Generics

V

a

ɩ

i

n

t

S

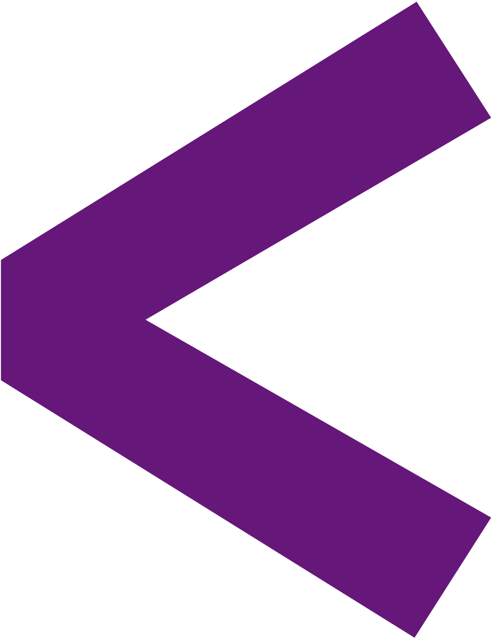
●

●

L

i

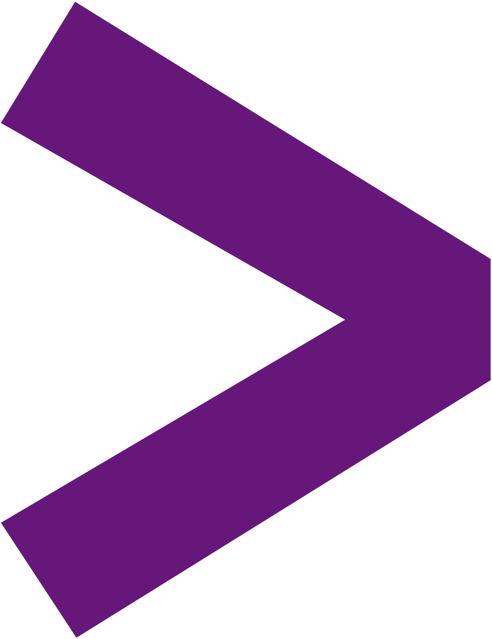
t



I

n

t



=

ɩ

i

S

t

0

f

1

,

2

,

3

,

4

,

5

,

6

V

a

l

t

0

D

O

u

b

l

e

●

●

N

u

m

b

e

r

[redacted]

>

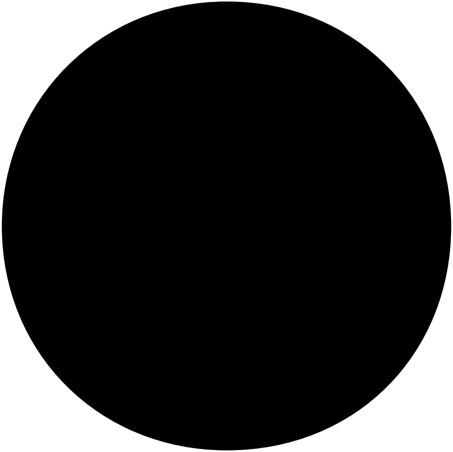
b

[redacted]

