Today

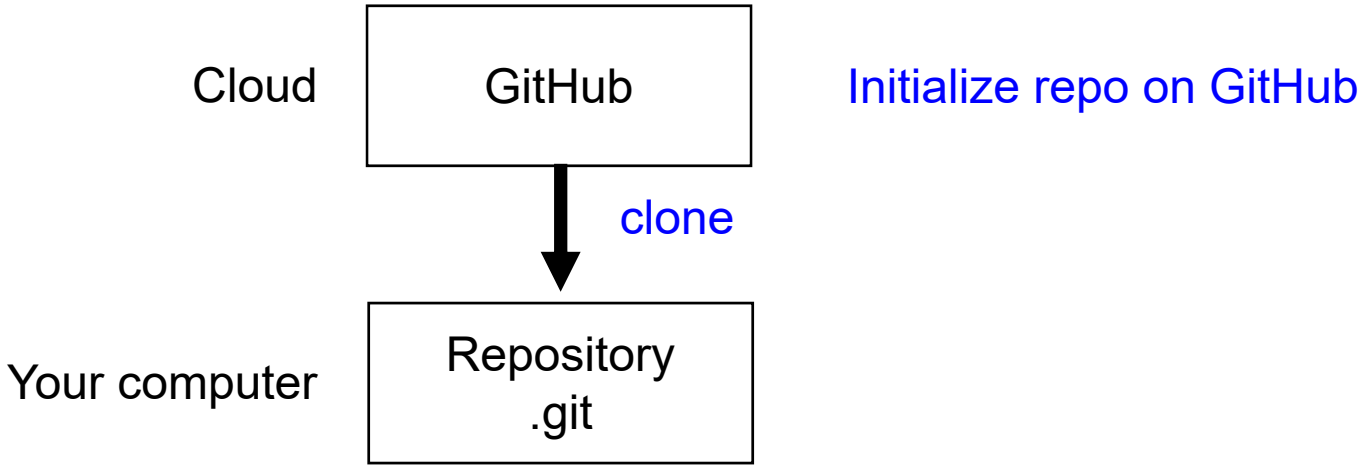
- HW recap & questions
  - Git & GitHub
  - Programming: selection structures
- Programming: repetition structures

Git & GitHub

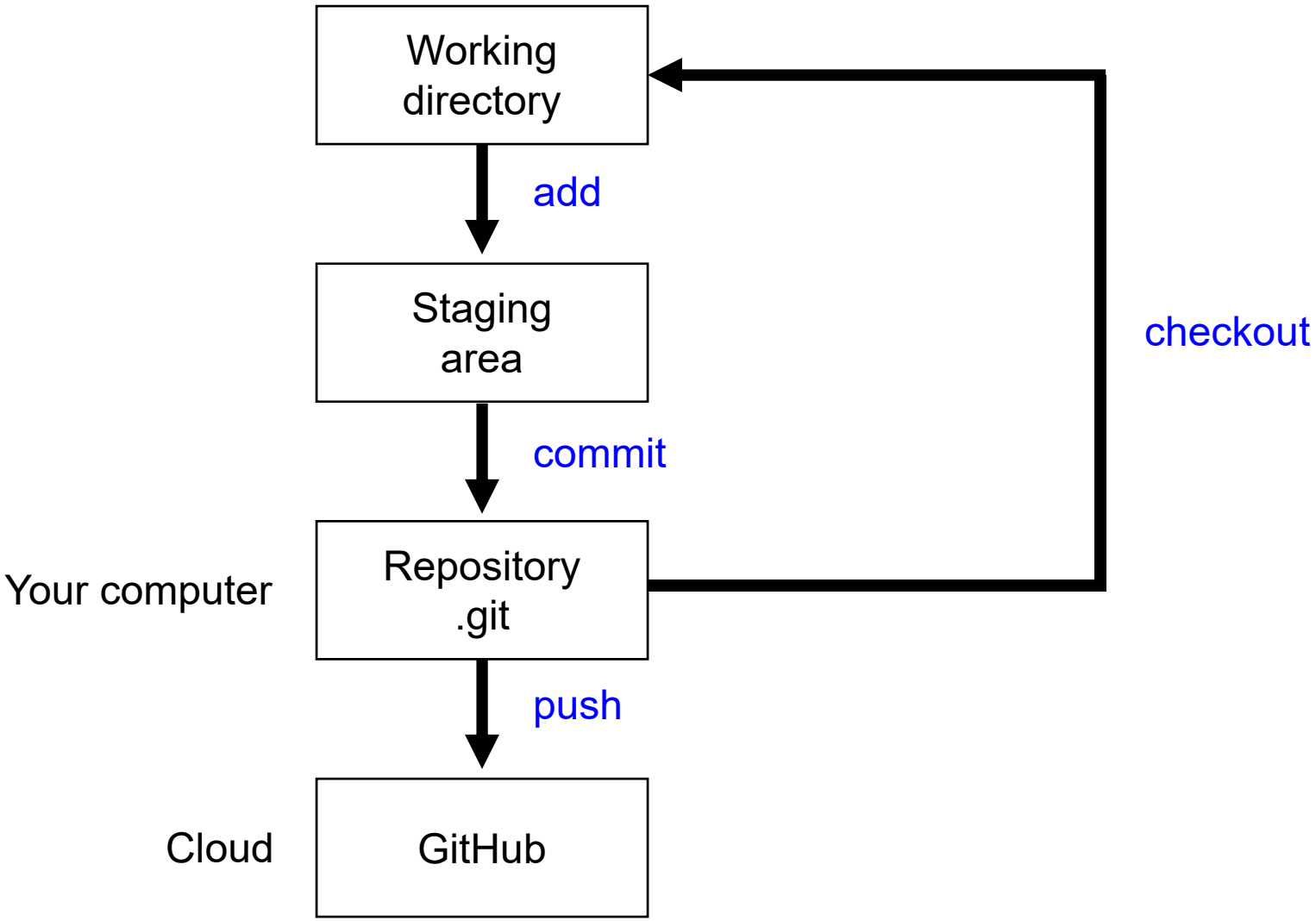
- Git
  - version control software
- GitHub
  - cloud service for storing and collaborating on git repositories

