MATHEMATICA OPERATIONS

`Do[`*expr*`, {`i_{max}`}]`
evaluates *expr* i_{max} times.

`Do[`*expr*`, {`*i*`, `i_{max}`}]`
evaluates *expr* with the variable *i* successively taking on the values 1 through i_{max} (in steps of 1).

`Do[`*expr*`, {`*i*`, `i_{min}`, `i_{max}`}]`
starts with $i = i_{min}$.

`Do[`*expr*`, {`*i*`, `i_{min}`, `i_{max}`, `*di*`}]`
uses steps *di*.

`Do[`*expr*`, {`*i*`, {`i_1`, `i_2`, ...}}]`
uses the successive values i_1, i_2, ...

`Do[`*expr*`, {`*i*`, `i_{min}`, `i_{max}`}, {`*j*`, `j_{min}`, `j_{max}`}, ...]`
evaluates *expr* looping over different values of *j*, etc. for each *i*.

`If[`*condition*`, `*t*`, `*f*`]`
gives *t* if *condition* evaluates to `True`, and *f* if it evaluates to `False`.

`If[`*condition*`, `*t*`, `*f*`, `*u*`]`
gives *u* if *condition* evaluates to neither `True` nor `False`.

`For[`*start*`, `*test*`, `*incr*`, `*body*`]`
executes *start*, then repeatedly evaluates *body* and *incr* until *test* fails to give `True`.

`While[`*test*`, `*body*`]`
evaluates *test*, then *body*, repetitively, until *test* first fails to give `True`.

`EvenQ[`*expr*`]`
gives `True` if *expr* is an even integer, and `False` otherwise.

`OddQ`[*expr*]
gives `True` if *expr* is an odd integer, and `False` otherwise.

`PrimeQ`[*expr*]
yields `True` if *expr* is a prime number, and yields `False` otherwise.