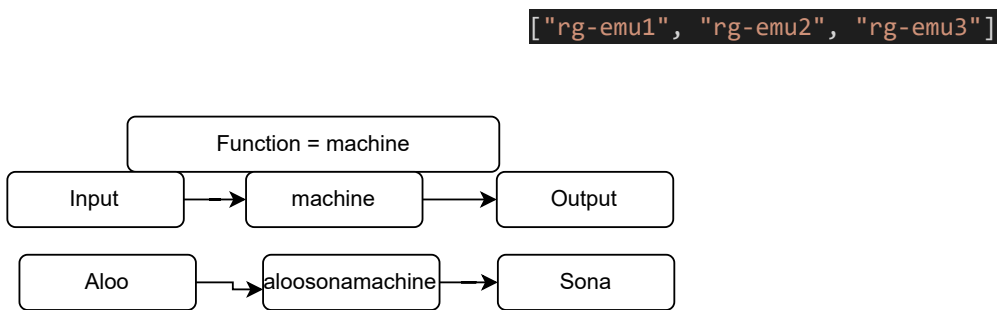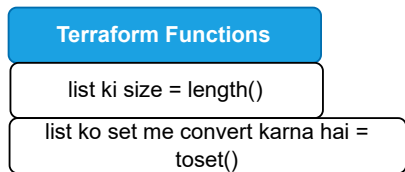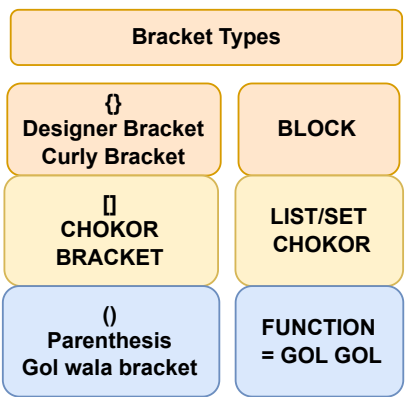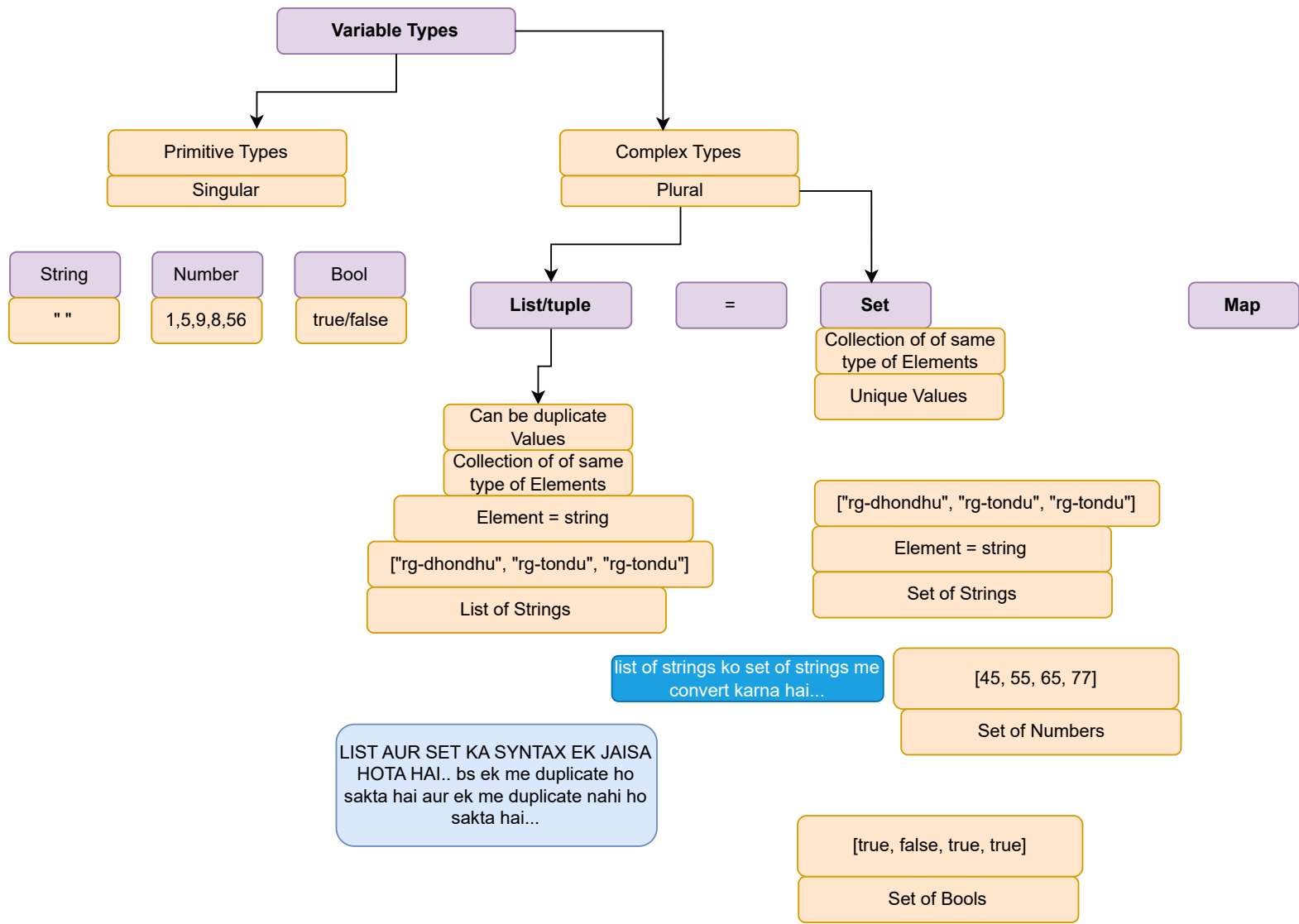**Padna kya padega foreach ke lie...**

