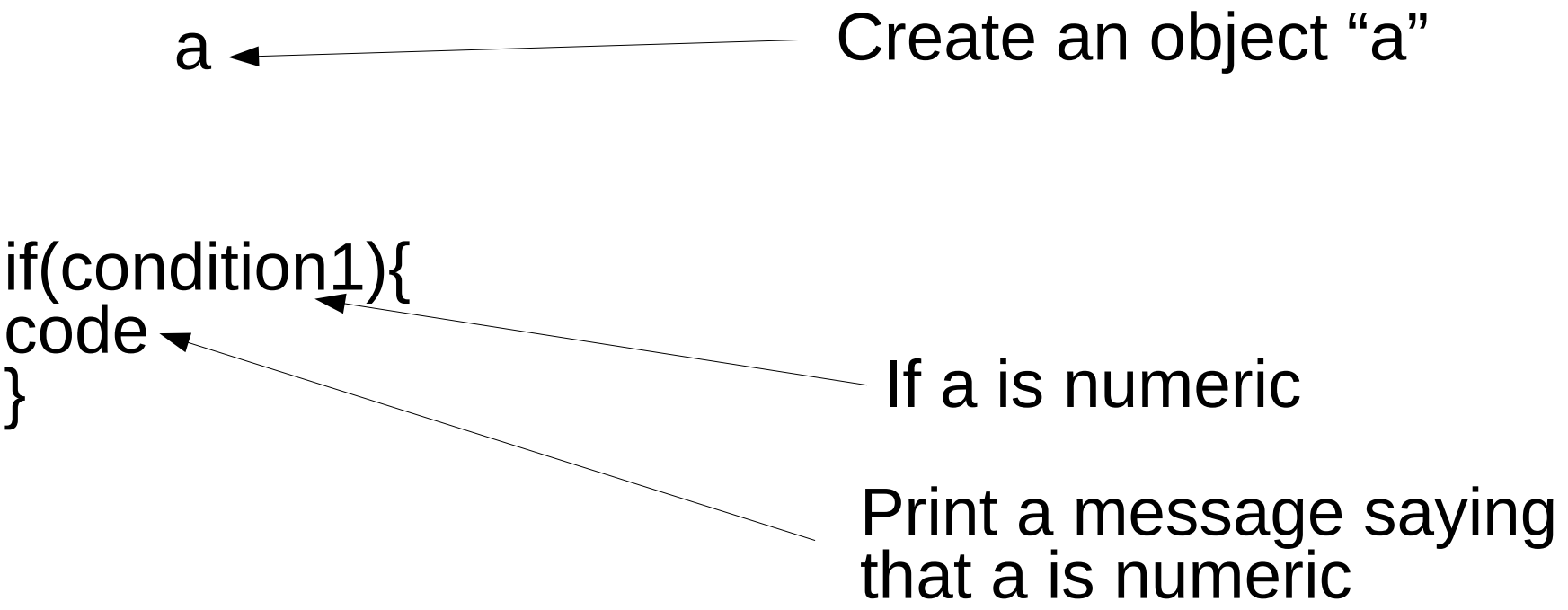
