Agenda

→ Memory of JS Datatypes

→ Shallow and Deep copy

→ Custom Deep copy

→ Truthy and falsy values

Primitive and Reference Datatype

Primitive

↓

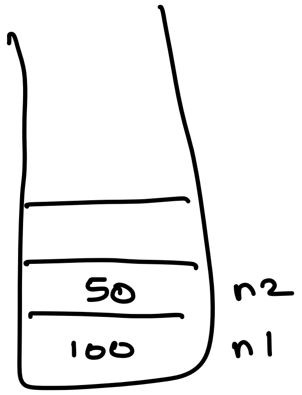
Simple and sometime referred as the lowest level of implementation.

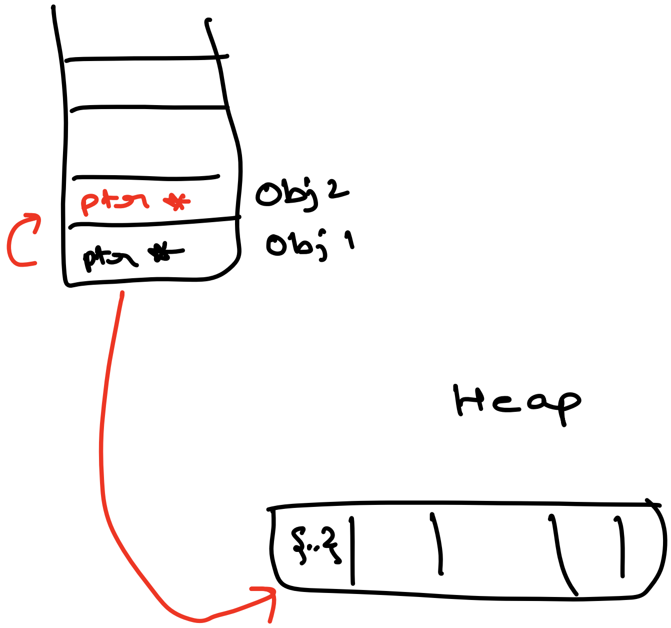
Eg:- Number, string, etc

Reference

→ Dynamic in nature. Don't have a fixed size.

Eg:- Array, function, objects, etc.





Immutability _

In functional programming paradigms, data immutability is a core concept. This means once data is created, it should not change.

When you need to update an object, you create a fresh copy of it, and update it rather than changing the original one.

$\underbrace{\{ \text{foo: "scaler"} \}}_{\text{original}}$ $\underbrace{\{ \text{foo: "scaler academy"} \}}_{\text{updated}}$

There are 2 ways of creating copies of object in JS :-

① Shallow copy

② Deep copy.

SC :- It refers to the process of creating a new object of an existing object, with it's property references the same values as original object.

→ Spread operator

→ Object.assign

DC :- JSON.stringify()

↓

Serialise the object In to a JSON string

JSON.parse()

↓

De-serialise the JSON string back to an Object.