

let (block-scoped) vs var (블록 범위 무시, 가급적 사용 자제)

```
let x = 1;

if (true) {
  let x = 2; // 상위 블록과 다른 변수
  console.log(x); // 2
}
console.log(x); // 1
```

```
var x = 1;

if (true) {
  var x = 2; //상위 블록과 같은 변수
  console.log(x); // 2
}
console.log(x); // 2
```

const (상수 선언, **immutable**, Java의 final)

```
> const x = 20;
> const x = 10; // 에러
```

사용 권고 :

1. **const**
2. **let**
3. **var 사용 금지**

immutable 변수

- (순수) 함수형 프로그래밍에서는 immutable 변수를 사용함
 - 한번 주어진 값이 변경되지 않음
 - 기존 값을 이용하여 새로운 값을 만들어 저장할 때는 복사가 이루어짐
- side-effect 함수에서는 mutable 변수 사용 가능함
 - 기존 저장된 값을 덮어쓰는(overwrite) 방법 적용 가능

quicksort (Haskell)

```
qsort :: Ord a => [a] -> [a]
```

```
qsort [] = []
```

```
qsort (x:xs) = qsort smaller ++ [x] ++ qsort larger
```

```
  where
```

```
    lesser = [a | a <- xs, a <= x]
```

```
    greater = [b | b <- xs, b > x]
```

quicksort