Initialize a Git repo

Github first workflow



Version control workflow



R’s selection structures

if	single selection structure
if/else	double selection structure
if/else if	multiple selection structure

Questions?

Structured programming

- Sequence structure
  - order to perform actions
- Selection structure (conditional, branches)
  - what to do depending on a decision
- Repetition structure (iteration, loops)
  - do something many times

```
?Control #for help in R
```

Structured programming

- Sequence structure
  - order to perform actions
- Selection structure (conditional, branches)
  - what to do depending on a decision
- Repetition structure (iteration, loops)
  - do something many times

```
?Control #for help in R
```

R's repetition structures

- while

```
while ( condition ) {
    expression
}
```

general

- for

```
for ( var in seq ) {
    expression
}
```

specialized

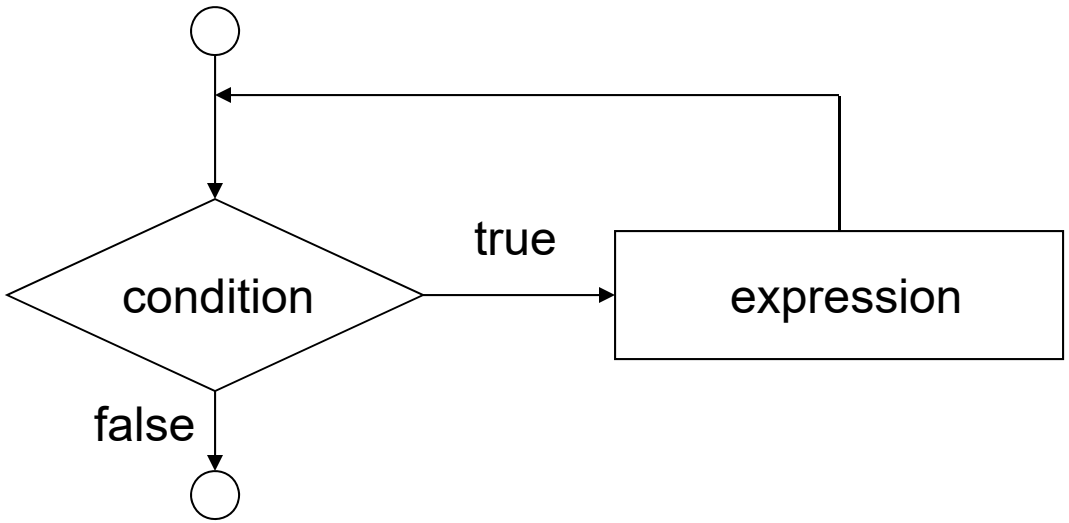
Repetition structures

- Two main types:
- Sentinel controlled repetition
  - number of reps is unknown from the start
  - recognize when the task is finished by testing a condition
- Counter controlled repetition
  - number of reps is known from the start (e.g. repeat 1000 times)

