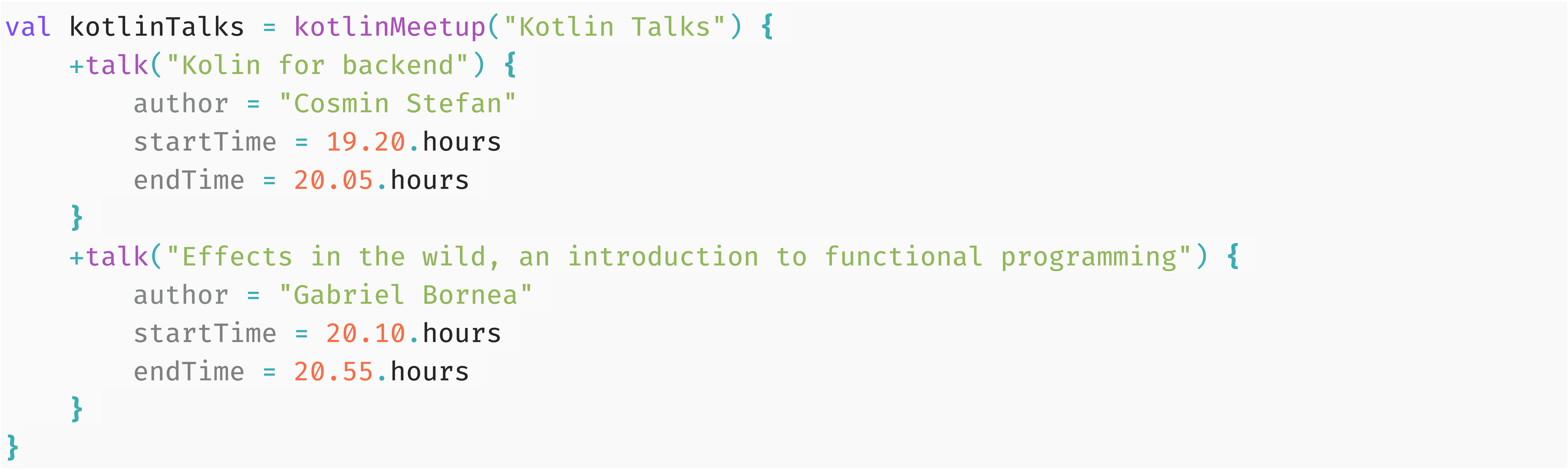
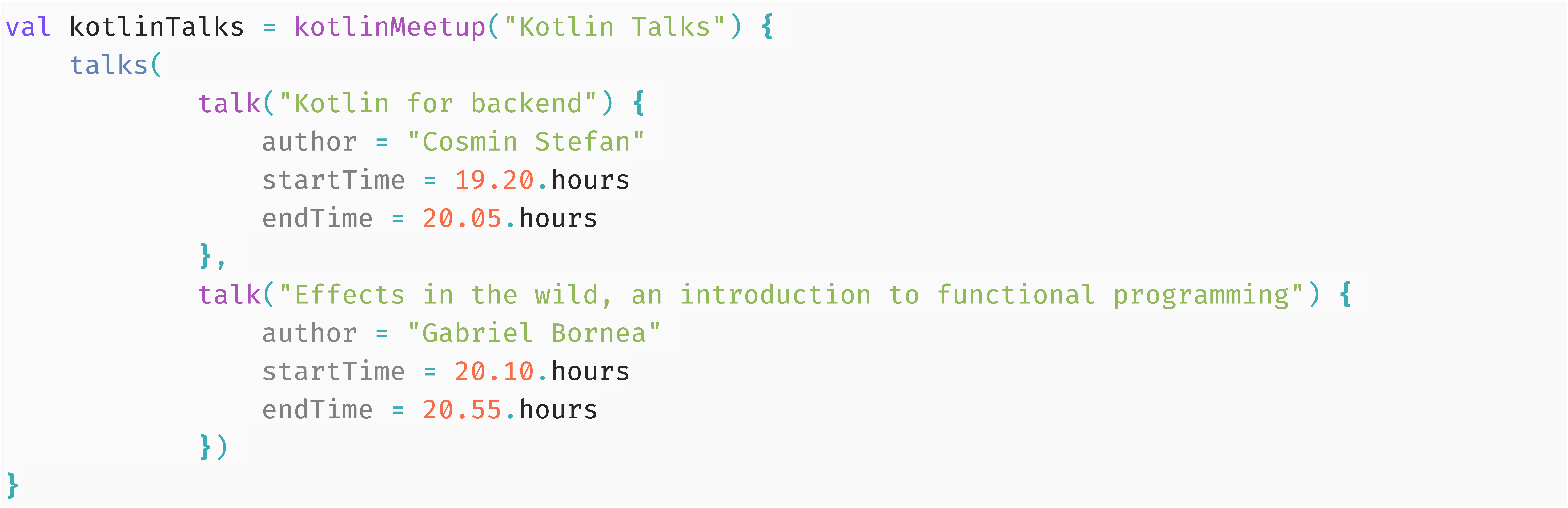
ING



```kotlin
val kotlinTalks = kotlinMeetup("Kotlin Talks") {
    +talk("Kolin for backend") {
        author = "Cosmin Stefan"
        startTime = 19.20.hours
        endTime = 20.05.hours
    }
    +talk("Effects in the wild, an introduction to functional programming") {
        author = "Gabriel Bornea"
        startTime = 20.10.hours
        endTime = 20.55.hours
    }
}
```



```kotlin
val kotlinTalks = kotlinMeetup("Kotlin Talks") {
    talks(
            talk("Kotlin for backend") {
                author = "Cosmin Stefan"
                startTime = 19.20.hours
                endTime = 20.05.hours
            },
            talk("Effects in the wild, an introduction to functional programming") {
                author = "Gabriel Bornea"
                startTime = 20.10.hours
                endTime = 20.55.hours
            })
}
```

