# Garbage collection

## Reachability

In JavaScript garbage collection is implemented through something called reachability.

Variable that are reachable are kept in memory and not deleted by the garbage collector.

A value is considered to be reachable if it's reachable from a root by a reference or by a chain of references.

However, if an object is not reachable anymore, then it will be deleted by the garbage collector.

### Example

```
let user = {
  name: "John"
};
```

Let's say we have this global variable `user` referencing to the object `{name: "John"}`. If `user = null;` then the reference to "John" is deleted, thus the object "John" becomes unreachable. It will be garbage collected.

### Example

```
let user = {
  name: "John"
};


let admin = user;
```

Here, we have two references that points to the "John" object. If `user = null;` the garbage collector will not delete "John" because it is still reachable via `admin`.

---

Revision #1
Created 2022-12-19 23:59:36 UTC by Tamarine
Updated 2022-12-20 00:13:54 UTC by Tamarine