Structured programming

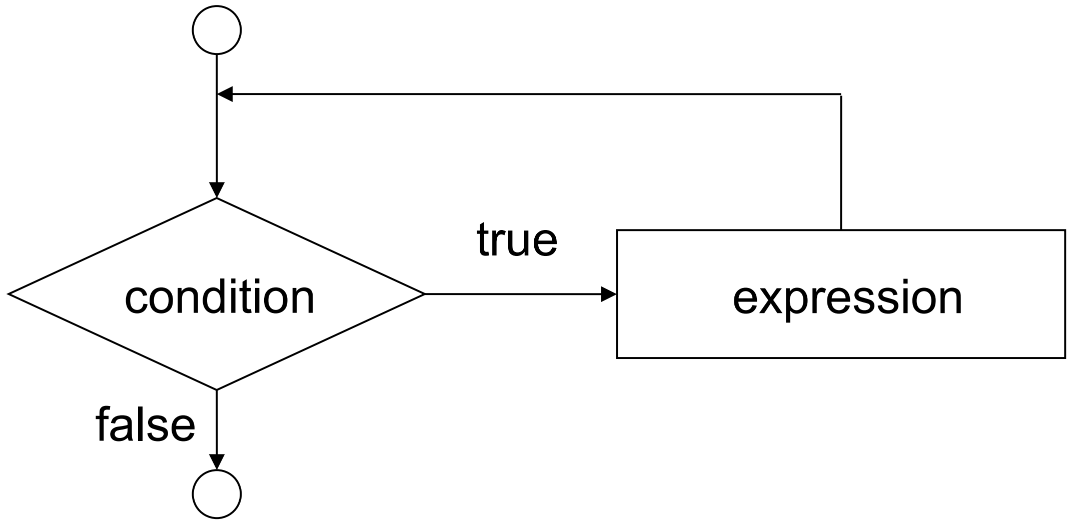
- Sequence structure
  - order to perform actions
- Selection structure (conditional, branches)
  - what to do depending on a decision
- Repetition structure (iteration, loops)
  - do something many times

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

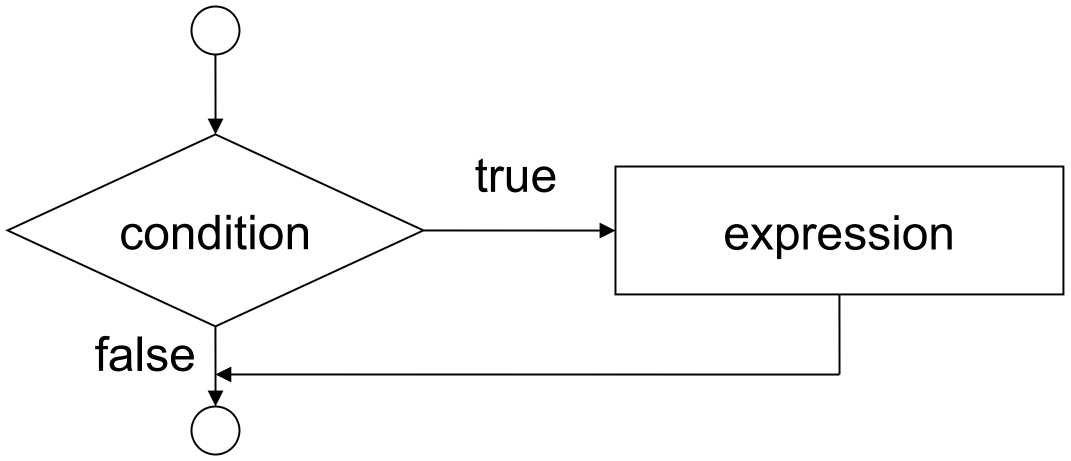
while repetition structure

```
while condition
    expression
```