{

i

t



t

O

D

0

u

b

ɩ

e

}

n

u

m

b

e

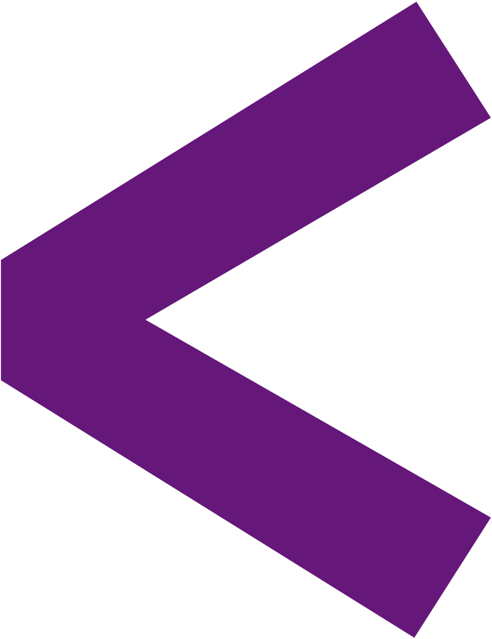
r

S

●

●

L



N

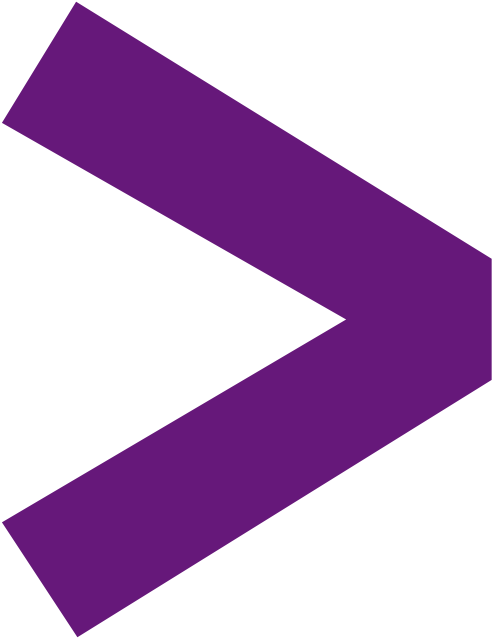
u

m

b

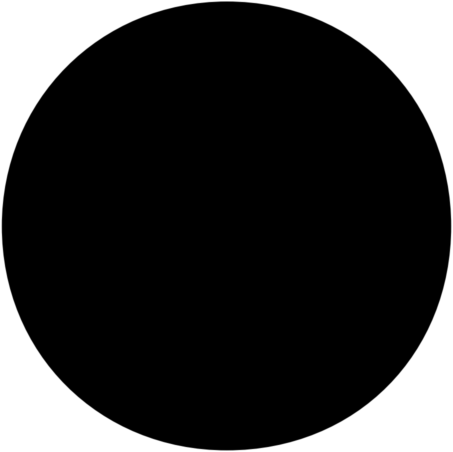
e

r



=

S



m

a

p

O

D

O

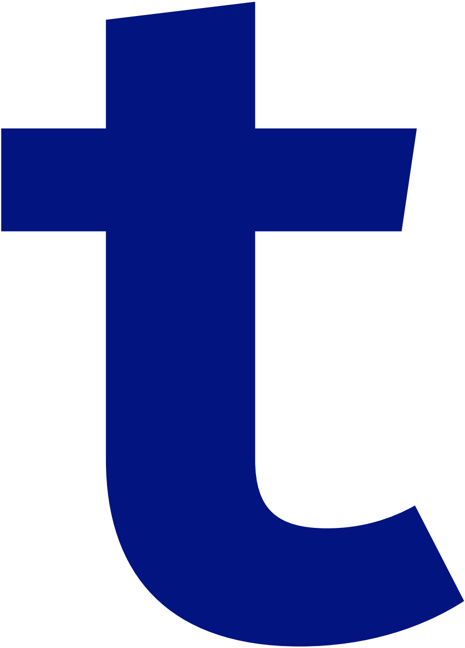
u

ɩ

e

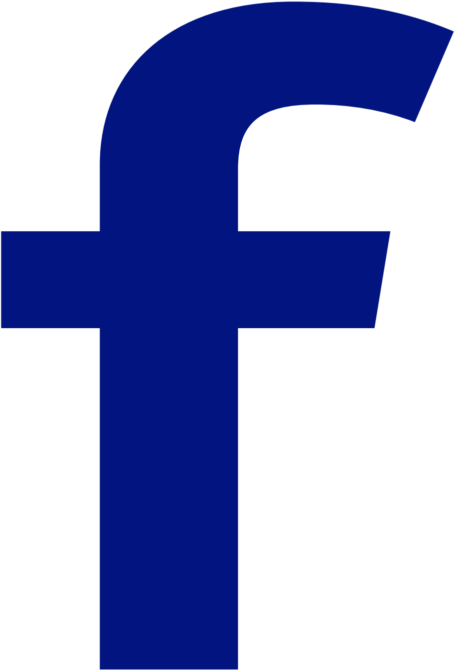
i

n


t

e

r



a

C

F

u

n

