



Lexical Environment는 뭘 뜻하는 걸까?

변수나 함수등의 실백자를 정의하는데 사용되는 객체

아래 코드를 실행해 보시고 왜 저렇게 나오는지 설명하시오

foo()함수를 사용하여 x가
10으로 선언되지만 아마
bar라는 함수는 전역으로
설정되어 x를
출력하는것이다
아마 순서는 이렇게
될것이다
x(지역)=1
x(전역)=1
Bar()x(전역)을 출력
Bar()x(전역)을 출력

```
var x = 1;
function foo() {
  var x = 10;
  bar();
}
function bar() {
  console.log(x);
}
foo();
bar();
```

1

1

undefined



아래 콘솔로그에서 뭐가 나올까? 돌려봅시다.

```
var x = 'globaaaaal';  
function banana(){  
  console.log(x); // 여기선 뭐가 나올까??  
  var x = 'loooooooooooooooooooooooooocal';  
}  
banana();  
console.log(x);  
  
undefined  
  
globaaaaal  
  
undefined
```



전역 객체란? 무엇인가?

객체는 객체인데 전역으로 잡힌 객체다.

전역이다 보니 다른 객체보다도 먼저
생성될 것이다.



아래 키워드로 사용 억제법을 한번 찾아보세요

1. 즉시 실행 함수

함수를 선언하자마자 즉시 실행되는 형태로 함수 내부의 변수들이 전역 스코프에 노출되는 것을 방지할 수 있다

2. 네임스페이스 객체

객체를 생성하고 해당 객체에 메서드를 추가하는 방법

3. 모듈 패턴

변수와 함수들을 실행함수로 감싸 하나의 모듈을 만들어 다른 코드와 격리시키는 방법이다

4. ES6 모듈

각모듈이 독립된 스코프를 가지며 모듈 내에서 선언한 변수와 함수는 모듈내에서만 접근이 가능



전역변수 + 지역변수 관련된 해석을 본인이 직접 최소 30줄 이상의 코드를 작성 해서 서술해보세요

```
var height = 174.3;
```

```
function oneYearAgoHeight() {  
    var agoHeight = 174;  
    console.log('지역변수 1년전키' + agoHeight)  
}
```

```
oneYearAgoHeight();  
console.log('전역변수 내 현재키' + height);
```

```
var weight = 56.8;
```

```
function oneYearAgoWeight() {  
    var agoWeight = 68;  
    console.log('1년전 몸무게: ' + agoWeight)  
}
```

```
oneYearAgoWeight();  
console.log('현재 몸무게: ' + weight)
```

```
var eyes = '왼쪽눈 0.2 오른쪽눈 0.8';
```

```
function wearGlasses() {  
    var eyes = '왼쪽눈 0.8 오른쪽눈 1.2'  
    console.log(eyes)  
}
```

```
wearGlasses();  
console.log(eyes)
```

```
var icetea = '그냥 아이스티: 2200원';
```

```
function plusCoffee() {  
    var icetea_Coffee = '커피추가한 아이스티: 2700원';  
    console.log(icetea_Coffee)  
}
```

```
plusCoffee();  
console.log(icetea);
```



아래 콘솔로그의 결과값을 추론해보세요.

```
var x = 1;
var y = 1;

var x = 100;
var y;

console.log(x);
console.log(y);
```

100과 1



하단의 코드를 실행하고 에러가 난다면 왜 나는지 설명하시오.

```
let a = 1;

{
  let a = 2;
  let b = 3;
}

console.log(a);
console.log(b);
```

B는 지역으로 설정되어있기에 전역을
출력하려고하는 b는 출력할수 없음
전역으로 설정된게 먼저 실행되기에



지금까지 했던 코드를 기반으로 `let`, `const` 키워드를 적절히 사용해서 `var` 키워드 대신 써보세요.

```

const readline = require('readline');

const rl = readline.createInterface({
  input: process.stdin,
  output : process.stdout
});

rl.question('', (M) => { // M을 입력받음
  rl.question('', (N) => { // N을 입력받음
    // 입력값을 정수로 변환
    M = parseInt(M);
    N = parseInt(N);

    let sum = 0; // 소수들의 합
    let min;    // 최소 소수
    let yes = 1; // 소수의 여부
    let count = 0; // 소수의 개수 카운트

    for (let i = M; i <= N; i++) {
      if (i === 1) continue; // 1은 소수가 아님
      else if (i === 2) { // 2는 소수이므로 초기값 설정
        yes = 0;
        count++;
        min = 2;
      }
      else {
        for (let j = i - 1; j > 1; j--) {
          yes = 0;
          if (i % j === 0) {
            yes = 1;
            break;
          }
        }
      }
      if (count === 0) {
        min = i;
      }
      if (yes === 0) {
        sum += i;
        count++;
      }
    }
    if (count === 0) {
      console.log("-1");
    }
    else {
      console.log(`${sum}
${ min }`);
    }
  });
});

rl.close();
});

```