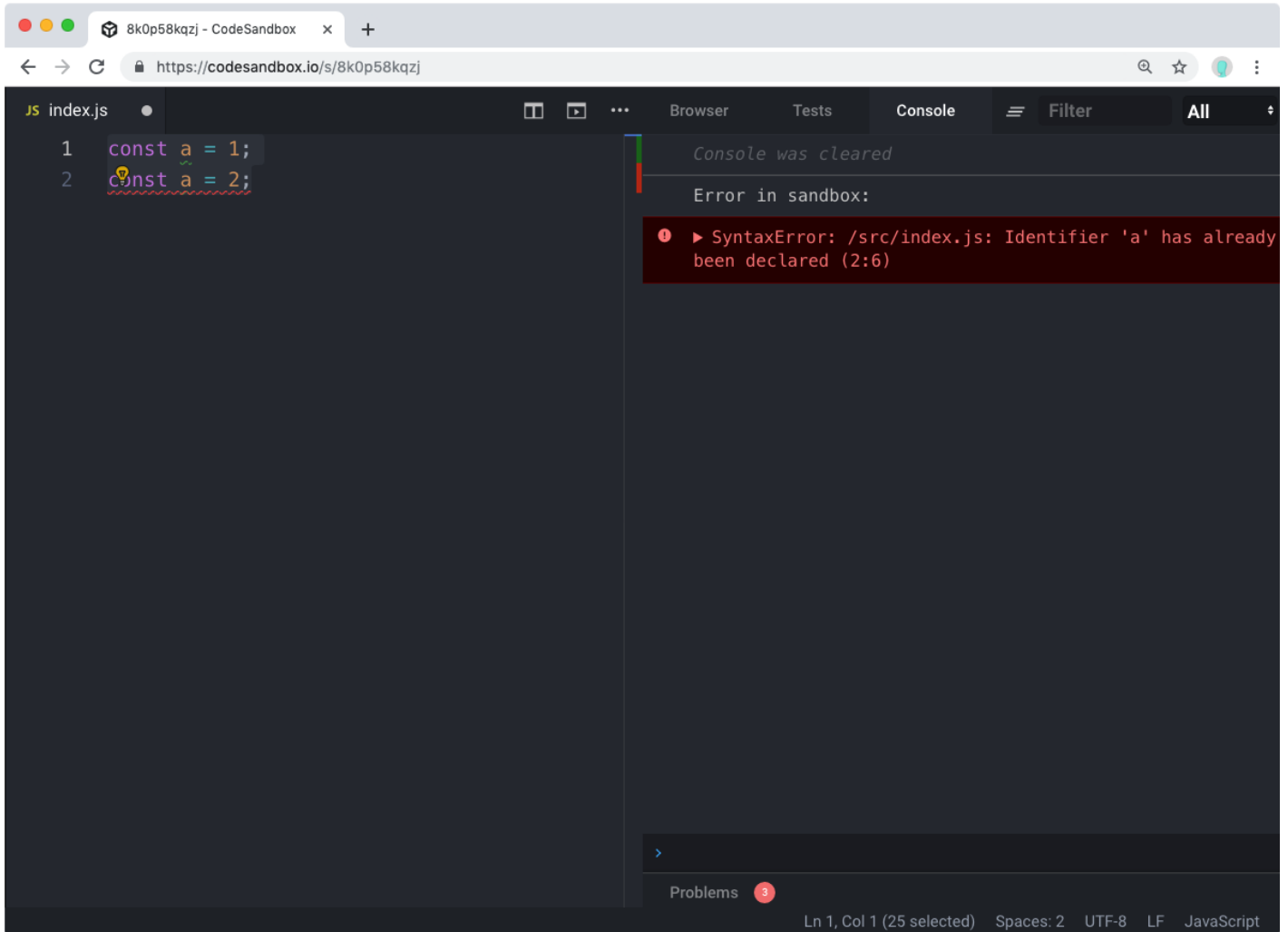
```
const a = 1;  
a = 2;
```



"Error: "a" is read-only" 라는 오류가 발생했습니다. 한번 상수로 선언했으면 값을 바꿀 수 없음을 의미합니다.