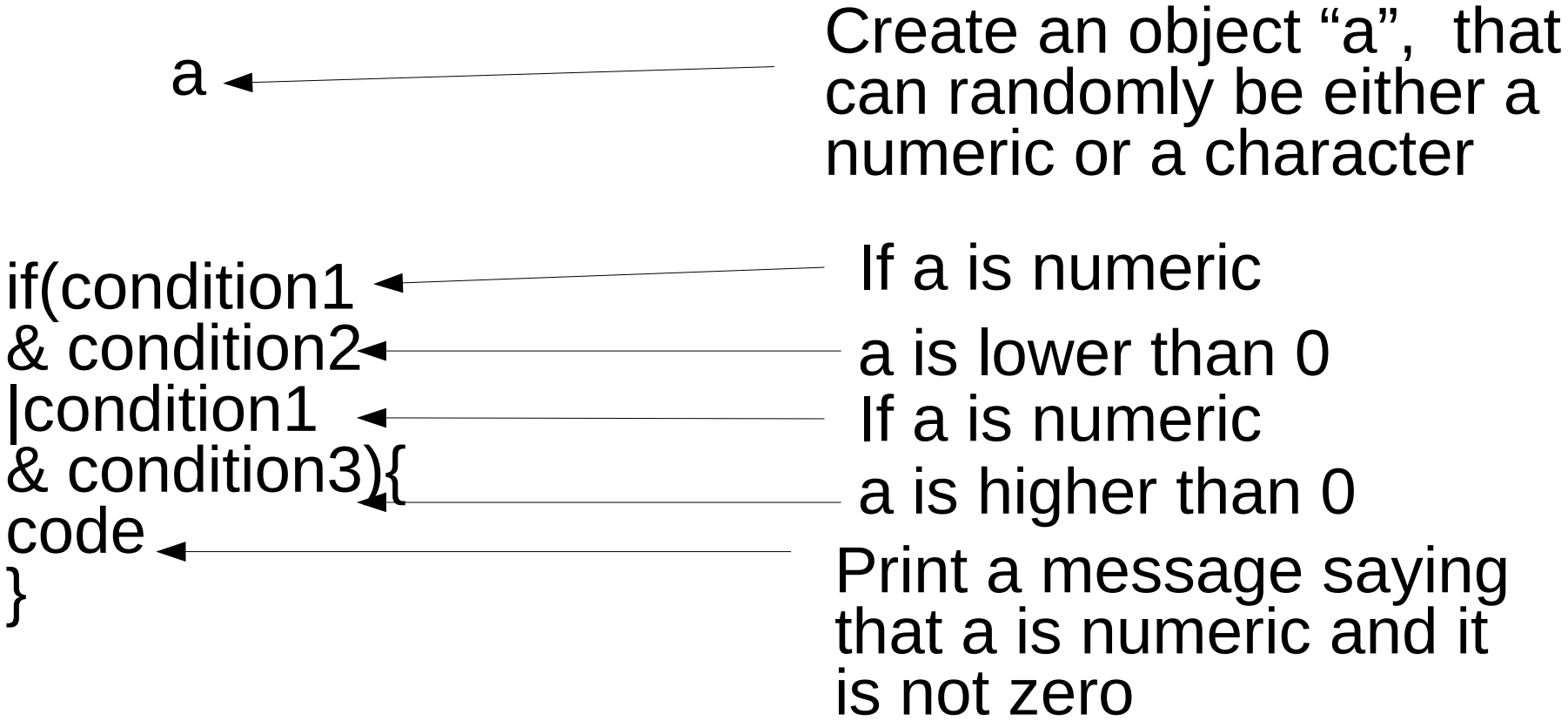
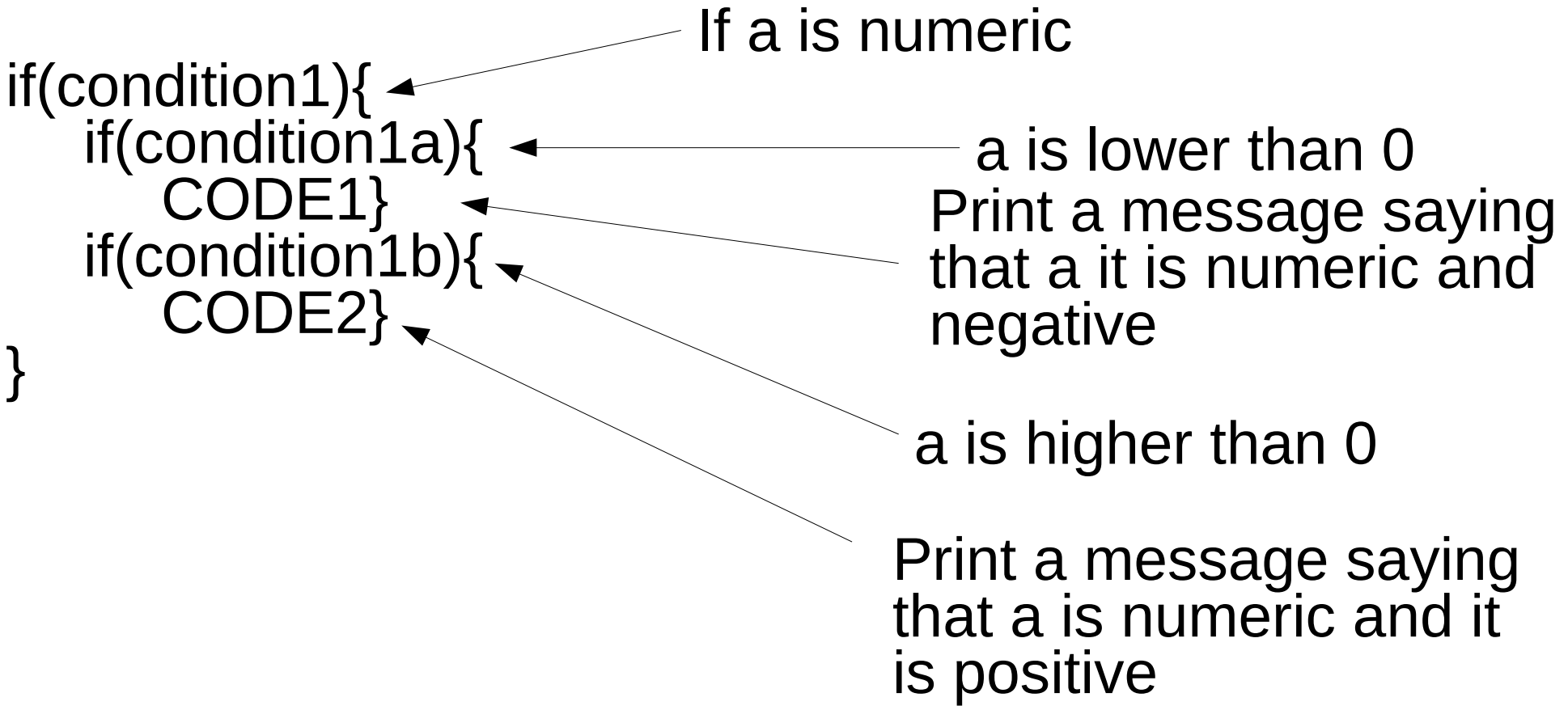
