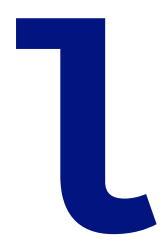
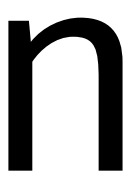
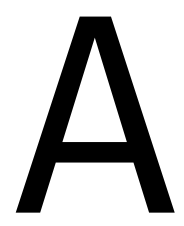
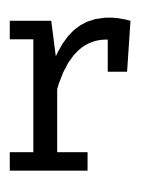
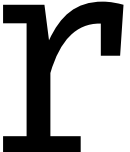
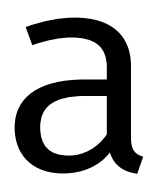


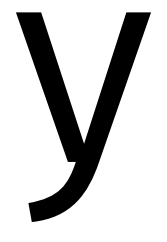


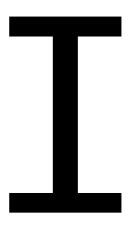

A

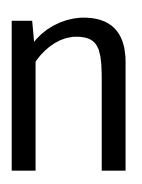


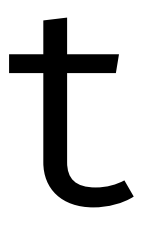


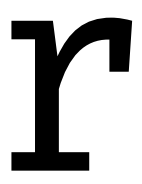


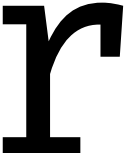


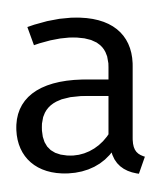


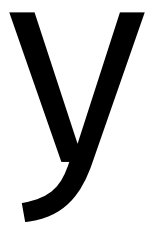


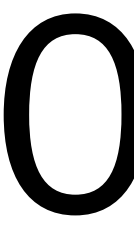


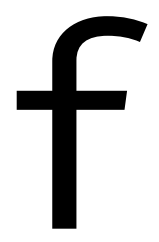




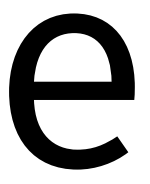


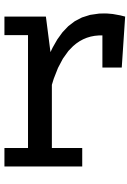


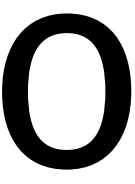


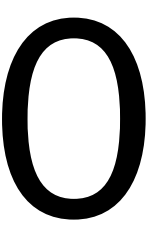


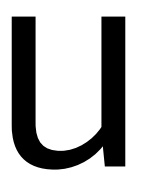

1

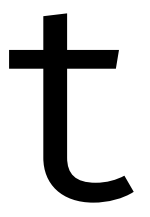


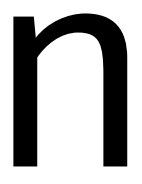


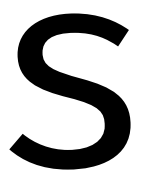






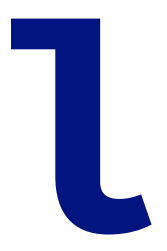






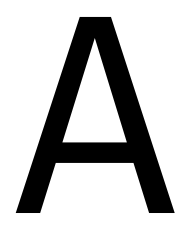
Generics

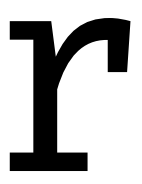
invariance

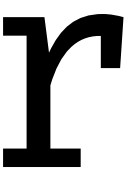


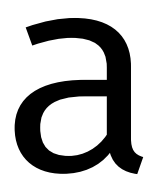

C

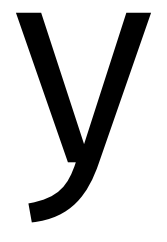
C


A



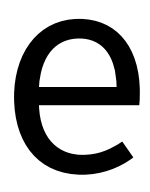


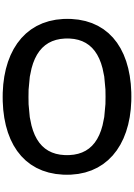


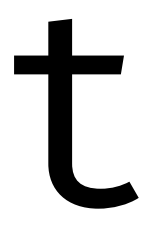


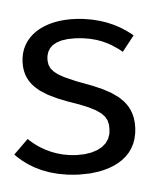
Type mismatch.
Required: Array<Number>
Found: Array<Int>

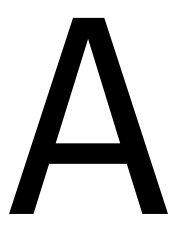
`Array<Int>` is not a subtype of `Array<Number>`
(`Array` is **invariant** in T)