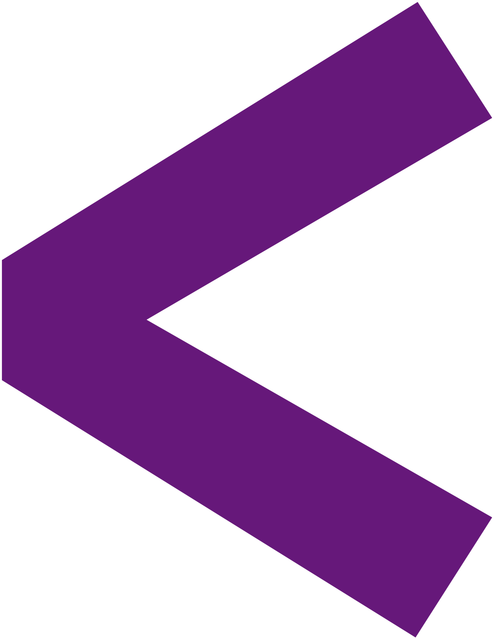
C

t

i

O

1



P

1

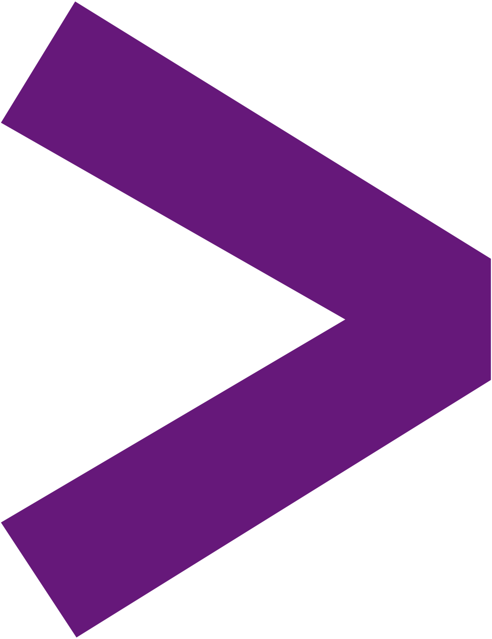
,

O

u

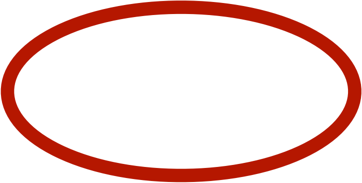
t

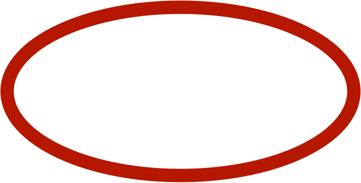
R



A function that takes a `Number` were a function that takes in `Int` is expected

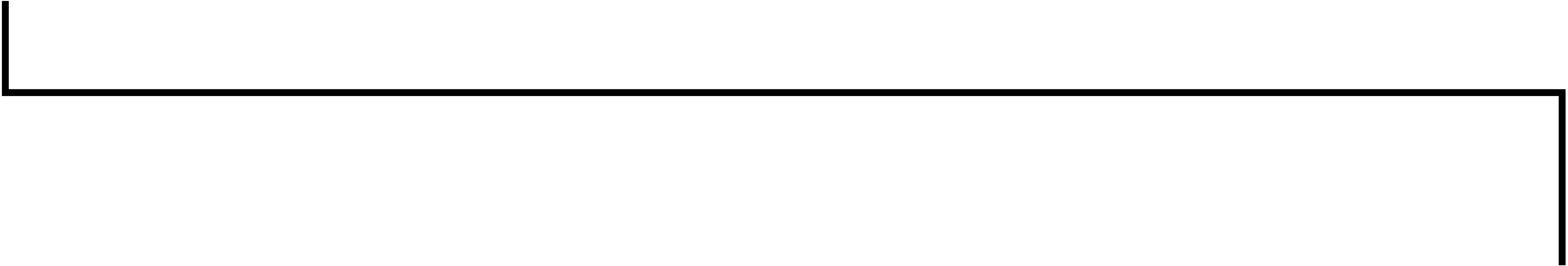
`List<Double>` is assigned to a `List<Number>` typed variable