while repetition structure

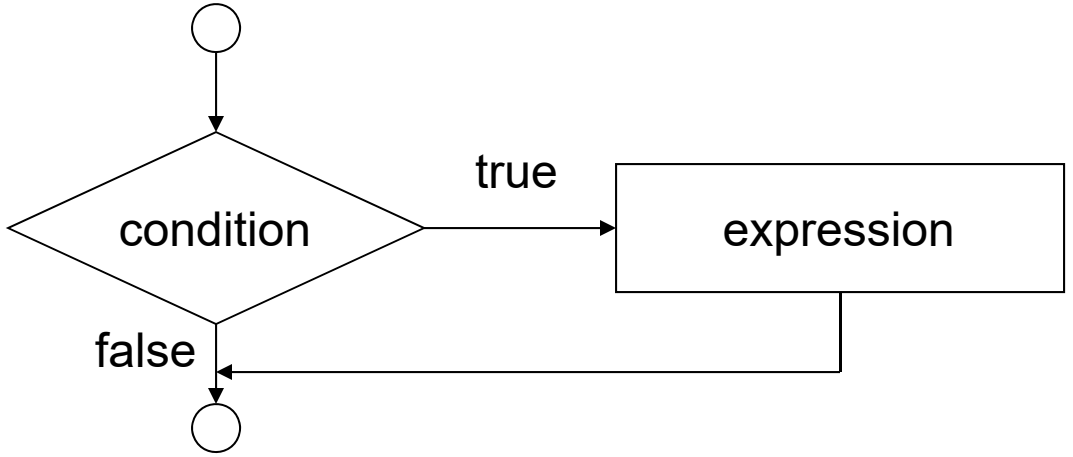
```
while ( condition ) {
    expression1
    expression2
    ...
}
```

