

```
x: Integer = 42  
y: Float = sin(x)  
z: String = "Value"
```

Type Checking

And use **Perspectives** (often combined)

```
assert(x >= 0)  
a <- angleCmp(x)  
  
verify(  
    0 <= a &&  
    a <= 40  
)
```

Verification

```
for (i in 1:n) {  
    sum <- sum + i  
}  
  
sum <- n*(n+1)/2
```

Optimization

```
x <- 42  
if (x < 0) {  
    print("Negative")  
} # Dead Code
```

Linting

```
inp <- read.csv("data.csv")  
clean <- filter(inp, a>5)  
plot(inp$a, inp$a)
```

Slicing

```
x <- u + 0  
print(x)  
  
y <- u + 0  
print(y)
```

Refactoring

Some of those results are the result of other analyses or dynamic analyses

And **Communicate or Use** results

