

Description

for is equivalent to

```
    exp1
while (exp2) {
    stmt(s)
    exp3
}
```

stmt(s) is executed zero or more times. The loop continues as long as *exp2* is not equal to zero.

Syntax

```
for (exp1; exp2; exp3) stmt
```

```
for (exp1; exp2; exp3) {
    stmts
}
```

where *exp₁* and *exp₃* are optional, and *exp₂* must evaluate to a real scalar.

Remarks and examples

To understand for, enter the following program

```
function example(n)
{
    for (i=1; i<=n; i++) {
        printf("i=%g\n", i)
    }
    printf("done\n")
}
```

and run `example(3)`, `example(2)`, `example(1)`, `example(0)`, and `example(-1)`.

Common uses of for include

```
for (i=1; i<=rows(A); i++) {
    for (j=1; j<=cols(A); j++) {
        ...
    }
}
```

Also see

[M-2] **break** — Break out of for, while, or do loop

[M-2] **continue** — Continue with next iteration of for, while, or do loop

[M-2] **do** — do ... while (exp)

[M-2] **Semicolons** — Use of semicolons

[M-2] **while** — while (exp) stmt

[M-2] **Intro** — Language definition

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on [citing Stata documentation](#).