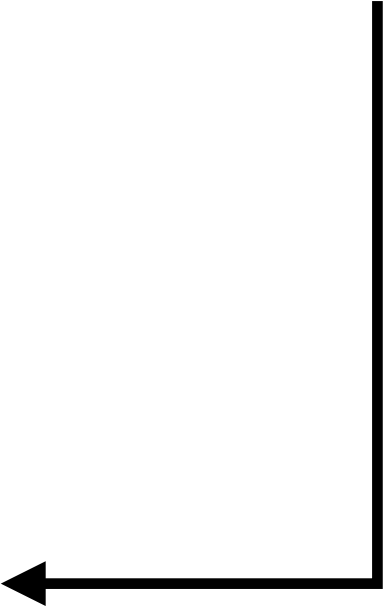




contravariance

covariance





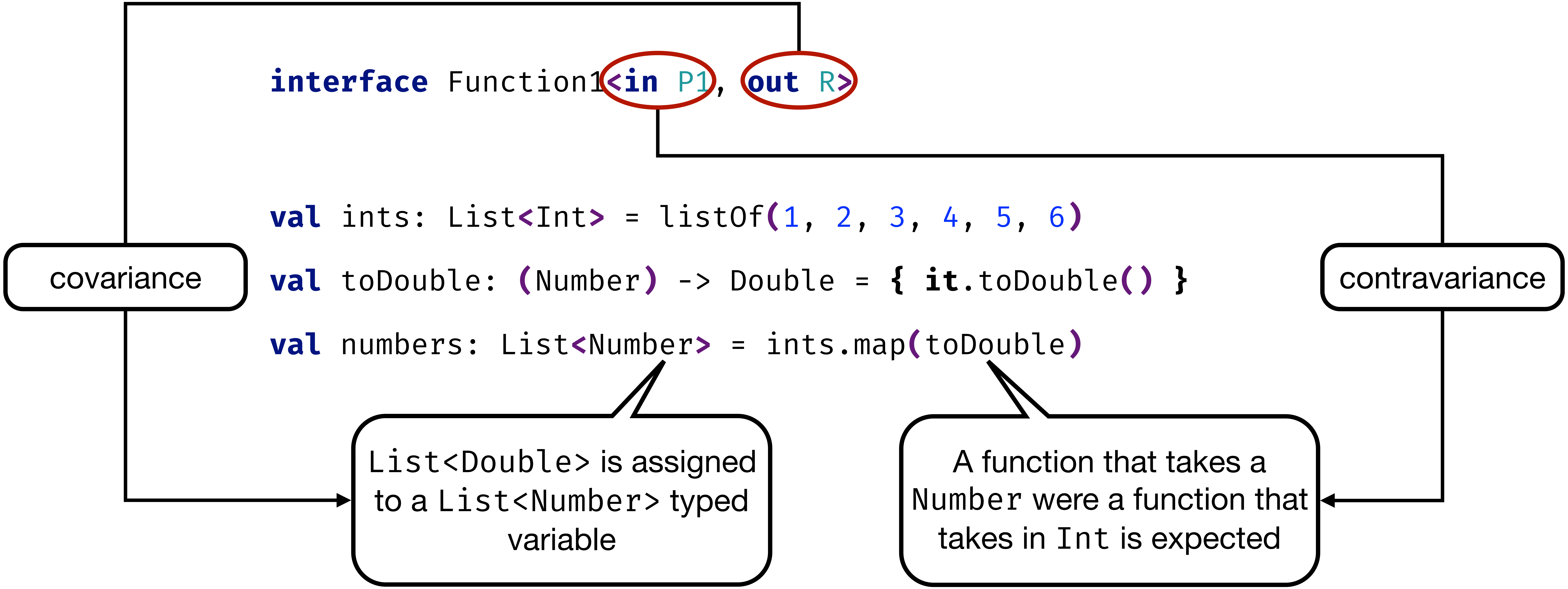
```kotlin
val ints: List<Int> = listOf(1, 2, 3, 4, 5, 6)

val toDouble: (Number) -> Double = { it.toDouble() }

val numbers: List<Number> = ints.map(toDouble)
```

```
interface Function1<in P1, out R>
```

Generics



```
interface Function1<in P1, out R>


val ints: List<Int> = listOf(1, 2, 3, 4, 5, 6)

val toDouble: (Number) -> Double = { it.toDouble() }

val numbers: List<Number> = ints.map(toDouble)
```

covariance

contravariance

List<Double> is assigned to a List<Number> typed variable

A function that takes a Number were a function that takes in Int is expected