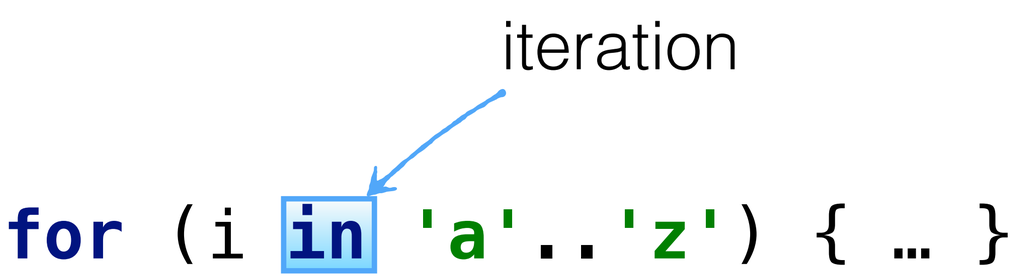
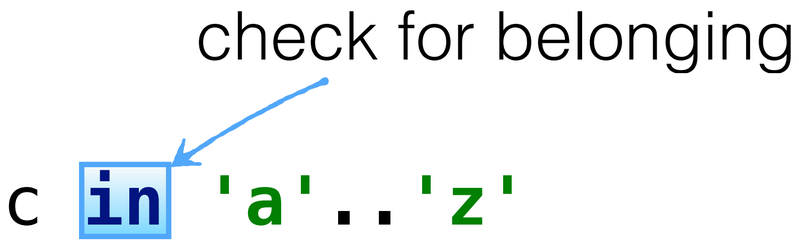
‘`in`’ checks and ranges

`in`: two use-cases





`in` a range

```
fun isLetter(c: Char) = c in 'a'..'z' || c in 'A'..'Z'
```

```
isLetter('q')     // true
isLetter('*')     // false
```

`in` a range

```
fun isLetter(c: Char) = c in 'a'..'z' || c in 'A'..'Z'
```

```
c in 'a'..'z'
```

is compiled to

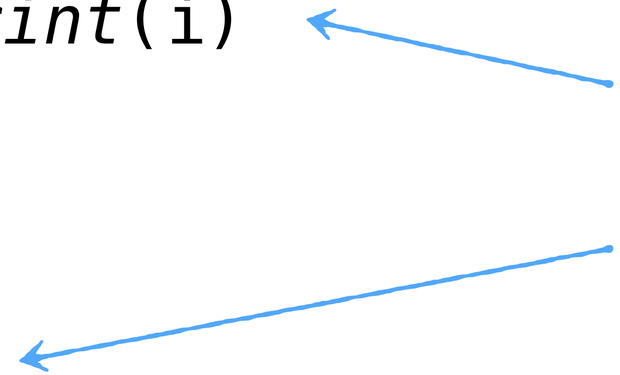
```
'a' <= c && c <= 'z'
```

Ranges

```
for (i in 1..9) {
    print(i)
}
```

```
c in '0'..'9'
```

Optimized. No extra `IntRange` or `CharRange` objects are created



not `in` a range

```
fun isNotDigit(c: Char) = c !in '0'..'9'
```

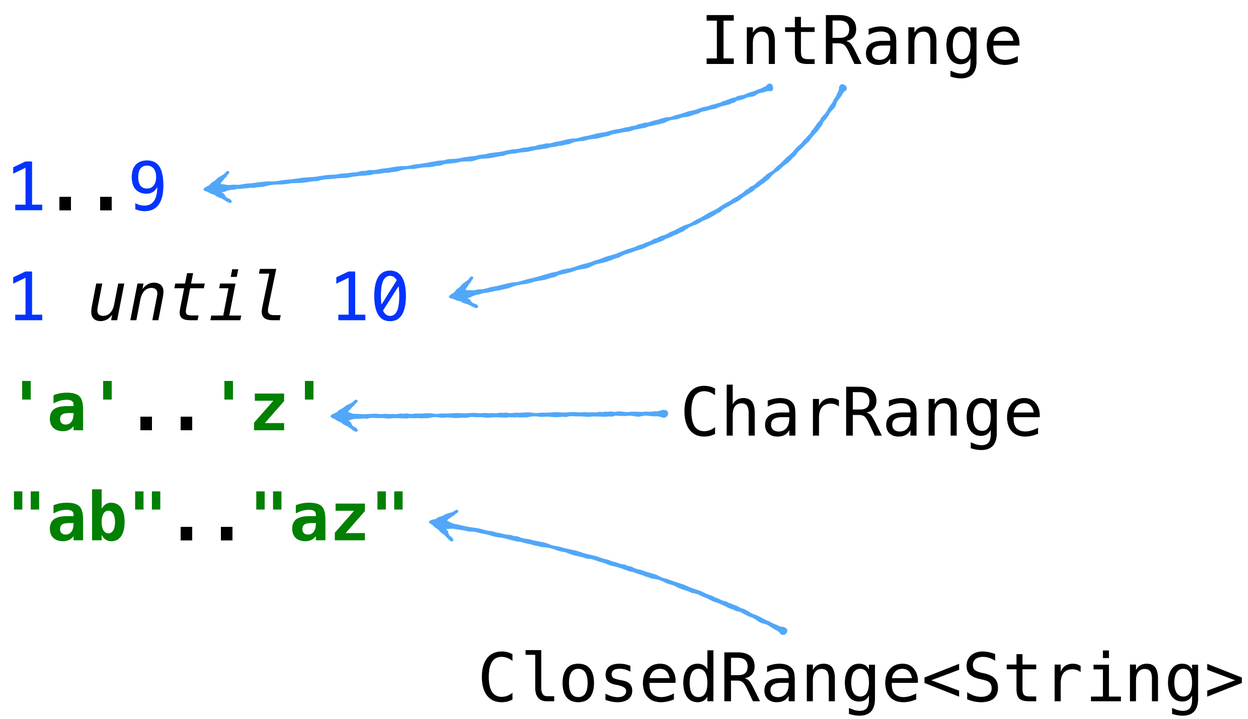
```
isNotDigit('x')   // true
```

`in` as `when`-condition

```
fun recognize(c: Char) = when (c) {
    in '0'..'9' -> "It's a digit!"
    in 'a'..'z', in 'A'..'Z' -> "It's a letter!"
    else -> "I don't know…"
}
```

```
recognize('$')     // I don't know…
```

Ranges


IntRange
1..9
1 until 10
'a'..'z'
CharRange
"ab".."az"
ClosedRange<String>

What will be printed?

```
println("Kotlin" in "Java".."Scala")
println("Kotlin" in setOf("Java", "Scala"))
```

1. `true true`
2. `true false`
3. `false true`
4. `false false`

What will be printed?

```
println("Kotlin" in "Java".."Scala")          // true
println("Kotlin" in setOf("Java", "Scala")) // false
```

1. true true
2. true false
3. false true
4. false false

Comparing `Strings`

```
"ball" in "a".."k"
```

```
"a" <= "ball" && "ball" <= "k"
```

```
"a".compareTo("ball") <= 0 && "ball".compareTo("k") <= 0
```

`Strings` are compared alphabetically

Range of `Strings`

```
"ball" in "a".."k"          // true
"zoo" in "a".."k"           // false

"Kotlin" in "Java".."Scala"     // true
```

What will be printed?

```
println("Kotlin" in "Java".."Scala")          // true
println("Kotlin" in setOf("Java", "Scala")) // false
```

1. true true
2. true false
3. false true
4. false false