Functional literal with receiver

special form of functional literals

```
val sum: (Int, Int) -> Int = { a: Int, b: Int -> a + b }
```

```kotlin
val sum: Int.(Int) -> Int = { other -> this + other }
```

V

a

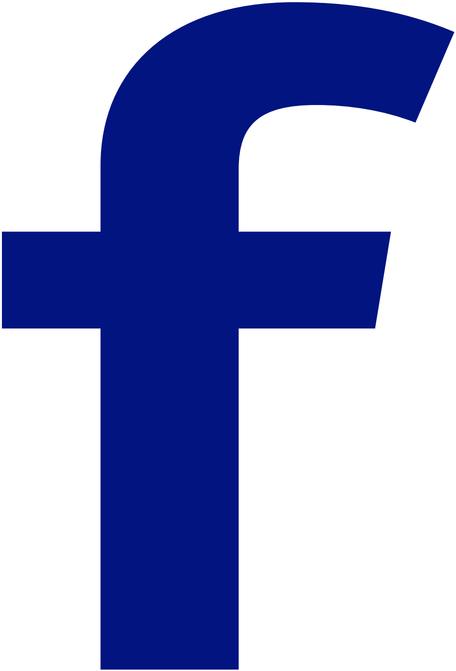
ι

S

u

m

[redacted]