FOR EACH ACCEPTS = SET OF STRINGS AND MAP

2. Foreach ko Set ke saath chalana hai aur map ke saath chala ke dekhna hai...

Variable Types

Primitive Types

Singular

Complex Types

Plural

String	Number	Bool
" "	1,5,9,8,56	true/false

List/tuple

=

Set

Collection of of same type of Elements

Unique Values

Can be duplicate Values

Collection of of same type of Elements

Element = string

["rg-dhondhu", "rg-tondu", "rg-tondu"]

List of Strings

["rg-dhondhu", "rg-tondu", "rg-tondu"]

Element = string

Set of Strings

Map

list of strings ko set of strings me convert karna hai...

LIST AUR SET KA SYNTAX EK JAISA HOTA HAI.. bs ek me duplicate ho sakta hai aur ek me duplicate nahi ho sakta hai...

[45, 55, 65, 77]

Set of Numbers

[true, false, true, true]

Set of Bools

Bracket Types

{} **Designer Bracket Curly Bracket**	**BLOCK**
[] **CHOKOR BRACKET**	**LIST/SET CHOKOR**
() **Parenthesis Gol wala bracket**	**FUNCTION = GOL GOL**

Terraform Functions

list ki size = length()

list ko set me convert karna hai = toset()

["rg-emu1", "rg-emu2", "rg-emu3"]

Function = machine

Input	machine	Output
Aloo	aloosonamachine	Sona

Input	**function**	Output
list	**toset**	set

Syntax of function

function_name(list)

toset(["rg-emu1", "rg-emu2", "rg-emu3"])

The `for_each` meta-argument accepts a set of strings, and creates an instance for each item in that set.

```
resource "azurerm_resource_group" "resource_group" {
  for_each = toset(["rg-emu1", "rg-emu2", "rg-emu3", "rg-emu4"])
  name     = each.key
  location = "West Europe"
}
```

```
resource "azurerm_resource_group" "resource_group["rg-emu1"]" {
  name     = each.key = rg-emu1
  location = "West Europe"
}
```
each.key = rg-emu1
each.value = rg-emu1

```
resource "azurerm_resource_group" "resource_group["rg-emu2"]" {
  name     = each.key = rg-emu2
  location = "West Europe"
}
```
each.key = rg-emu2
each.value = rg-emu2

```
resource "azurerm_resource_group" "resource_group["rg-emu3"]" {
  name     = each.key = rg-emu3
  location = "West Europe"
}
```
each.key = rg-emu3
each.value = rg-emu3

```
resource "azurerm_resource_group" "resource_group["rg-emu3"]" {
  name     = "each.key = rg-emu4
  location = "West Europe"
}
```
each.key = rg-emu4
each.value = rg-emu4

- `each.key` — set member corresponding to this instance.

- `each.value` — If a set was provided, this is the same as `each.key`

Jab hum set of strings ka upyog karte hai tab each.key aur each.value dono ek hi hota hai, jo marzi use karo, koi farak nahi padta....