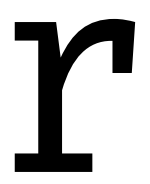


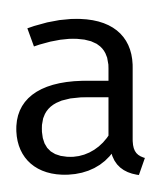


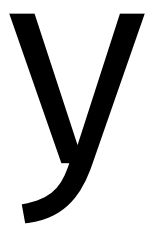


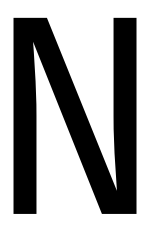


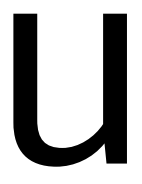

A

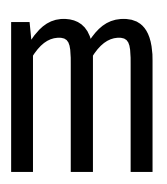


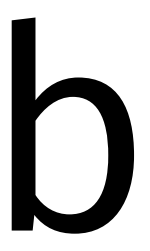


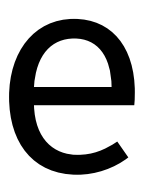


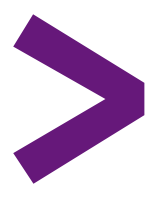


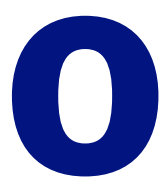


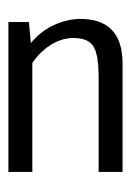


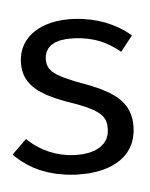

b



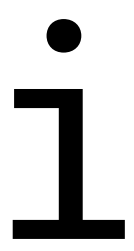


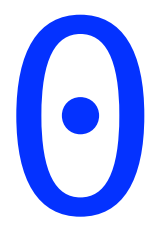


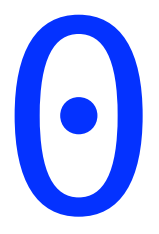




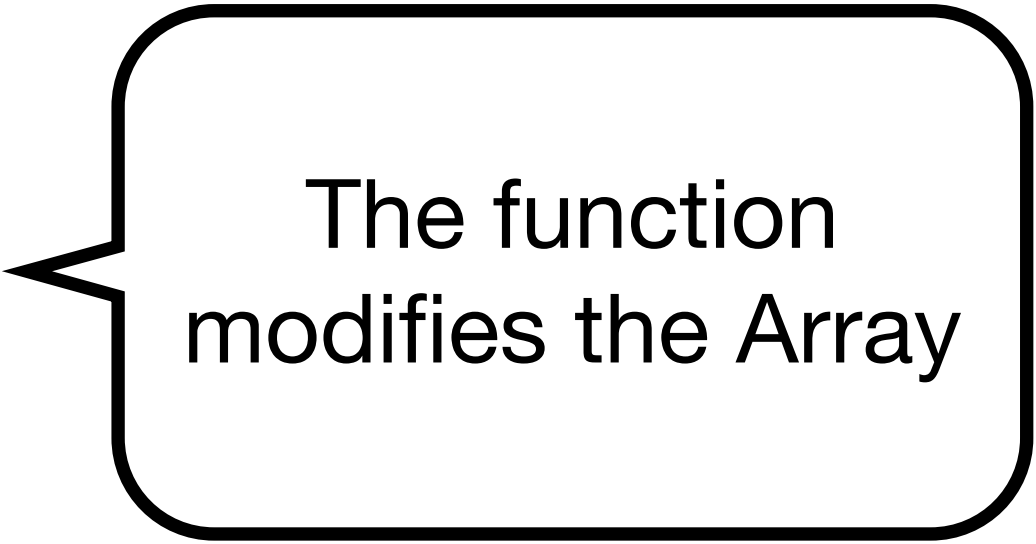

C








[0.0, 0.0, 0.0, 0.0]


The function
modifies the Array

```
fun zeroOut(ns: Array<Number>) {
    for (i in ns.indices) {
        ns[i] = 0.0
    }
}
```

```
val ns: Array<Int> = arrayOf(1, 2, 3, 4)

zeroOut(ns)
```

```
public class Array<T>
```

Generics

invariance

```
public class Array<T>

fun zeroOut(ns: Array<Number>) {
    for (i in ns.indices) {
        ns[i] = 0.0
    }
}

val ns: Array<Int> = arrayOf(1, 2, 3, 4)

zeroOut(ns)
```

[0.0, 0.0, 0.0, 0.0]

Type mismatch.
Required: Array<Number>
Found: Array<Int>

The function modifies the Array

`Array<Int>` is not a subtype of `Array<Number>`
(`Array` is **invariant** in T)

Generics

use site variance / projections

```
public class Array<T>
```