상수를 선언할 때에도 마찬가지로 한번 선언했으면 같은 이름으로 선언 할 수 없습니다.

```
const a = 1;  
const a = 2;
```



이제는 더 이상 사용하지마세요, var

변수를 선언하는 또 다른 방법으로, var 이라는 키워드가 있습니다. 이 키워드를 이미 알고 계신 분들도 있을텐데, 모오오오던 자바스크립트에서는 더 이상 사용하지 않습니다.

```
var a = 1;
```

var 이 let 과 다른 주요 차이점으로는, 똑같은 이름으로 여러번 선언 할 수도 있습니다. 추가적으로, var 과 let 은 사용 할 수 있는 범위가 다른데요, 이에 대해선 다음 번에 더 자세히 알아보도록 하겠습니다.

일단, var 키워드는 그냥 모르는걸로 하셔도 무방합니다.

추가적으로, IE9, IE10 같은 구형 브라우저에서는 let 과 const 를 사용 할 수 없습니다. 하지만, 보통 개발을 하게 될 때는 Babel 과 같은 도구를 사용하여 코드가 구형 브라우저에서도 돌아갈 수 있게끔 변환작업을 합니다. 만약에, 여러분이 나중에 별도의 도구 없이 구형 브라우저를 호환시켜야 하는 상황이 온다면 (그럴 일은 거의 없을 겁니다.) var 를 사용하게 될 수도 있습니다.

데이터 타입

우리가 변수나 상수를 선언하게 될 때, 숫자 외에도 다른 값들을 선언 할 수 있습니다. 종류는 굉장히 많은데요 그 중에서 가장 기본적인 것들을 알아보겠습니다.

숫자 (Number)

