

intro — Language definition

[Contents](#)[Description](#)[Remarks and examples](#)[Also see](#)

Contents

[M-2] Entry	Description
-------------	-------------

Syntax

syntax	Grammar and syntax
subscripts	Use of subscripts
reswords	Reserved words
comments	Comments

Expressions & operators

exp	Expressions
op_assignment	Assignment operator
op_arith	Arithmetic operators
op_increment	Increment and decrement operators
op_logical	Logical operators
op_conditional	Conditional operator
op_colon	Colon operators
op_join	Row- and column-join operators
op_range	Range operators
op_transpose	Conjugate transpose operator
op_kronecker	Kronecker direct-product operator

Declarations & arguments

declarations	Declarations and types
optargs	Optional arguments
struct	Structures
class	Object-oriented programming (classes)
pragma	Suppressing warning messages
version	Version control

Flow of control

if	<code>if (<i>exp</i>) ... else ...</code>
for	<code>for (<i>exp1</i>; <i>exp2</i>; <i>exp3</i>) <i>stmt</i></code>
while	<code>while (<i>exp</i>) <i>stmt</i></code>
do	<code>do ... while (<i>exp</i>)</code>
break	Break out of for, while, or do loop
continue	Continue with next iteration of for, while, or do loop
goto	<code>goto <i>label</i></code>
return	return and <code>return(<i>exp</i>)</code>

Special topics

semicolons	Use of semicolons
void	Void matrices
pointers	Pointers
ftof	Passing functions to functions

Error codes

errors	Error codes
---------------	-------------

Description

This section defines the Mata programming language.

Remarks and examples

[stata.com](#)

[M-2] [syntax](#) provides an overview, dense and brief, and the other sections expand on it.

Also see [M-1] [intro](#) for an introduction to Mata.

Also see

[M-0] [intro](#) — Introduction to the Mata manual