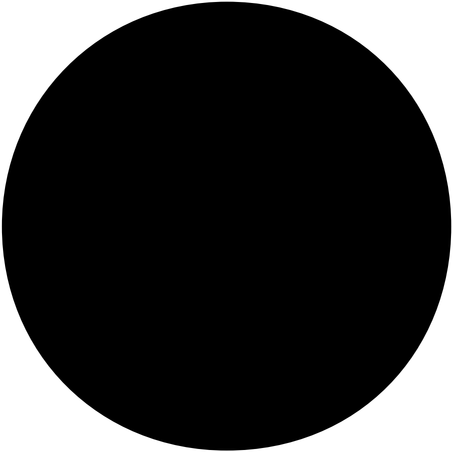
u

n

I

n

t



O

t

h

e

r

●

●

I

n

t

I

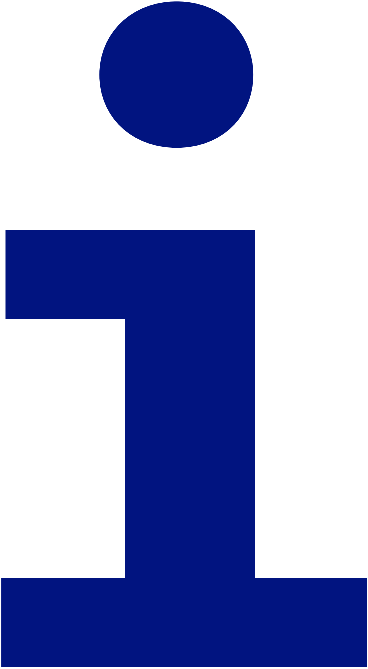
n

t

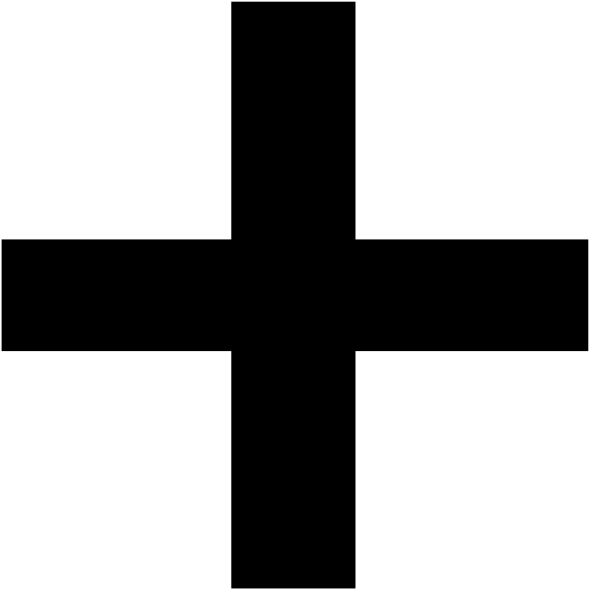
=

t

h



S

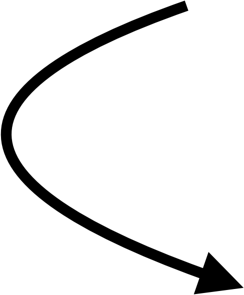


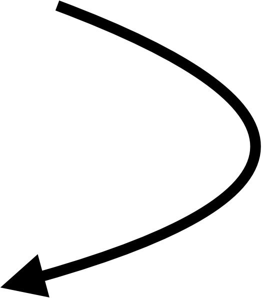
0

h

e

r





lambda with receiver

anonymous function syntax

```
sum(2, 3)
```

22

$\{$

$\}$

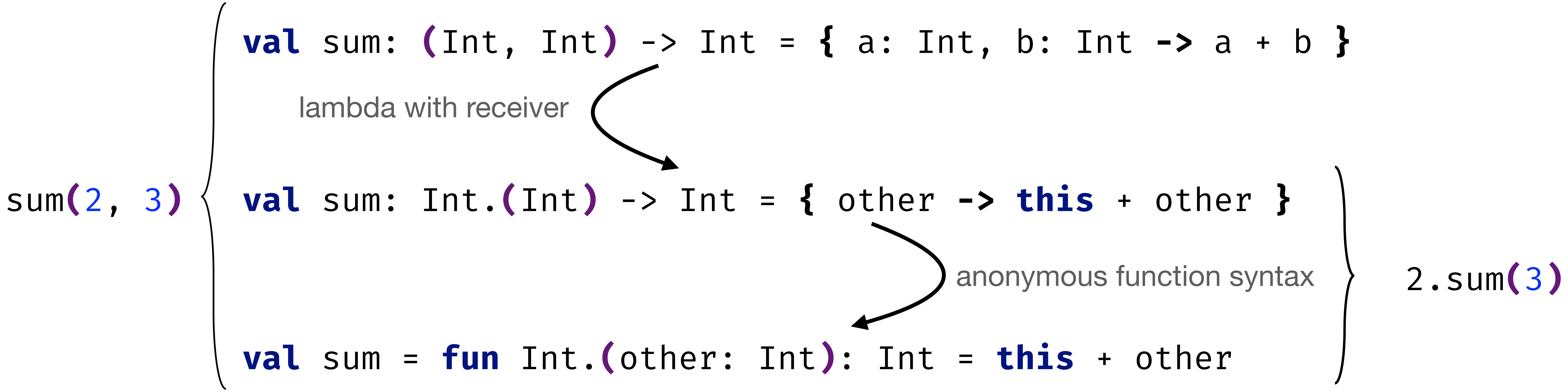
receiver type

```
2.sum(3)
```

```kotlin
val sum = fun Int.(other: Int): Int = this + other
```

Functional literal with receiver

special form of functional literals



```
sum(2, 3)
```

```
val sum: (Int, Int) -> Int = { a: Int, b: Int -> a + b }
```

lambda with receiver

```
val sum: Int.(Int) -> Int = { other -> this + other }
```

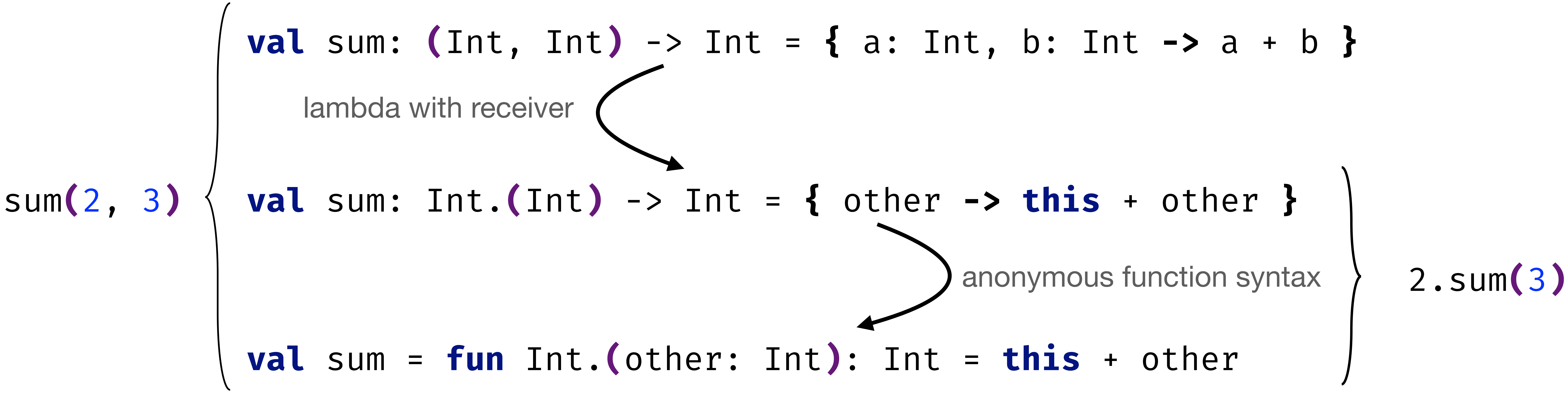
anonymous function syntax

```
val sum = fun Int.(other: Int): Int = this + other
```

```
2.sum(3)
```

Functional literal with receiver

special form of functional literals



sum(2, 3)

```
val sum: (Int, Int) -> Int = { a: Int, b: Int -> a + b }

val sum: Int.(Int) -> Int = { other -> this + other }

val sum = fun Int.(other: Int): Int = this + other
```

lambda with receiver

anonymous function syntax

2.sum(3)