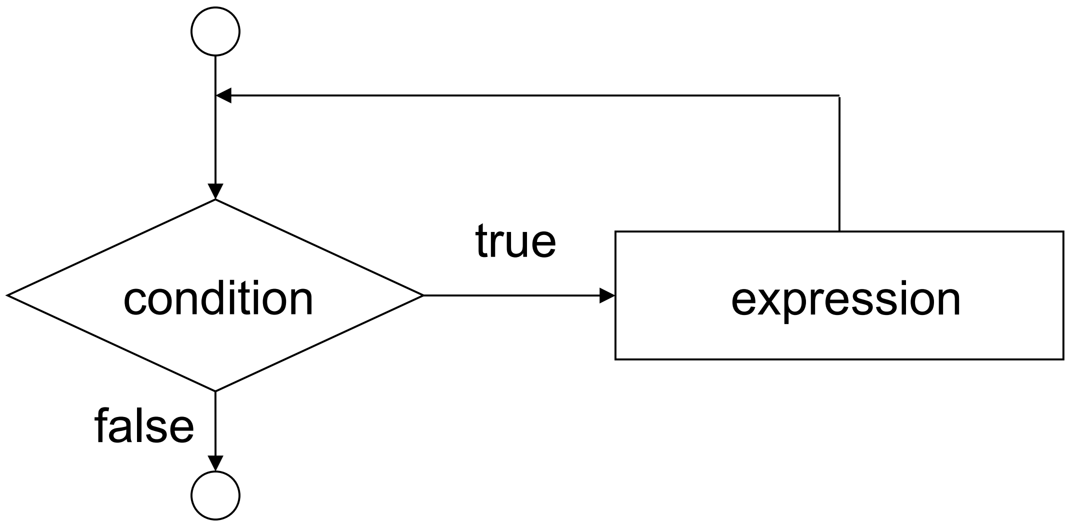


if and while are fundamental

if



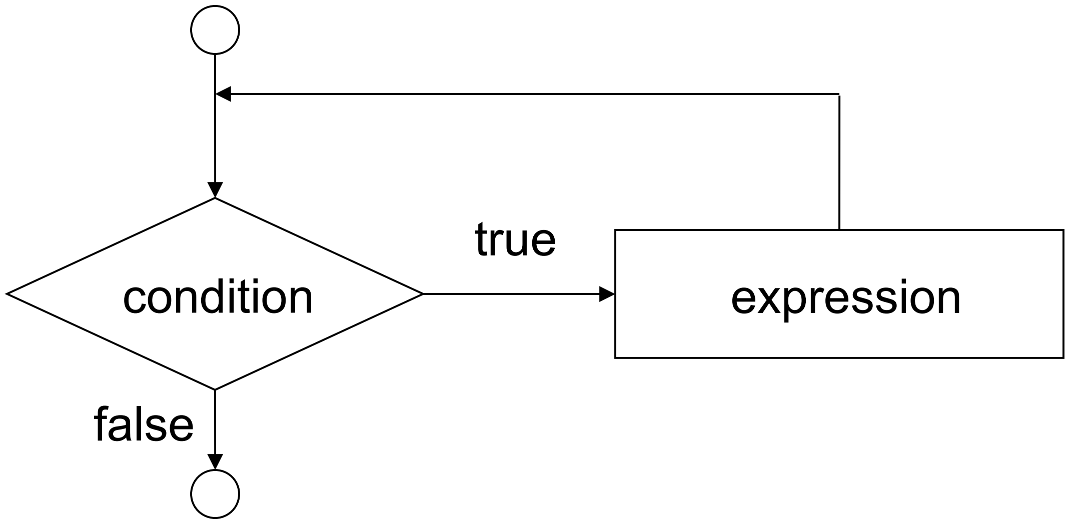
while



R's while repetition structure

```
while ( condition ) {
    expression_1
    expression_2
    ...
}
```

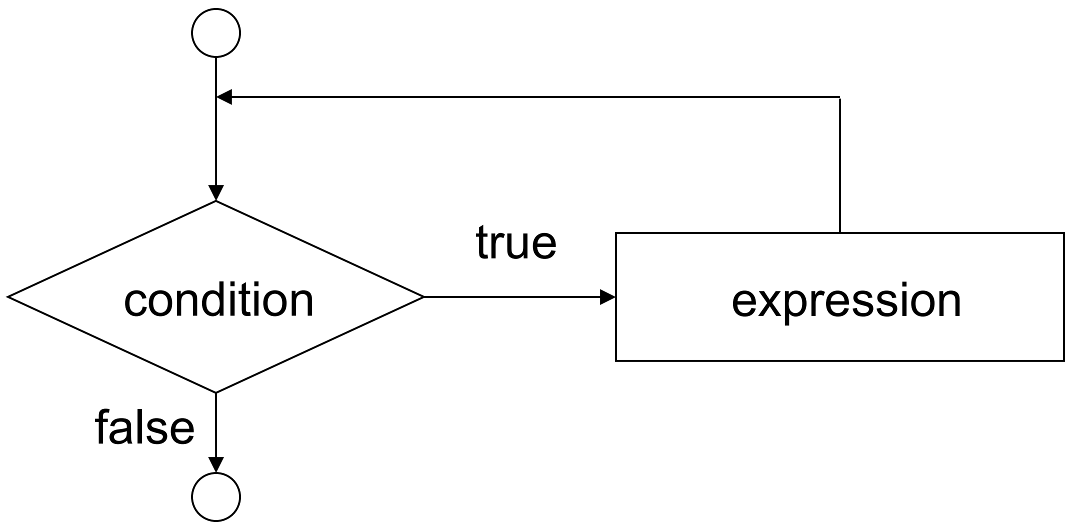
Good programming practice: brace, space, indent



