

ING



```
val kotlinDsl = talk("Kotlin, a language ..", {  
    it.author = "Tiberiu Tofan"  
    it.startTime = 18.30.hours  
    it.endTime = 19.15.hours  
})
```

```
val kotlinDsl = talk("Kotlin, a language ..", {  
    this.author = "Tiberiu Tofan"  
    this.startTime = 18.30.hours  
    this.endTime = 19.15.hours  
})
```

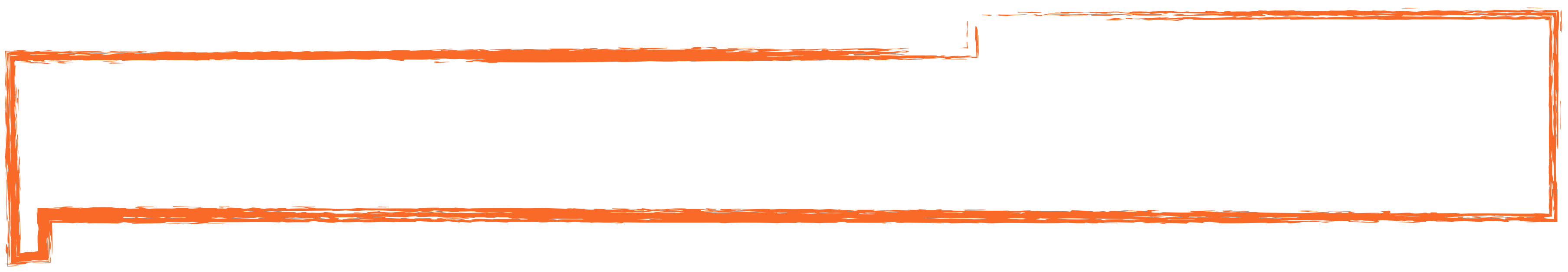
```
fun talk(title: String, init:      ) → Unit): Talk {  
    val talk = Talk(title)  
  
    return talk  
}
```

```
fun talk(title: String, init:Talk.() → Unit): Talk {  
    val talk = Talk(title)  
    talk.init()  
    return talk  
}
```



```
data class KotlinMeetup(val title: String, val talks: List<Talk>)
data class Talk(val title: String, var author: String, var startTime: LocalTime, var endTime: LocalTime)
```

TYPE-SAFE BUILDERS USING FUNCTIONAL INTERFACES WITH RECEIVER





```
data class KotlinMeetup(val title: String, val talks: List<Talk>)  
data class Talk(val title: String, var author: String, var startTime: LocalTime, var endTime: LocalTime)
```

TYPE-SAFE BUILDERS USING FUNCTIONAL LITERALS WITH RECEIVER

```
fun talk(title: String, init: Talk.() → Unit): Talk {  
    val talk = Talk(title)  
    talk.init()  
    return talk  
}
```

```
val kotlinDsl = talk("Kotlin, a language ..", {  
    this.author = "Tiberiu Tofan"  
    this.startTime = 18.30.hours  
    this.endTime = 19.15.hours  
})
```

```
data class KotlinMeetup(val title: String, val talks: List<Talk>)  
data class Talk(val title: String, var author: String, var startTime: LocalTime, var endTime: LocalTime)
```

TYPE-SAFE BUILDERS USING FUNCTIONAL LITERALS WITH RECEIVER

```
fun talk(title: String, init: Talk.() → Unit): Talk {  
    val talk = Talk(title)  
    talk.init()  
    return talk  
}
```

```
val kotlinDsl = talk("Kotlin, a language ...", {  
    this.author = "Tiberiu Tofan"  
    this.startTime = 18.30.hours  
    this.endTime = 19.15.hours  
})
```