

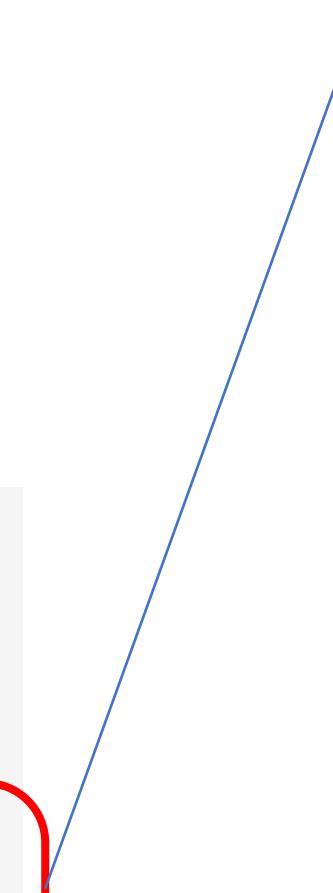
Gradle: Kotlin Grammer for Gradle

```
fun <T, R> Collection<T>.fold(  
    initial: R,  
    combine: (acc: R, nextElement: T) -> R  
>: R {  
    var accumulator: R = initial  
    for (element: T in this) {  
        accumulator = combine(accumulator, element)  
    }  
    return accumulator  
}
```

combine: (acc : R, next : T) -> R = { acc, I -> body }

// Lambdas are code blocks enclosed in curly braces.

```
items.fold(0, {  
    // When a lambda has parameters, they go first, followed by '->'  
    acc: Int, i: Int ->  
    print("acc = $acc, i = $i, ")  
    val result = acc + i  
    println("result = $result")  
    // The last expression in a lambda is considered the return value:  
    result  
})
```



Gradle: Kotlin Grammer for Gradle

"Lambda with receiver"에 대한 이해

sum : Int.(other: Int) -> Int

fun main(args: Array<String> {

 val sum1 : Int.(num: Int) -> Int = {this+it}

 val sum2 : Int.(Int) -> Int = {this+it}

 println(1.sum1(2)) Lambda의 Parameter가 하나인 경우 it으로 참조 가능하다.

 println(1.sum2(2))

}

Receiver 덕분에 body가 Receiver 객체 내부에 존재하는 것처럼
만들어 준다.

따라서, this를 통해 Receiver 객체 참조가 가능하다.

Receiver를 사용하면 Lambda의 Body에서 해당 Body가 Receiver 객체
내부에 존재하듯이 Receiver 객체의 method 또는 member를 참조할
수 있다.

```
fun main(args: Array<String>) {  
    val test : (HTML.() -> Unit) = {  
        println("test")  
    }  
    html(test)  
}
```

```
class HTML {  
    fun body() {  
        println("HTML")  
    }  
}
```

```
fun html(aaa: HTML.() -> Unit): HTML {  
    val html = HTML() // receiver 객체 생성  
    html.aaa() // receiver 객체를 람다로 전달  
    return html  
}
```

Fun html의 파라미터가 함수 하나이므로 Html {}처럼 선언 가능하다.
이때, Body 안에서 Receiver의 method 호출 가능

```
//html { // receiver가 있는 람다의 시작점  
//    body() // receiver 객체의 메소드를 호출  
//}
```

Gradle: Kotlin Grammer for Gradle

```
repositories {  
    mavenCentral()  
}
```

해당 scrip가 어떻게 Kotlin으로 이루어 지는지 살펴보자.

Repositories의 선언

Repositories method의 parameter는 하나의 lambda function을 받고 있다.

- repositories({ lambda }) → repositories { lambda } 가능
- repositories { lambda } 의 lambda body 안에서 receiver 접근 가능

```
public fun org.gradle.api.Project.repositories (configuration: org.gradle.api.artifacts.dsl.RepositoryHandler.() -> kotlin.Unit)  
{ /* compiled code */ }
```

Receiver가 존재한다
→ 따라서 해당 lambda의 body 안에서 Receiver 객체의 method 또는 member에 접근 가능하다.

❖ Receiver인 RepositoryHandler 내부에 mavenCentral() method가 존재한다.

```
@HasInternalProtocol  
public interface RepositoryHandler extends ArtifactRepositoryContainer {  
    //...  
    MavenArtifactRepository mavenCentral(Action<? super MavenArtifactRepository> action)  
}
```