우선, 이미 사용해봤지만, 숫자는 그냥 바로 값을 대입하면 됩니다.

```
let value = 1;
```

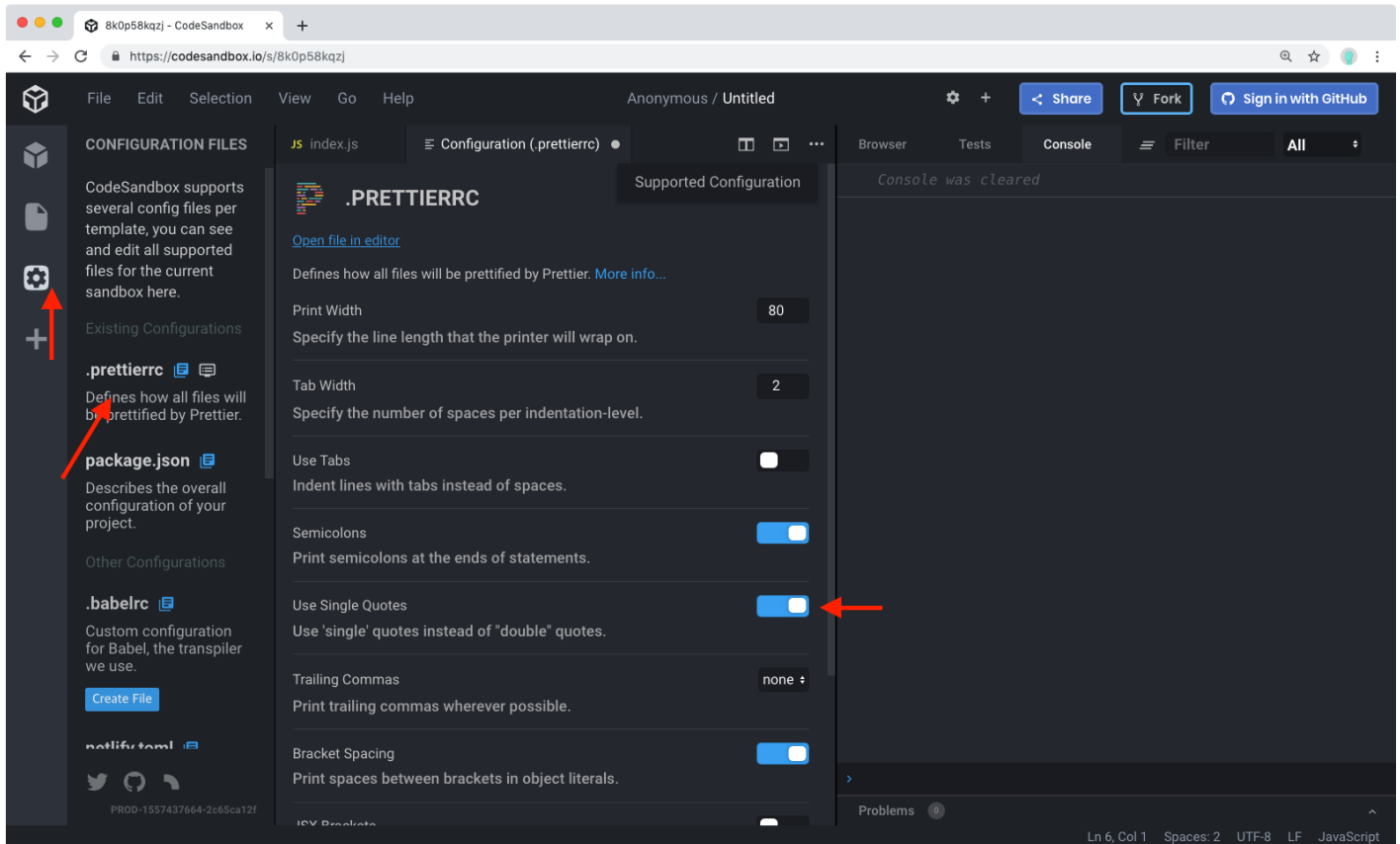
문자열 (String)

그리고, 텍스트 (주로, 프로그래밍 언어에서는 이를 문자열이라고 부릅니다.) 형태의 값은 작은 따옴표 혹은 큰 따옴표로 감싸서 선언합니다.

```
let text = 'hello';
let name = '좌박스크립트';
```

작은 따옴표와 큰 따옴표 사용에 있어서 큰 차이는 없습니다. 둘다 사용하셔도 되는데, 하나만 선택하셔서 사용하시면 됩니다. 저는 개인적으로 작은 따옴표 사용을 선호합니다.

CodeSandbox 에서는 작은 따옴표로 작성을 하면 자동으로 큰 따옴표로 변환을 해줄 것입니다. 자동 변환되는 것을 방지하기 위해서는 CodeSandbox의 좌측의 설정 아이콘을 누르고, .prettierrc 의 Create File 버튼을 누르고 나서 Use Single Quotes 를 활성화하세요.



참/거짓 (Boolean)

이번에는 boolean 이라는 것에 대해서 알아보겠습니다. boolean 은, 참 혹은 거짓 두가지 종류의 값을 나타낼 수 있습니다.

```
let good = true;
let loading = false;
```

