DATA STRUCTURE

A way of organizing data that suits a specific purpose such that operations could be performed on it appropriately.

By Keith Leonardo & Chia Amisola

Linear

A sequence of data.

Each item is called an **"element"** of the sequence and is referenced by an **integer index** starting at 0.

```
f[0]  f[1]
```

The first item of sequence f is f[0],
the second is f[1], etc.

ARRAY

A sequence of data of the same data type.

```
[3, 5, 6, 7]
```

(all integers)

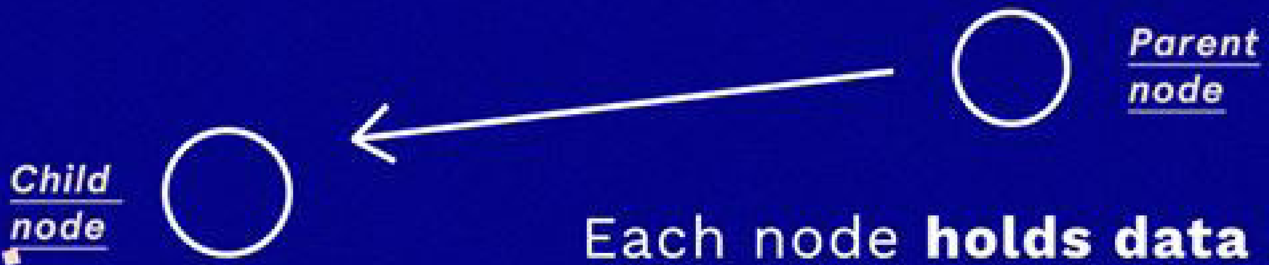
LIST

A sequence of data that could be of any data type.

```
[1, true, “hello”]
```

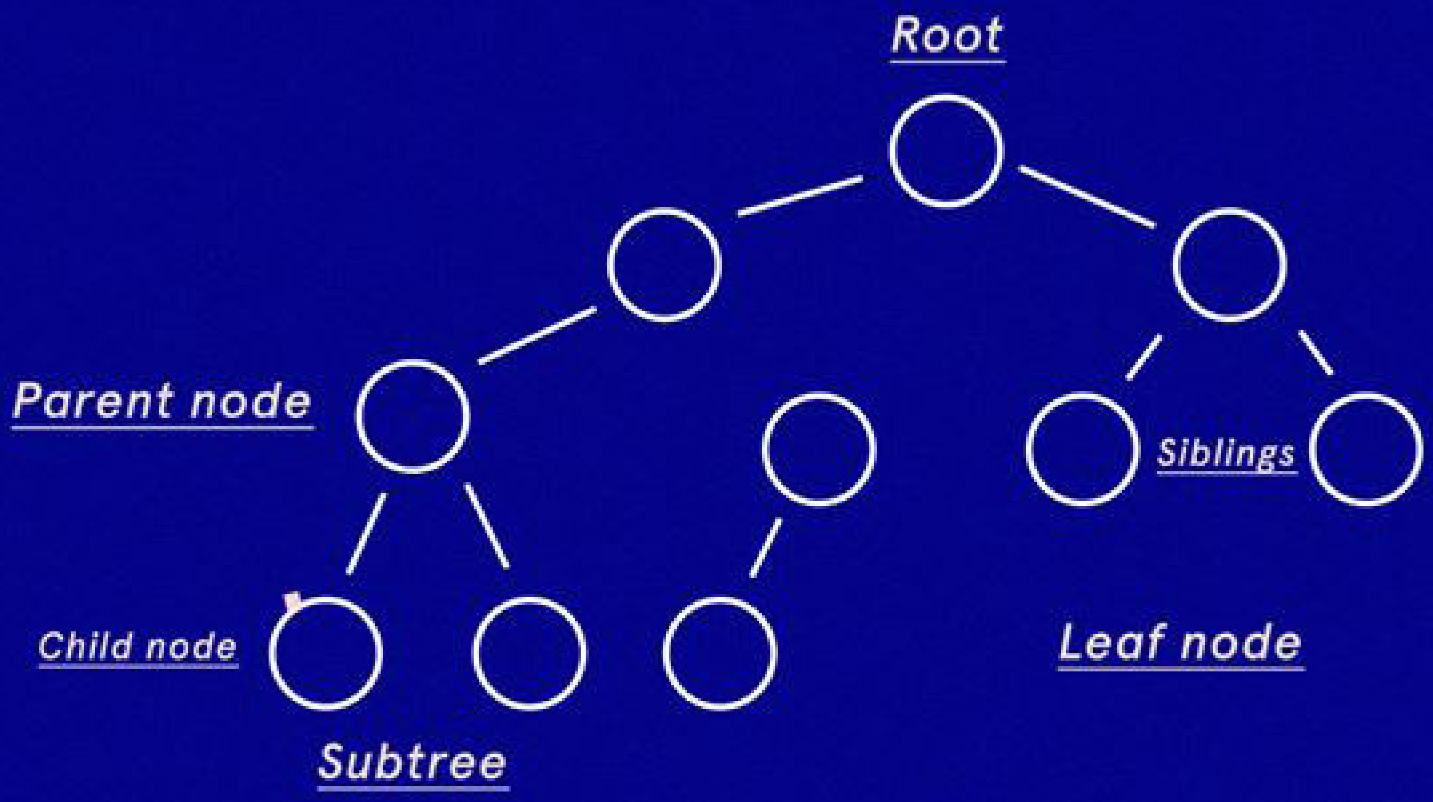
Trees

A collection of nodes and edges that **does not** have a cycle.



