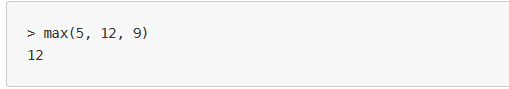
**TERRAFORM FUNCTIONS**  
  
The Terraform language includes a number of built-in functions that you can call from within expressions to transform and combine values. The general syntax for function calls is a function name followed by comma-separated arguments in parentheses: max (5, 12, 9).  
  
The Terraform language does not support user-defined functions, and so only the functions built in to the language are available for use. The navigation for this section includes a list of all of the available built-in functions.

You can experiment with the behavior of Terraforms built-in functions from the Terraform expression console, by running [the terraform console command](https://www.terraform.io/docs/cli/commands/console.html):

  
  
  
The examples in the documentation for each function use console output to illustrate the result of calling the function with different parameters.  
  
Terraform Functions types  
  
**Numeric Functions**

**String Functions**

**Collection Functions**

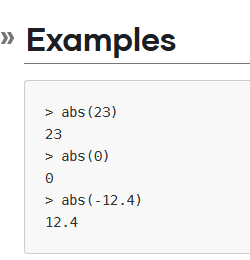
**Encoding Functions**

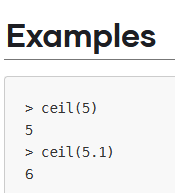
**Filesystem Functions**

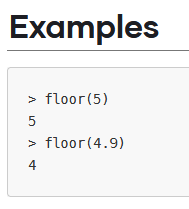
**Date and Time Functions**

**Hash and Crypto Functions**

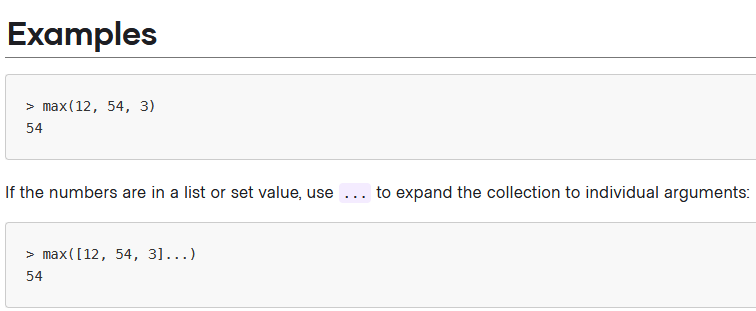
**IP Network Functions**

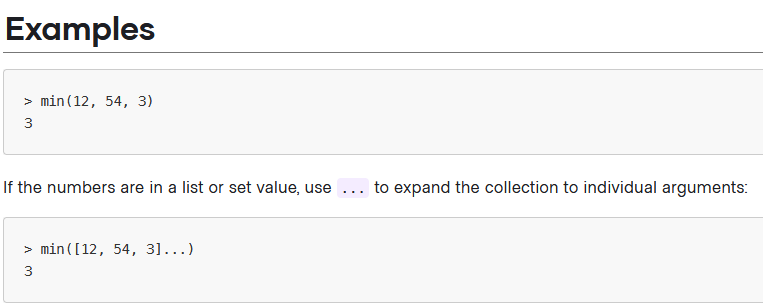
**Type Conversion Functions.  
  
  
  
Numeric Functions:  
  
  
abs:  
  
abs** return the absolute value of the given number. In other words, if the number is zero or positive then it is returned as-is, but if it is negative then it is multiplied by -1 to make it positive before returning it.

**Ceil:  
  
ceil** returns the closest whole number that is greater than or equal to the given value, which may be a fraction.

**Floor:  
  
floor** returns the closest whole number that is less than or equal to the given value, which may be a fraction.

**Log:   
  
log** returns the logarithm of a given number in a given base.log and ceil can be used together to find the minimum number of binary digits required to represent a given number of distinct values:

**Max :**max takes one or more numbers and returns the greatest number from the set.  
  
  
  
  
  


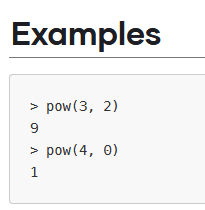
**Min:**min takes one or more numbers and returns the smallest number from the set.  
  


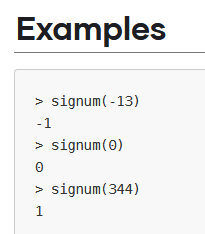
**Parseint:**parseint parses the given string as a representation of an integer in the specified base and returns the resulting number. The base must be between 2 and 62 inclusive.

All bases use the Arabic numerals 0 through 9 first. Bases between 11 and 36 inclusive use case-insensitive Latin letters to represent higher unit values. Bases 37 and higher use lowercase Latin letters and then uppercase Latin letters.

If the given string contains any non-digit characters or digit characters that are too large for the given base, then parseint will produce an error.

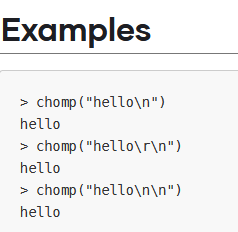


**Pow :**pow calculates an exponent, by raising its first argument to the power of the second argument.  
  


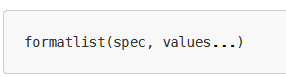
**Signum :**signum determines the sign of a number, returning a number between -1 and 1 to represent the sign.  
  
 **String Functions**

**Chomp:**chomp removes newline characters at the end of a string.

This can be useful if, for example, the string was read from a file that has a newline character at the end.



**Format :**format produces a string by formatting a number of other values according to a specification string. It is similar to the printf function in C, and other similar functions in other programming languages.

**Formatlist :**formatlist produces a list of strings by formatting a number of other values according to a specification string.The specification string uses [the same syntax as format](https://www.terraform.io/docs/language/functions/format.html#specification-syntax).

The given values can be a mixture of list and non-list arguments. Any given lists must be the same length, which decides the length of the resulting list.

The list arguments are iterated together in order by index, while the non-list arguments are used repeatedly for each iteration. The format string is evaluated once per element of the list arguments.



**Indent :**indent adds a given number of spaces to the beginnings of all but the first line in a given multi-line string.  
