

Functional literacy

special form aliterals









































































































































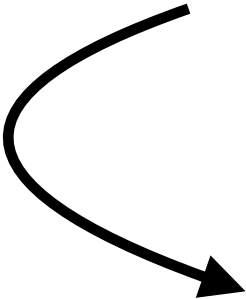




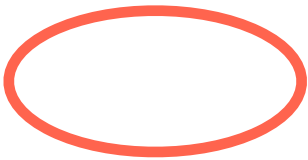


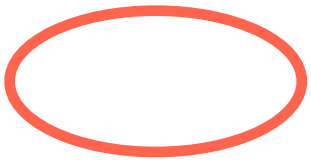


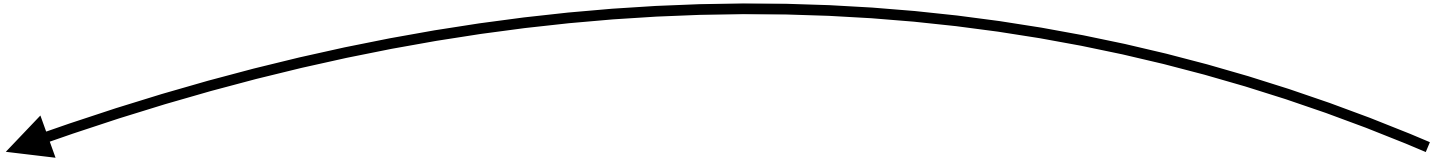




amboda with receiver



































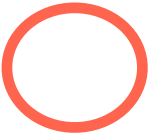






receiving *rent* type





researcher object

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

sum(2,3)

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

2. sum(3)

Functional literal with receiver

special form of functional literals

val sum: (Int, Int) -> Int = { a: Int, b: Int -> a + b } sum(2, 3)

lambda with receiver

receiver type

receiver object

val sum: Int.(Int) -> Int = { other -> this + other } 2.sum(3)

```
graph TD; A["lambda with receiver"] --> B["Int."]; C["receiver type"] --> D["Int"]; E["receiver object"] --> F["2"];
```

Functional literal with receiver

special form of functional literals

```
val sum: (Int, Int) -> Int = { a: Int, b: Int -> a + b }    sum(2, 3)
```

lambda with receiver

receiver type

```
val sum: Int.(Int) -> Int = { other -> this + other }    2.sum(3)
```