Each node **holds data** and references to its **“children”** nodes

It can be defined recursively, as long as no node references the root and there are no duplicate nodes **(each node is unique)**.



Terminologies

Path **-** A sequence of nodes along the edges of the tree

Root **-** The top most node of the tree

Leaf **-** A node with no children

Level **-** A “generation” of nodes

Hash Table

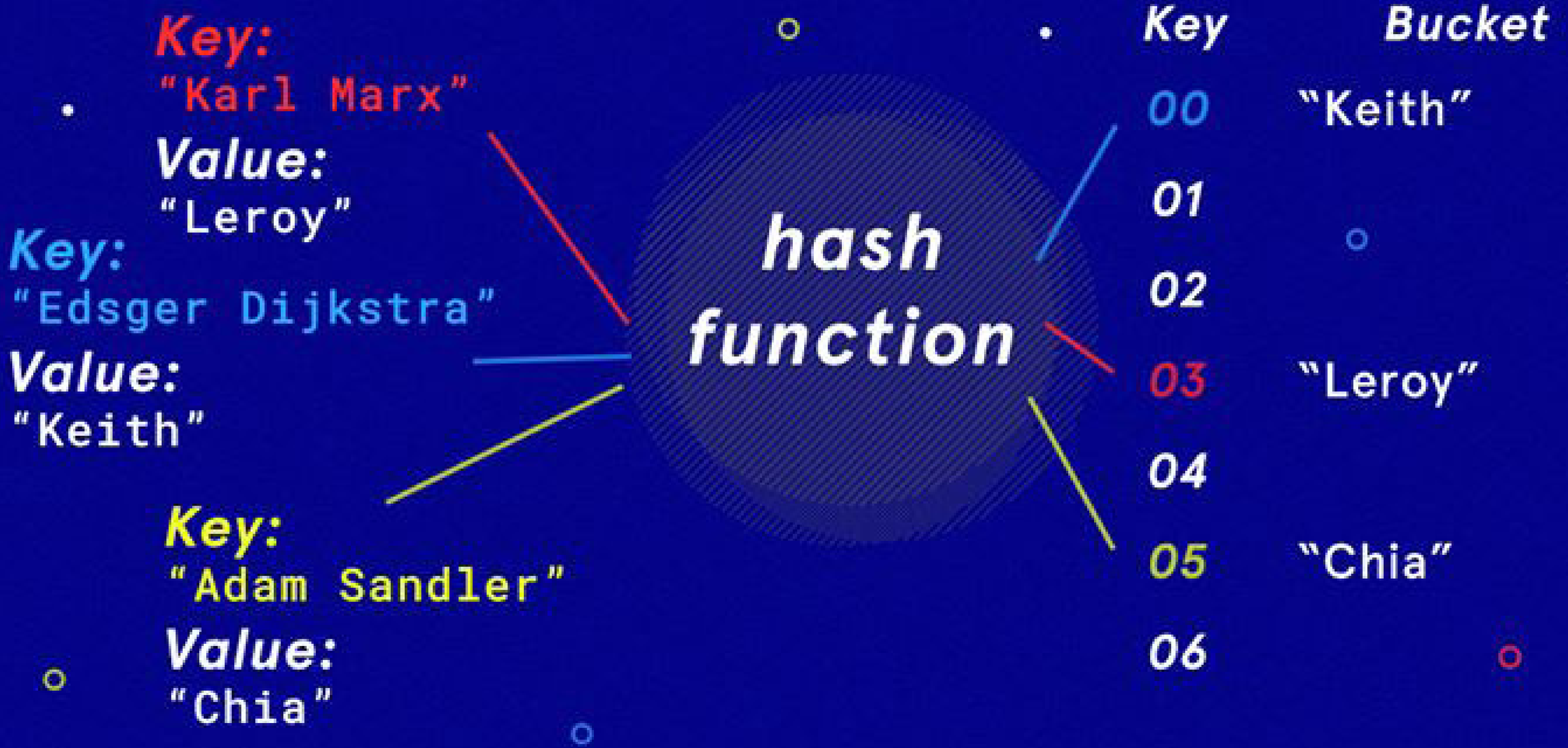
It allows the mapping of **keys to values**

There are two parts: the **"bucket" array, and the hash function.**

Hash function

The hash function is an **algorithm** that determines the index of a value in the bucket based from its **corresponding** key.

Every value must have a **unique index,** and collisions (values with the same index) should be avoided. This is a common flaw among imperfect hash functions.



Key:
"Karl Marx"
Value:
"Leroy"

Key:
"Edsger Dijkstra"
Value:
"Keith"

Key:
"Adam Sandler"
Value:
"Chia"

hash function

Key	*Bucket*
00	"Keith"
01	
02	
03	"Leroy"
04	
05	"Chia"
06	