



**Generics**

**use site variance / projections**





















































































































**Type mismatch.**

**Required: Array<Any>**

**Found: Array<Int>**





















































































































public class Array<T>



The function just reads  
from the Array

```
fun joinInt(a:Array<Any>):String=a.joinToString{it.toString()}
```

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

```
val concat = joinInt(ns)
```

# Generics

## use site variance / projections

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

The function just reads  
from the Array

```
fun joinIt(a: Array<Any>): String = a.joinToString { it.toString() }
```

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

```
val concat = joinIt(ns)
```

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



# Generics

## use site variance / projections

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

The function just reads  
from the Array

```
fun joinIt(a: Array<out Any>): String = a.joinToString { it.toString() }
```

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

```
val concat = joinIt(ns)
```