

# LOOPS IN DELPHI

*While, for..do and repeat..until. Why we test before or after — and why that choice changes the program.*

## 1. What a loop is

A loop repeats a job. The three Delphi loops all repeat. They differ in when they ask “should we do this again?”

## 2. Two places you can ask that question

Test at the top (before the body) — While and For do this.

The program asks first. Only if the answer allows it does the body run. If the answer is already “no” at the start, the body is skipped completely.

Test at the bottom (after the body) — Repeat does this.

The program does the job first, then asks. So the job always happens at least once. The question only decides whether to go around again.

## 3. Why that position matters — a picture

Think of a security gate.

- While / For = the guard stands in front of the door. If you do not qualify, you never enter the room.
- Repeat = you are already in the room. After you try the task, the guard asks whether you must try again.

That is not decoration. It changes real programs.

## 4. A concrete “why it matters” example

Job: keep asking for a PIN until it is 4 digits.

Repeat (test after) — natural:

```
repeat
  sPin := InputBox('PIN', 'Enter 4 digits', '');
until Length(sPin) = 4;
```

The box must appear once. You cannot test the PIN before the user has typed it. Test-after is the honest shape.

While (test before) — you need a starter value first:

```
sPin := '';
while Length(sPin) <> 4 do
  sPin := InputBox('PIN', 'Enter 4 digits', '');
```

While cannot ask “is the PIN long enough?” until some value exists. So you invent an empty string just to get through the first test. It works, but the extra line is only there because the test sits too early.

Other way around — a file that might be empty:

While is right. If there is no next line, you must not process a dummy line. Test first, enter the body only when there is something to work on.

```
while not Eof(tFile) do
begin
  ReadLn(tFile, sLine);
  redOutput.Lines.Add(sLine);
end;
```

If you used Repeat on an empty file, the body would still run once and try to read a line that is not there. That is why “at least once” can be wrong.

## 5. For..do — counted, test built in

Use For when you already know the start and the end: print 1 to 10, visit every item in a list of 20 names. Delphi counts. You do not write the test yourself.

```
for i := 1 to 10 do
  redOutput.Lines.Add(IntToStr(i));
```

The test is still “before each pass”: if you write for i := 5 to 3 do the body never runs, because 5 is already past 3. For can also run zero times.

## 6. While — keep going WHILE this is True

Use While when you do not know the count, and skipping the body is allowed.

```
i := 1;
while i <= 10 do
begin
  redOutput.Lines.Add(IntToStr(i));
  Inc(i);
end;
```

You must change the variable yourself (Inc, read the next record). If you forget, the test stays True forever.

## 7. Repeat..until — stop WHEN this becomes True

Until is the opposite wording of While. While says keep going while True. Until says stop when True.

```
i := 1;
repeat
  redOutput.Lines.Add(IntToStr(i));
  Inc(i);
until i > 10;
```

Same 1-to-10 job. Notice <= 10 on While and > 10 on Until. Flip the words and you flip the test.

## 8. Side-by-side

|                    | for..do            | while                | repeat..until        |
|--------------------|--------------------|----------------------|----------------------|
| Test sits...       | At the top (count) | At the top           | At the bottom        |
| Can skip the body? | Yes                | Yes                  | No — always once     |
| Know the count?    | Yes                | No                   | No                   |
| Words mean         | Walk A to B        | While True, continue | Until True, stop     |
| Use when           | 1 to n, lists      | Maybe nothing to do  | Must try, then check |

## 9. How to choose

- I know how many times, or I must visit every number from A to B → for.
- I might have nothing to do (empty file, stock already 0) → while. Test first so we can skip.
- I must do the action once before I can judge it (ask, roll a die, show a menu) → repeat. Test after so we cannot skip the first try.

- 

## 10. One-line summaries

Test before means “only enter if it is already allowed” — the body may run zero times.

Test after means “do it, then decide” — the body always runs once.

That single difference is why While and Repeat are not interchangeable, even when they can be forced to print the same 1 to 10.