Good programming practice:
brace, space, indent

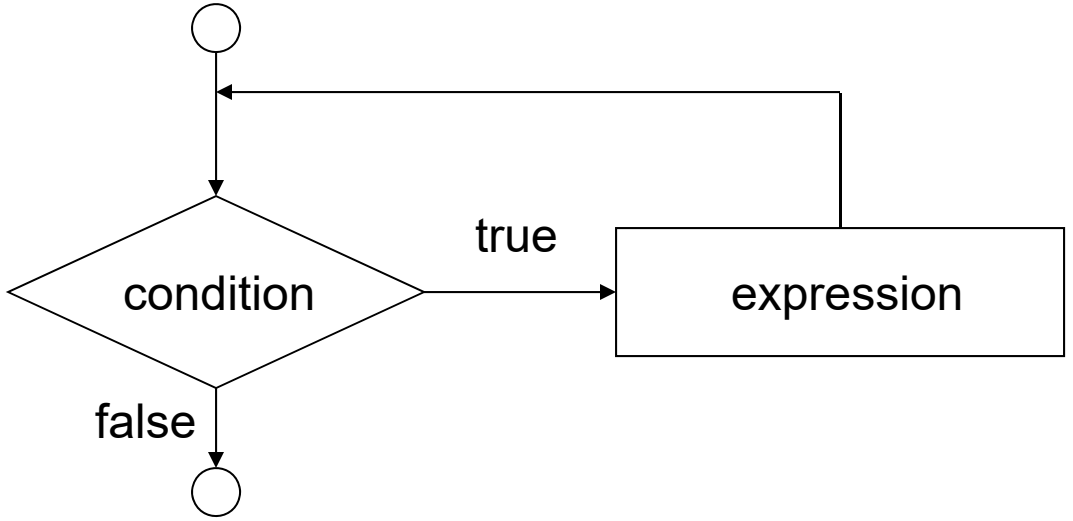


if and while are fundamental

if



while

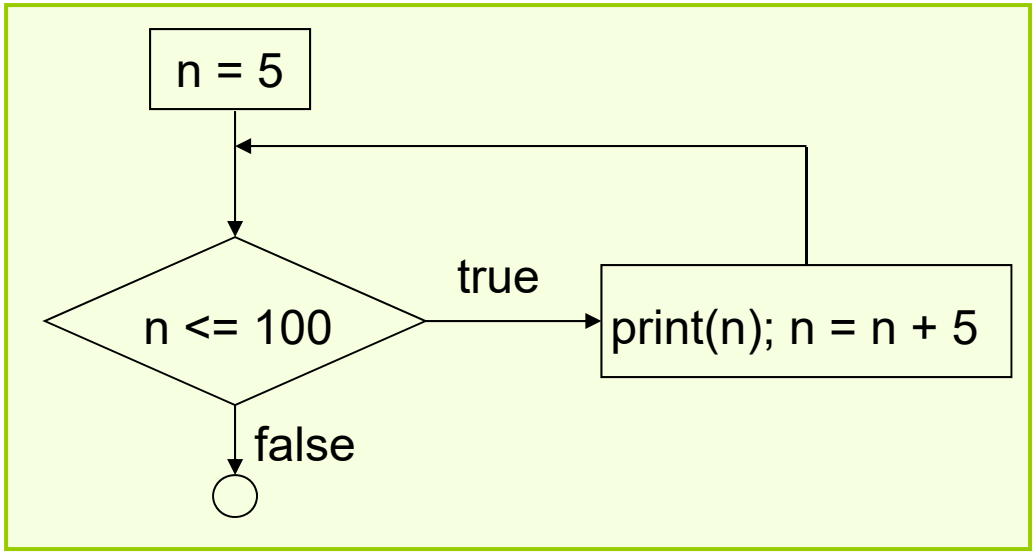


Both sentinel-controlled and counter-controlled repetition can be done with while

while repetition structure

- Sentinel controlled repetition
- e.g. print every fifth integer up to 100

```
n <- 5
while ( n <= 100 ) {
    print(n)
    n <- n + 5
}
```



Algorithms

Often have three phases:

1) Initialization phase

   e.g. setting up vectors, matrices, initial values

2) Processing phase

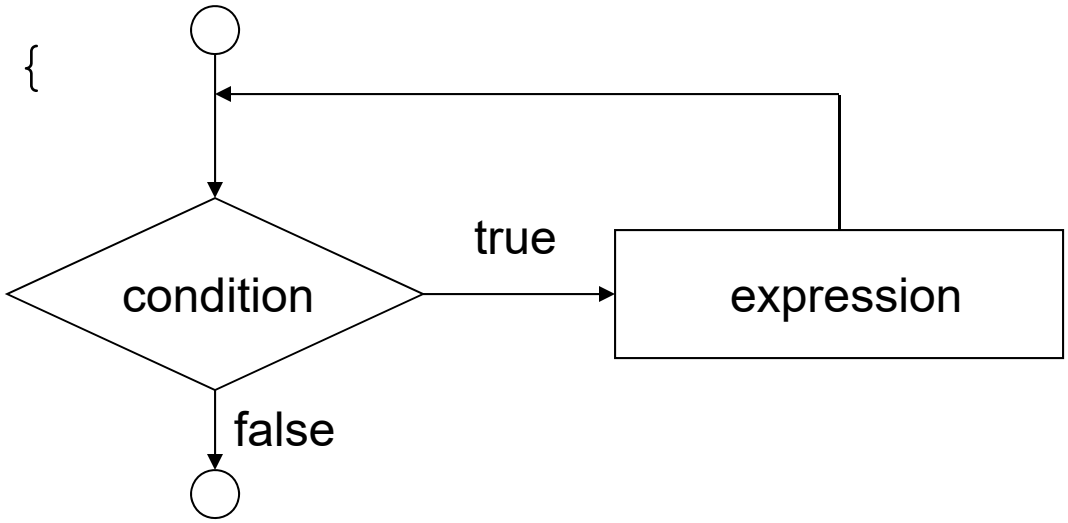
   e.g. calculations, manipulations

3) Termination phase

   e.g. printing or graphing the result

Exercise: while, sentinel control

```
while (condition) {
    expression1
    expression2
    ...
}
```



Exercise: sentinel controlled repetition

Using the `while` structure, find the first number greater than 1000 that is a power of 2, and print the result. You can't use the "^" operator.