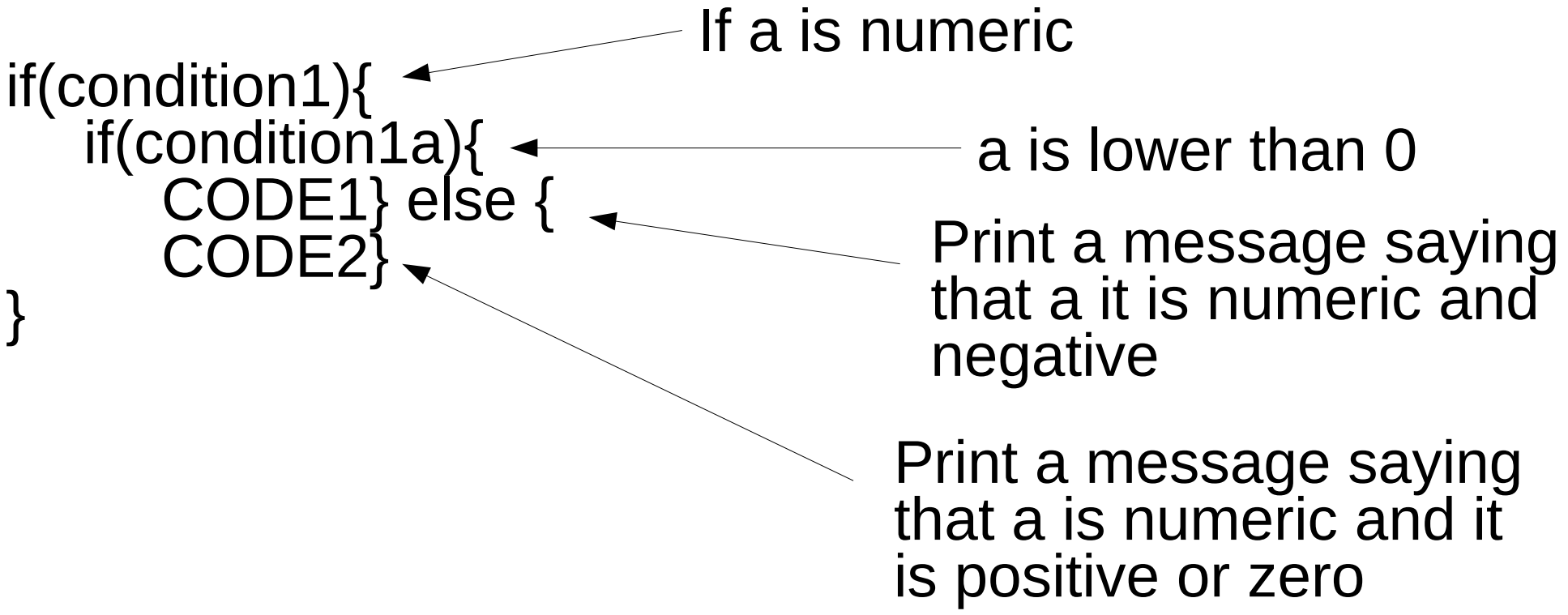

