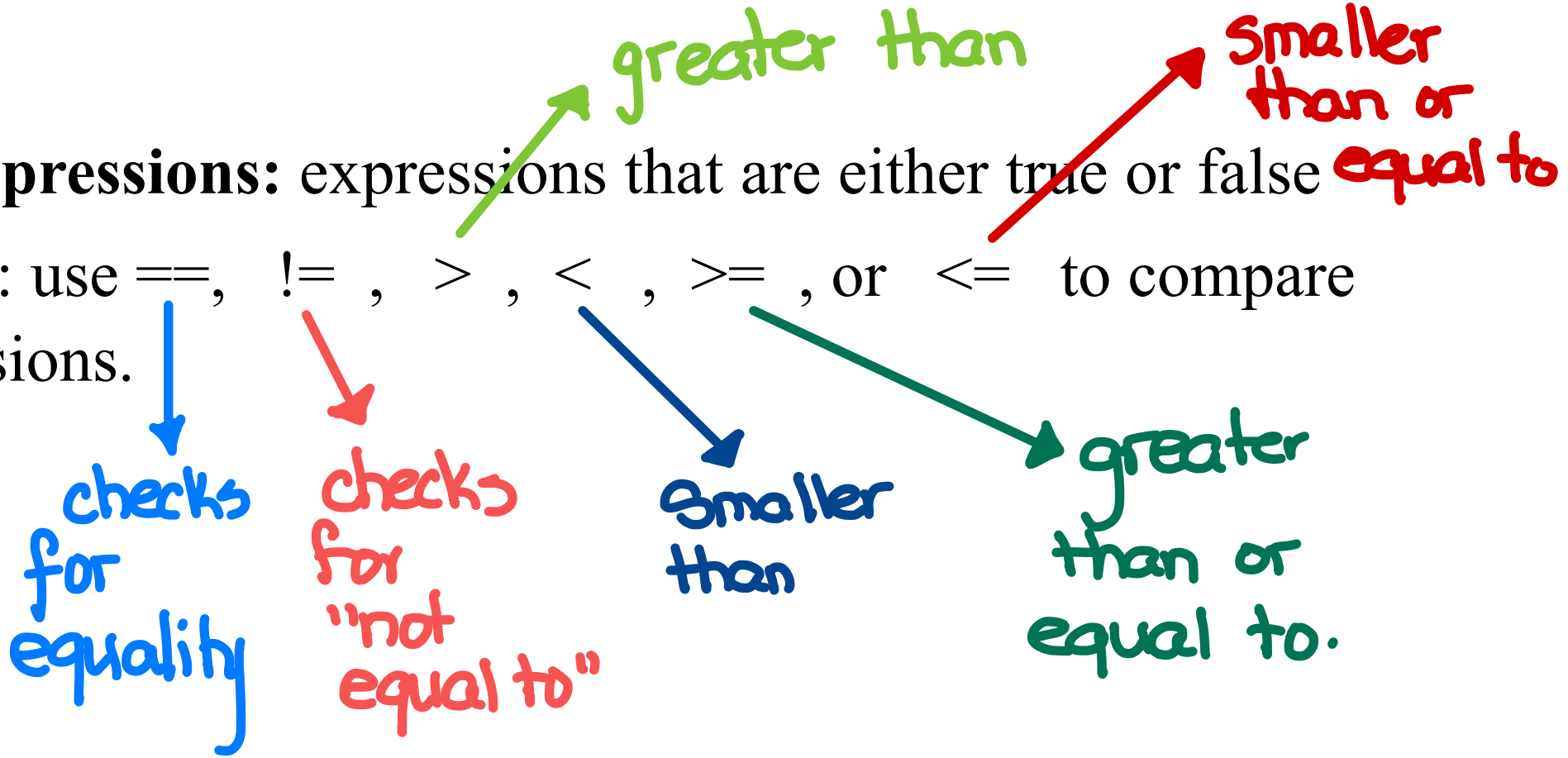
If/Else Statements & For Loops

Boolean Expressions

Boolean expressions: expressions that are either true or false

Commands: use ==, != , > , < , >= , or <= to compare two expressions.



Example 1: Suppose you defined the following vectors:

```
s <- 5
m <- 5
```

What are the outputs if I ran the following lines of codes?