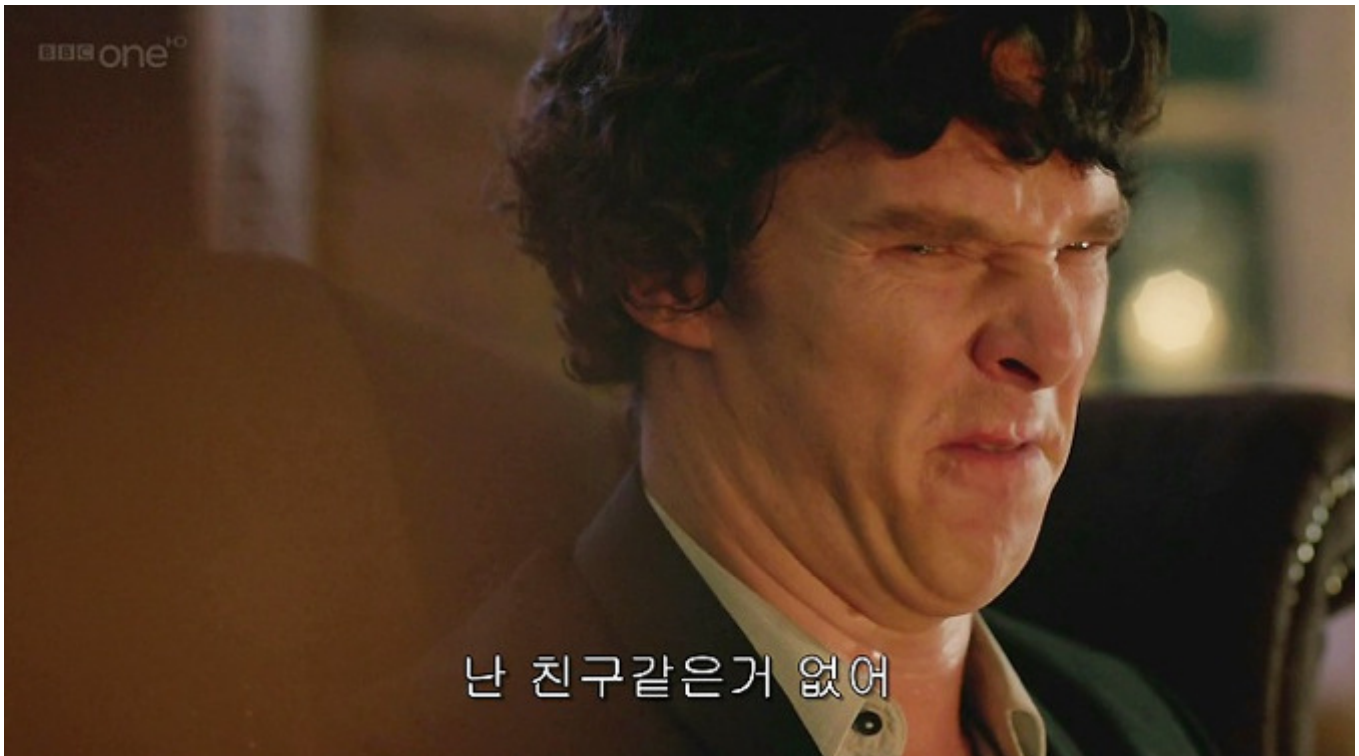
참은 true, 거짓은 false 입니다.

null 과 undefined

자바스크립트에서는 "없음" 을 의미하는 데이터 타입이 두 종류가 있는데요, 하나는 null 이고 하나는 undefined 인데, 둘의 용도가 살짝 다릅니다.

null 은 주로, 이 값이 없다! 라고 선언을 할 때 사용합니다.

영국 드라마 주인공 셜록의 드라마 대사 중에 "난 친구 같은거 없어" 라는 대사가 있습니다.



