ING

```kotlin
//omitted imports
fun main() {
    System.out.appendHTML().html {
        head {
            meta {
                title = "Generated in Kotlin"
            }
        }
        body {


           div {
                a(href = "https://kotlinlang.org") {
                    target = ATarget.blank
                    +"Main site"
                }
            }
        }
    }
}
```

```kotlin
@DslMarker
annotation class HtmlTagMarker
```

KOTLINX HTML DSL

h

e

a

d

{

}

```kotlin
@HtmlTagMarker
fun HTML.head(block : HEAD.() → Unit = {}) : Unit
```

```kotlin
@HtmlTagMarker
fun HTML.body(classes : String? = null,
                  block : BODY.() -> Unit = {}) : Unit
```

t

h

i

S

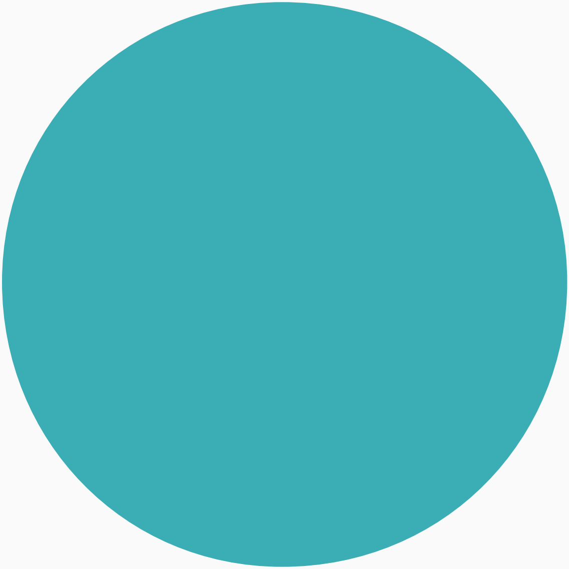
@

h

t

m

Լ



h

e

a

d

{

}

```
'fun HTML.head(block: HEAD.() → Unit = … ):
Unit' can't be called in this context by
implicit receiver. Use the explicit one if
necessary
```

```
head {
}
```

KOTLINX HTML DSL

```
//omitted imports
fun main() {
    System.out.appendHTML().html {
        head {
            meta {
                title = "Generated in Kotlin
            }
        }
        body {
            this@html.head {
            }
            div {
                a(href = "https://kotlinlang.org") {
                    target = ATarget.blank
                    +"Main site"
                }
            }
        }
    }
}
```

```
@DslMarker
annotation class HtmlTagMarker
```

```
@HtmlTagMarker
fun HTML.head(block : HEAD.() → Unit = {}) : Unit
```

```
@HtmlTagMarker
fun HTML.body(classes : String? = null,
              block : BODY.() → Unit = {}) : Unit
```

@DSLMARKER

When applied to annotation class X specifies that X defines a DSL language

The general rule:

- an implicit receiver may belong to a DSL @X if marked with a corresponding DSL marker annotation
- two implicit receivers of the same DSL are not accessible in the same scope
- the closest one wins
- other available receivers are resolved as usual, but if the resulting resolved call binds to such a receiver, it's a compilation error