Hint: start by initializing `number <- 2` and then repeatedly multiply by 2. The first four powers of 2 are 2, 4, 8, 16.

Flowchart and pseudocode first!!

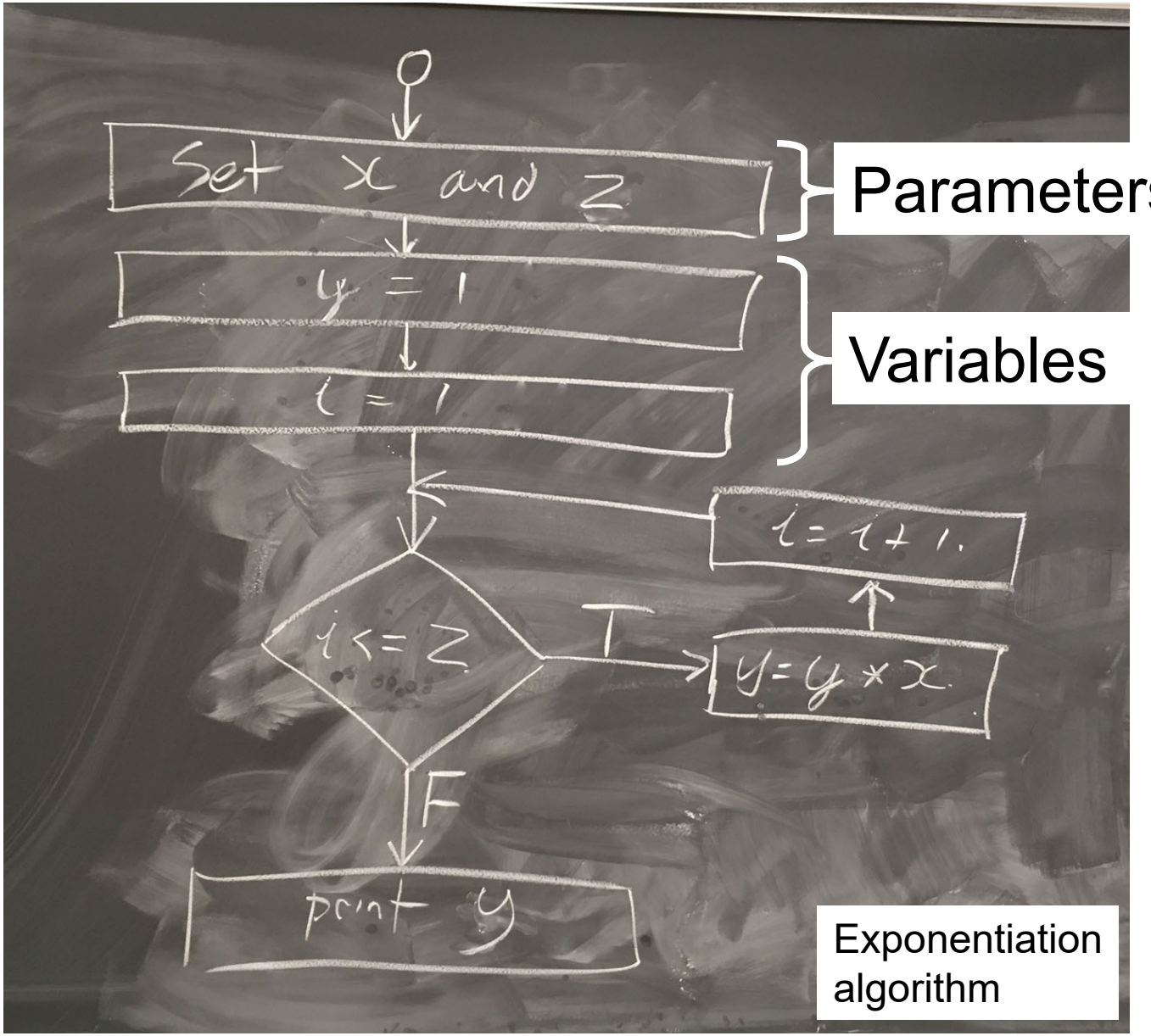
while repetition structure

- Counter controlled repetition

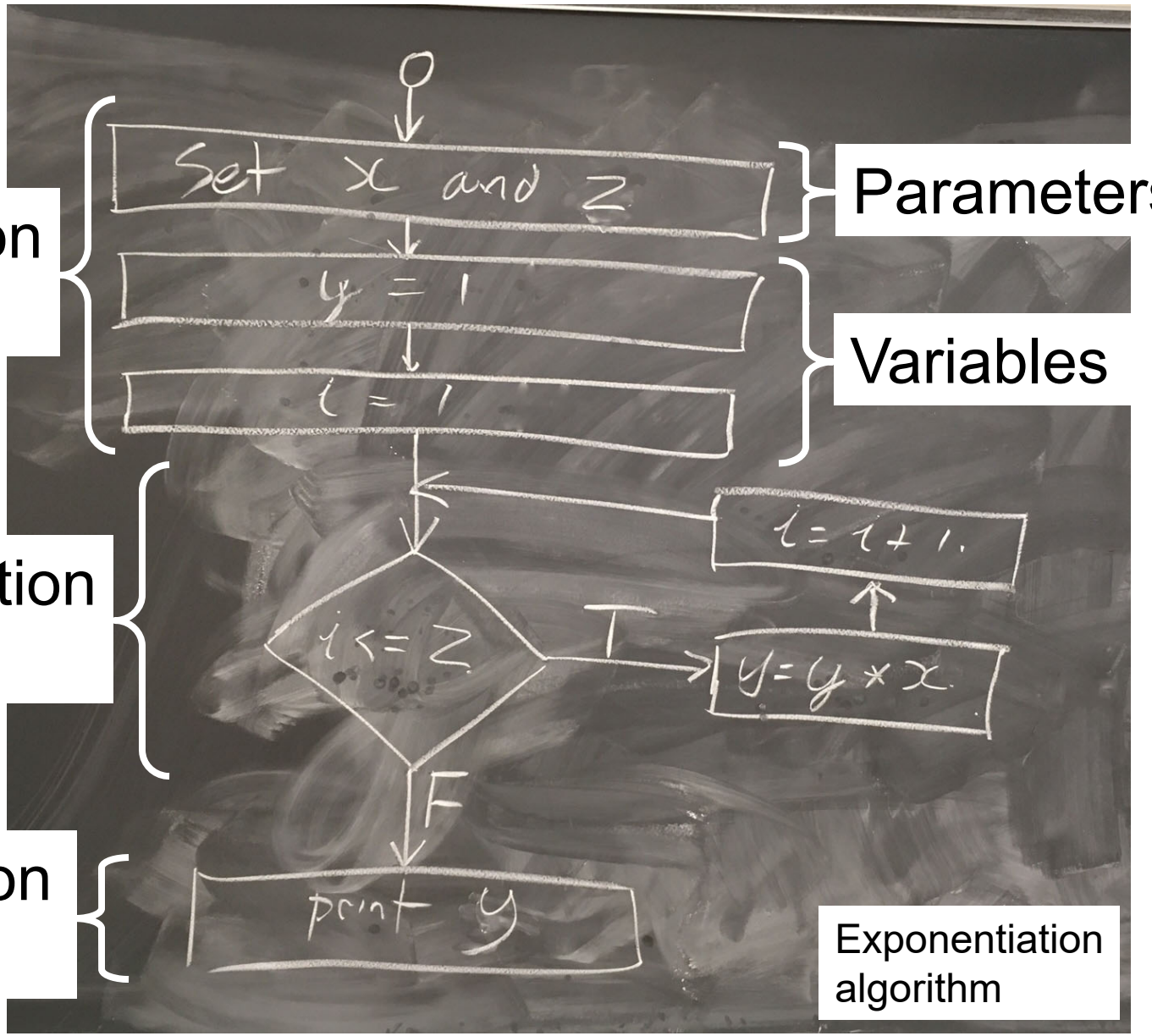
```
i <- 1 #initialize the counter
while ( i <= n ) {
    expression
    i <- i + 1
}
```

Exercise: counter controlled repetition
Using the `while` structure, write a program where you can enter any real number x and positive integer z and the program will calculate y = x^z. BUT you cannot use the "^" operator. Check your result using R's native exponentiation. Flowchart and pseudocode first!!

Common parts of an algorithm



Common parts of an algorithm



Initialization phase

Parameters

Variables

Calculation phase

Termination phase