R

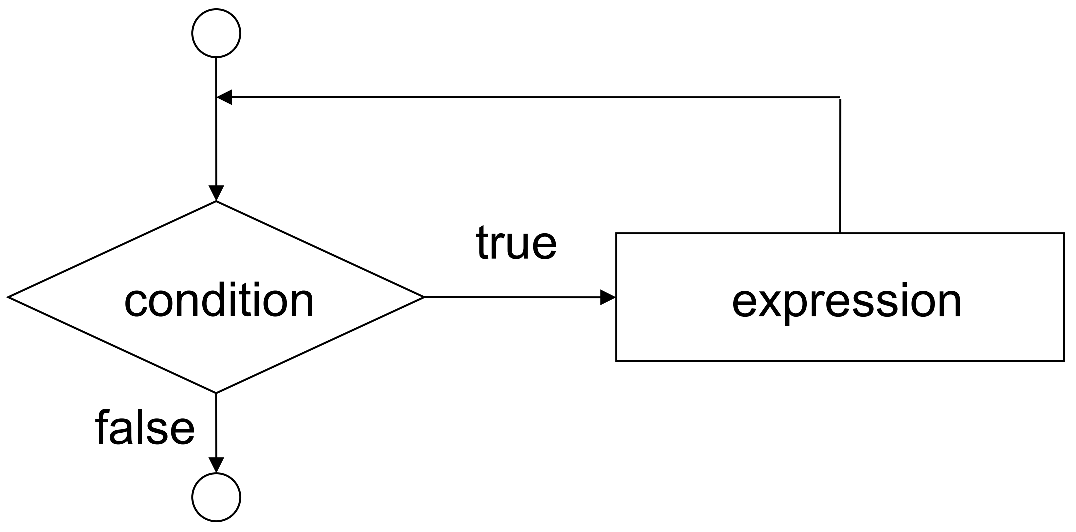
C's while repetition structure

```
while ( condition ) {
    expression_1;
    expression_2;
    ...
}
```



Py's while repetition structure

```
while condition:
    expression_1
    expression_2
    ...
```



Repetition structures

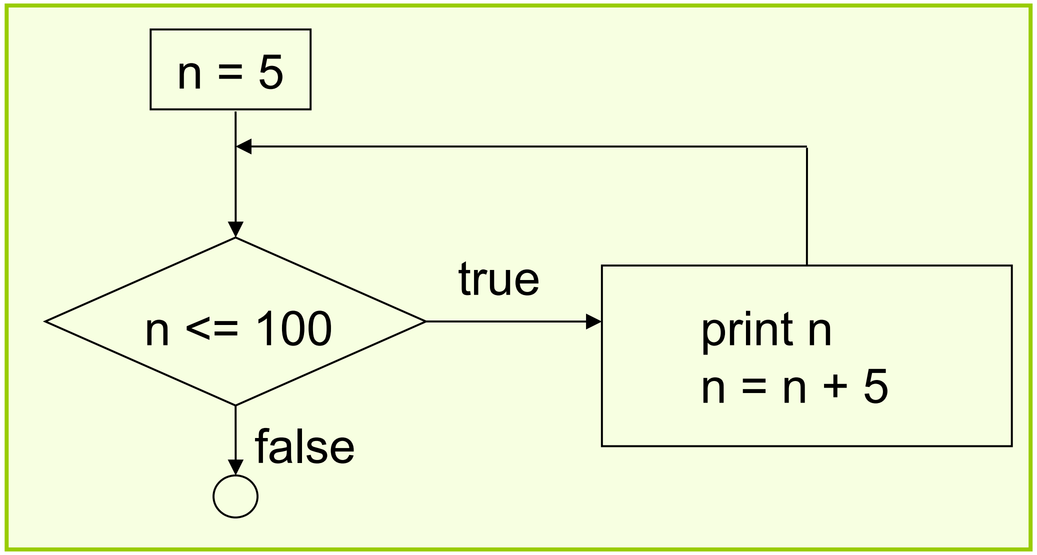
- Two main types:
- Sentinel controlled repetition
  - number of reps is unknown from the start
  - recognize when the task is finished by testing a condition
- Counter controlled repetition
  - number of reps is known from the start (e.g. repeat 1000 times)

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

while repetition structure

- Sentinel controlled repetition
- e.g. print every 5th positive integer up to 100