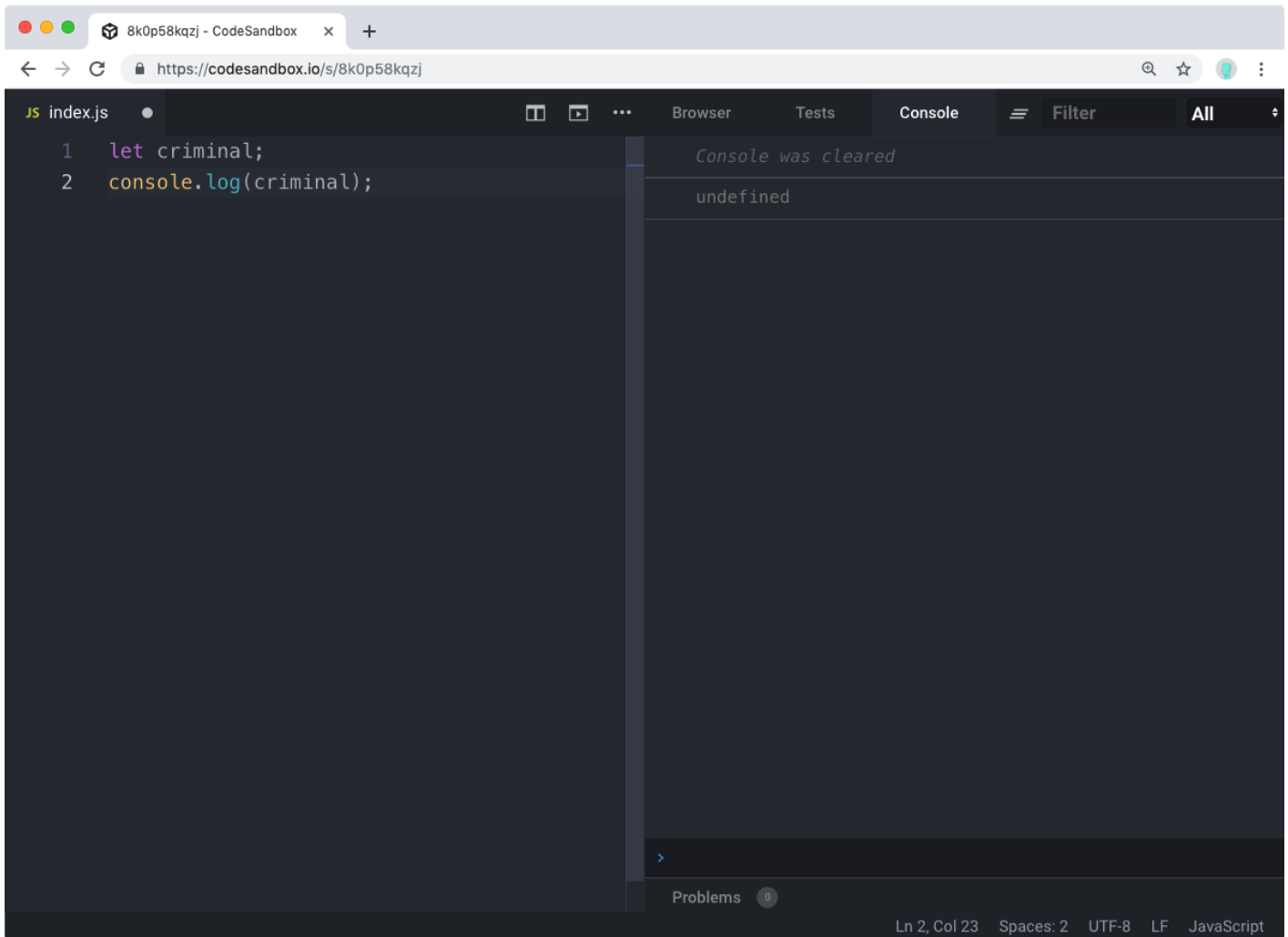
```
const friend = null;
```

반면, `undefined` 는, 아직 값이 설정되지 않은 것을 의미합니다.

다음 코드를 입력해보세요.

```
let criminal;  
console.log(criminal);
```

`criminal` 이라는 변수를 선언하긴 했지만 값을 지정해주지는 않았습니다. 이를 `console.log` 를 통해 보여주도록 하면 `undefined` 라는 값이 나타나게 됩니다.



```
JS index.js
1 let criminal;
2 console.log(criminal);
```

Console was cleared

undefined

Problems 0

Ln 2, Col 23 Spaces: 2 UTF-8 LF JavaScript

`null` 과 `undefined` 는 둘 다 값이 없음을 의미하는건 맞는데, `null` 은 우리가 없다고, 고의적으로 설정하는 값을 의미하고, `undefined` 는 우리가 설정을 하지 않았기 때문에 없는 값을 의미합니다