a
Create an object “a”
if(condition1){
code
}
If a is numeric
Print a message saying
that a is numeric


a
Create an object “a”, that can randomly be either a numeric or a character
if(condition1
If a is numeric
& condition2
a is lower than 0
|condition1
If a is numeric
& condition3){
a is higher than 0
code
Print a message saying that a is numeric and it is not zero
}


if(condition1){
if(condition1a){
CODE1}
if(condition1b){
CODE2}
}
If a is numeric
a is lower than 0
Print a message saying that a it is numeric and negative
a is higher than 0
Print a message saying that a is numeric and it is positive


if(condition1){
if(condition1a){
CODE1} else {
CODE2}
}
If a is numeric
a is lower than 0
Print a message saying that a it is numeric and negative
Print a message saying that a is numeric and it is positive or zero

```
if(condition1){
    if(condition1a){
        CODE1} else {
        CODE2}
}
```

If a is numeric

a is lower than 0

Print a message saying that a it is numeric and negative

Print a message saying that a is numeric and it is positive or zero

Another way, using ifelse()

```
if(condition1){
    ifelse(condition1a,CODE1,CODE2)
}
```

Example 1: The logarithm
#==========

1.- Create an object containing a random number that can be either positive (with a probability of 50%) or negative (50%).

2.- Calculate the logarithm of this number, but only if it is numeric and positive

Example 1: The logarithm
#==========

1.- Create an object containing a random number that can be either positive (with a probability of 50%) or negative (50%).

2.- Calculate the logarithm of this number, but only if it is numeric and positive

3.- If the number is negative, provide an error

Example 1: The logarithm
#==========

1.- Create an object containing a random number that can be either positive (with a probability of 50%) or negative (50%).

2.- Calculate the logarithm of this number, but only if it is numeric and positive

3.- If the number is negative, provide an error

4.- Try to provide the result at the end of the code, not in the middle

Example 2: Comparing with mean values
==========

1.- Read the bumpus dataset

2.- Select one element from one numeric variable

3.- Print in the screen if this element is bigger or lower than the average

4.- Print a message saying if this element is below, above of within the 95% distibution*

*remember the 95% distribution is $\overline{X} \pm 1.96\sigma$

Example 3: Tossing a coin
==========

1.- Flip a coin (to get head or tail randomly)

2.- Define your choice

3.- Compare the coin (1) with your choice (2)

3.- If you won, say it

4.- If you lost, also say it

* Also try to provide the result at the end of the code, not in the middle

```
# Example 4: Guessing the number
# ============================
```

1.- Randomly select one number between 9 and 11

2.- Guess the number by using only two logical evaluations

3.- Print your guess

Example 5: rock–paper–scissors
==========

1.- Define your choice (rock, paper, scissors)

2.- Define R choice (randomly)

3.- Provide a message with the result of the game*

4.- Include some advanced errors:
- if your choice is not one of the 3 possibilities
- Do the game case-insensitive

* remember:
rock wins scissors;
paper wins rock;
rock wins scissors