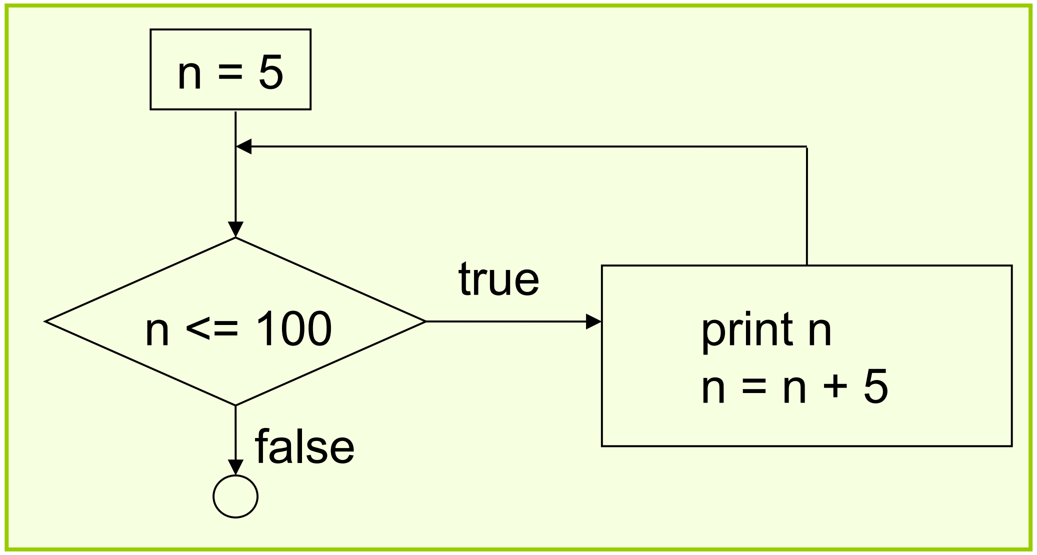
```
set n to 5
while n <= 100
    print n
    add 5 to n
```



while repetition structure

- Sentinel controlled repetition
- e.g. print every 5th positive integer up to 100

```
set n to 5
while n <= 100
    print n
    add 5 to n
```



What if we instead initialize n to 0?

Algorithms

Often have three phases:

1) Initialization phase

   e.g. setting up initial values and data structures

2) Processing phase

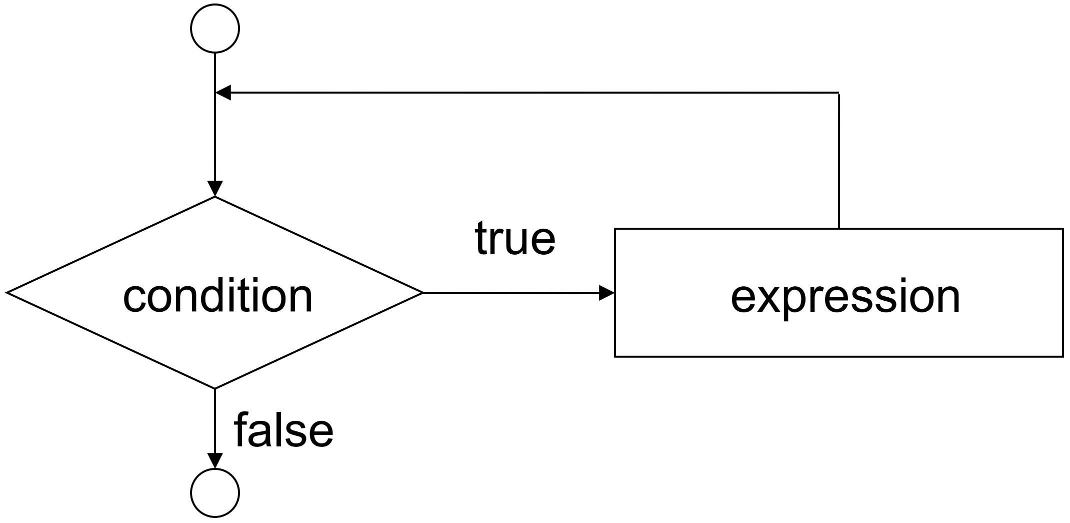
   e.g. calculations, manipulations, storing results

3) Termination phase

   e.g. printing or graphing the result

Exercise: while, sentinel control

```
while condition
    expression_1
    expression_2
    ...
```



Exercise: sentinel controlled repetition

A population starts with 2 individuals. Each generation, it doubles in size. What is the population size the first time the population exceeds 1000 individuals?

Use the `while` structure. The algorithm should finish by printing the answer. You can't use an exponentiation operator.

Hint: what are the first 4 numbers?

Flowchart and pseudocode first! Then Python.

while repetition structure

- Counter controlled repetition

counter

number of repetitions

```
i = 1
while i <= n
    expression_1
    expression_2
    ...
    i = i + 1
```

increment the counter

while repetition structure

- Counter controlled repetition

```
i = 1                       counter
while i <= n                n repetitions
    ... expressions
    i = i + 1               increment counter
```

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

while repetition structure

- Counter controlled repetition

counter

number of repetitions

```
i = 1
while i <= n
    expression_1
    expression_2
    ...
    i = i + 1
```

increment the counter