0

P

E

R

A

T

0

R

0

V

E

R

L

0

A

D

N

G

```kotlin
class KotlinMeetupBuilder(private var title: String) {
    private val talks = mutableListOf<Talk>()
    fun build() = KotlinMeetup(title, talks)

}
```

O

p

e

r

a

t

O

r

f

u

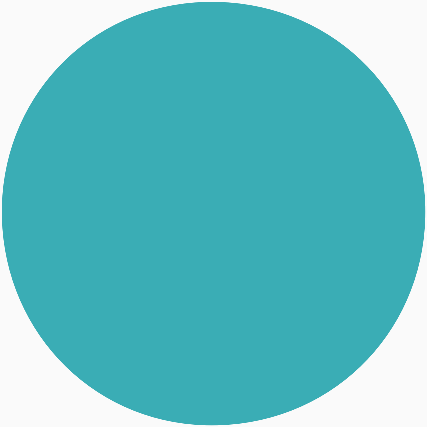
n

T

a

l

k



u

n

a

r

y

P

l

u

S

(

)

:

T

a

l

k

a

p

p

l

y

{

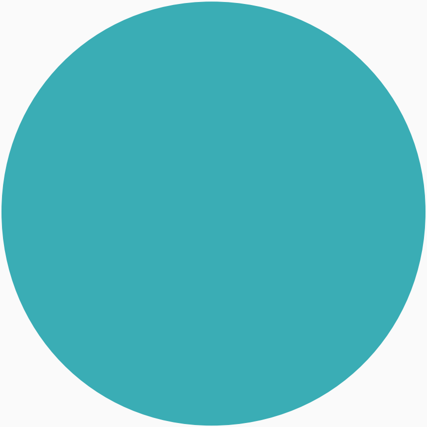
t

a

ʅ

k

S



a

d

d

t

h

i

S

}

```
fun kotlinMeetup(title: String, init: KotlinMeetupBuilder.() → Unit): KotlinMeetup
fun talk(title: String, init: Talk.() → Unit): Talk
```

OPERATOR OVERLOADING

```kotlin
operator fun Talk.unaryPlus(): Talk = apply { talks.add(this) }
```

```kotlin
fun kotlinMeetup(title: String, init: KotlinMeetupBuilder.() → Unit): KotlinMeetup
fun talk(title: String, init: Talk.() → Unit): Talk
```

```kotlin
val kotlinTalks = kotlinMeetup("Kotlin Talks") {
    +talk("Kolin for backend") {
        author = "Cosmin Stefan"
        startTime = 19.20.hours
        endTime = 20.05.hours
    }
    +talk("Effects in the wild, an introduction to functional programming") {
        author = "Gabriel Bornea"
        startTime = 20.10.hours
        endTime = 20.55.hours
    }
}
```