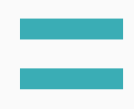
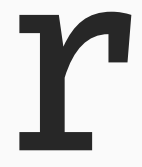
ING

```
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
    }
}
```

```
val kotlinTalks = kotlinMeetup("Kotlin Talks") {
    +"Kotlin, a language to rule DSLs".by("Tiberiu Tofan").from(18.30).to(19.15)
    +"Kotlin for backend".by("Cosmin Stefan").from(19.20).to(20.05)
    +"Effects in the wild, an introduction to fun…”.by(“Gabriel Bornea").from(20.10).to(20.55)
}
```

S





d

d

r

S

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

MORE EXTENSION FUNCTIONS

```
fun String.by(author: String): Talk = Talk(this, author)
```

```
fun Talk.from(startTime: Double): Talk = apply { this.startTime = startTime.hours }
```

```
fun Talk.to(endTime: Double): Talk = apply { this.endTime = endTime.hours }
```

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

```
fun String.by(author: String): Talk = Talk(this, author)
fun Talk.from(startTime: Double): Talk = apply { this.startTime = startTime.hours }
fun Talk.to(endTime: Double): Talk = apply { this.endTime = endTime.hours }
```

```
val kotlinTalks = kotlinMeetup("Kotlin Talks") {
    +"Kotlin, a language to rule DSLs".by("Tiberiu Tofan").from(18.30).to(19.15)
    +"Kotlin for backend".by("Cosmin Stefan").from(19.20).to(20.05)
    +"Effects in the wild, an introduction to fun…".by(“Gabriel Bornea").from(20.10).to(20.55)
}
```

ING