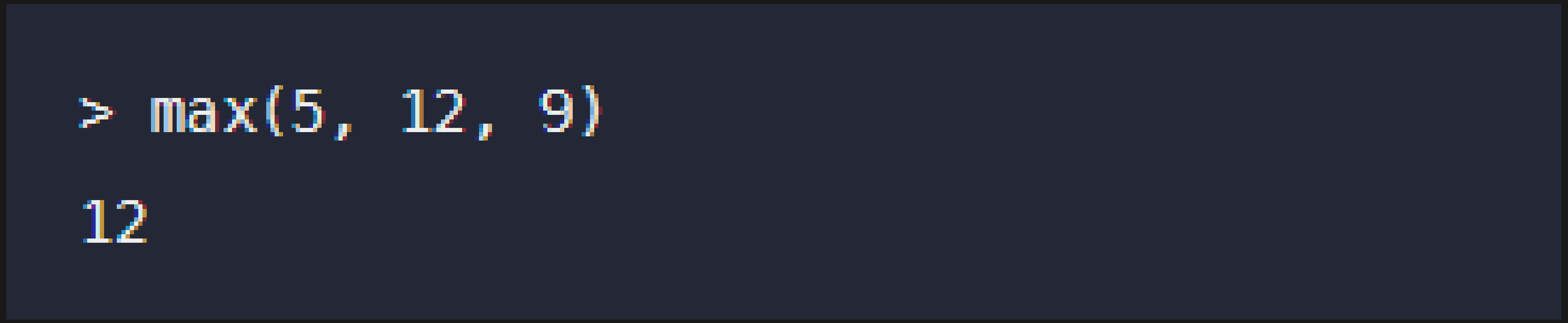
Terraform Common Functions

HashiCorp
Terraform

Common Functions

- Terraform provides several built-in functions which covers a variety of use cases
  - Currently, no possible way to create your own functions
  - Some functions are more commonly used than others - most likely won't touch 90%
- Functions perform some complex operation and returns an output
  - Functions are called by their name, and providing arguments inside the parenthesis
  - max(5, 12, 9) - the max() function takes in any number of arguments, and returns the maximum value
- View Terraform documentation to get full list of supported functions (there's a lot!)



```
> max(5, 12, 9)
12
```

HashiCorp Terraform

Common Functions

- Some of the more common and useful functions:
    - Numeric - max, min, parseInt
    - String - regex, trim, split, join, substr
    - Collection - concat, contains, keys, length, list, lookup, range, coalesce
    - Filesystem - abspath, dirname, file
    - Network - cidrsubnet
    - Conversion - tolist, tomap, toset
- Always refer to documentation, do not try to memorize all the common functions and how they work
    - There's too many of them!

HashiCorp
Terraform

Numeric Functions

- Numeric functions are least commonly used, and deal with manipulating numeric values
- Common functions include min, max, and parseInt
- min(...args) - receives any number of numeric values, and returns the minimum
- max (...args) - receives any number of numeric values, and returns the maximum
- parseInt(string value, number base) - Receives a string and a numer representing the base, and returns the integer version of that string
  - parseInt("100", 10) -> returns 100 as a number
  - Base numbers are represented between 2 and 62

HashiCorp
Terraform

Collection Functions

- Collection functions provide additional logic working with array or object/map values
- contains(list, value) - given a list of values & a value you want to search for, return a true/false if that value is found in the list
    - Useful for constructing conditional logic - If value is contained in list, perform x, otherwise y
- length(list) - given a list of values, returns the length of the list as a number
    - Useful for conditionally creating resources - if the variable is a list & contains values, create length(list) of that resource
- lookup(map, key, default) - given a map, a key to search for, and a value if that key is not found, return the key within the map if found, else return the default value
- concat(...list) - given two or more lists, combine them into a single list and return it

Conversion Functions

- Conversion functions aid in transforming one datatype to another
- tolist(set) - given an argument, convert it as a list
    - Usually provide a set datatype as the argument, to turn it into a list
- toset(list) - given an argument, convert it as a set
    - Usually provide a list datatype as the argument, and turn it into a set
    - Useful for creating a group of resources by looping with for_each

HashiCorp Terraform