Code	Output
`s == m`	TRUE
`s < m`	FALSE
`s != m`	FALSE
`s >= m`	TRUE

Example 2: Suppose you defined the following vectors:

```
x <- c(4,5,6,7)
y <- 4:7
z <- c(4, 5, 7, 9)
```

4 5 6 7

4 5 6 7

4 5 7 9

What are the outputs if I ran the following lines of codes?

When you compare vectors, it makes a comparison between corresponding elements

Code	Output
`x == y`	TRUE TRUE TRUE TRUE
`x == z`	TRUE TRUE FALSE FALSE
`x != z`	FALSE FALSE TRUE TRUE
`x > 6`	FALSE FALSE FALSE TRUE
`x <= 5`	TRUE TRUE FALSE FALSE

Conditional Statements

If Statements

The “if” statement allows you to execute a block of code only if a given condition is true.

Example 3: What is the output when you run this code?

```
sky <- "sunny"
if (sky == "sunny"){
  print("Leave your umbrella at home!")
}
```

if (boolean expression) {
code
}

Code will run if boolean expression is TRUE.

Output: "Leave your umbrella at home!"

What is the output when you change the first line to `sky <- "cloudy"`?

Output: (blank)

Example 4: Define a one element vector named *number* whose one element is 5. Write an if statement that prints "It's a positive number" if the value of vector *number* is greater than 0. What is the output once you run these lines?

Code:

```
number <- 5
if (number > 0) {
   print("It's a positive number")
}
```

Output: "It's a positive number"

What is the output when you change the first line to `number <- -5`?

Output: (blank)

If-Else Statements

The “if else” statement extends the “if” statement to execute different blocks of code based on whether a condition is true or false.

Example 5: What is the output when you run this code?

```
number <- -3
if (number > 0) {
  print("The number is positive.")
} else if (number < 0) {
  print("The number is negative.")
} else {
  print("The number is zero.")
}
```

Output: "The number is negative"

Example 6: Define a one element vector named age whose one element is 18. Write an if-else statement about driving in Illinois. The statement should print "you are too young to drive" if the age is less than 16, "you need written consent from parents" if the age is between 16 and 17, and "no age restriction to get a license" if age is 18 or greater.

Code:

```
age <- 18
if(age < 16){
    print("you are too young to drive")
}else if (age >= 16 & age <= 17){
    print("you need consent from your parents")
}else if (age >= 18){
    print("no age restriction to get a license")
}
```

In R
AND &
OR |

Both condition need to be met

either condition could be met

Output: "no age restriction to get a license"

Ifelse Function ← use when vectors in your boolean expression

This is for when you want to use if else statements repeatedly to return a vector.

Example 7: What is the output when you run this code?

```
numbers <- c(-2, 3, 0, -5, 7)
positive_indicator<-ifelse(numbers>0, "Positive", "Not Positive")
numbers
positive_indicator
```

value it will take if TRUE

Boolean expression

value it will take if FALSE

Output: -2 3 0 -5 7

"Not Positive" "Positive" "Not Positive" "Not positive"

"Positive"

Example 8: Define a vector named numbers_2 with elements 5, -2, 0. Then define a new vector named eval_5 using the ifelse function that tells you whether the numbers in numbers_2 are greater or equal to 5 or smaller than 5.

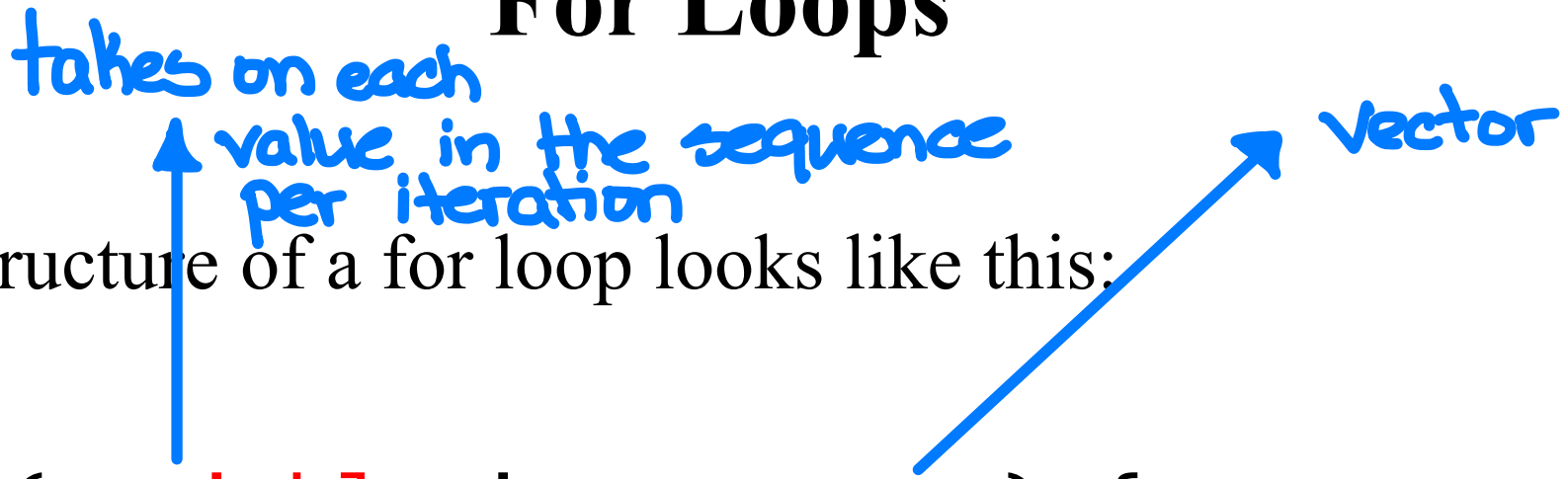
Code:

```
numbers_2 <- c(5,-2,0)
eval_5 <- ifelse(numbers_2 >= 5, "Greater or equal to 5",
                 "Smaller than 5")
```

Output:

```
"Greater or equal to 5"  "Smaller than 5"
"Smaller than 5"
```

For Loops



The basic structure of a for loop looks like this:

```
for (variable in sequence) {
    code to be executed in each iteration
}
```

1 2 3 4 5 (we have 5 elements, therefore we will have 5 iterations)

Example 9: What is the output when you run this code?

```
for (i in 1:5){
  print(i)
}
```

Output:
1
2
3
4
5

Behind the scenes
5 Iterations

1st Iteration i=1 → print(1)
2nd Iteration i=2 → print(2)
3rd Iteration i=3 → print(3)
4th Iteration i=4 → print(4)
5th Iteration i=5 → print(5)

3 elements, therefore 3 iterations

Example 10: What is the output when you run this code?

```
for (i in c(2, 3, 5)){
  print(i+1)
}
```

Output: 3
4
6

Behind the scenes
3 Iterations

i+1

1st Iteration i=2 → print(3)
2nd Iteration i=3 → print(4)
3rd Iteration i=5 → print(6)

Example 11: What is the output when you run this code?

```
for (i in c("Mike", "Mary", "Tom")){
  print(paste(i, "works at Loyola", sep = " - "))
}
```

Output: "Mike - works at Loyola"
"Mary - works at Loyola"
"Tom - works at Loyola"

Behind the scenes
3 Iterations
1st It. i = "Mike" → "Mike - work...
2nd It. i = "Mary" → "Mary - work...
3rd It. i = "Tom" → "Tom - work...

Example 12: Create a character vector called student_names that contains the elements “Rohan”, “Ana”, and “Amir”. The create a for loop to print the following information:

Student 1 is Rohan

Student 2 is Ana

Student 3 is Amir

Code:

```
student_names<-c("Rohan","Ana","Amir")
for(i in 1:3){
  print(paste("Student",i,"is",student_names[i],
              sep=" "))
```

Output:

```
Student 1 is Rohan
Student 2 is Ana
Student 3 is Amir
```

Example 13: Doing computations with for loops. What is the output when you run this code?

```
x <- matrix(c(9, 0, 3, 17, 5, 2), ncol = 3, byrow = FALSE)
y<-NA
for(i in 1:3){
    y[i] <- mean(x[,i])
  }
  y
```

y<-NA ← place holder, opens the vector y.

$$\begin{bmatrix} 9 & 3 & 5 \\ 0 & 17 & 2 \end{bmatrix}$$

3 iterations

1st: i=1; y[1]<-mean(x[,1])

Will compute the mean of column 1 in matrix m and will save it in the 1st element of vector y

⋮

Output: 4.5 10 3.5