**Namespaces:**

A namespace is a container for variables & functions.

In reality, JavaScript doesn’t have namespaces. Due to the nature of the objects, we just fake namespaces.

**Example:**

var greet = "Hello";

var greet = "Hola !";

console.log(greet);

When global execution phase 1 will run then it will think greet is undefined but when phase 2 will run then it will assign “Hello” value to var greet then “Hola !” to var greet.

var greet = "Hello";

var greet = "Hola !";

console.log(greet);

var english = {}; // we will use these objects as a container

var spanish = {};

english.greet = "Hello"; // Now the same variable is inside different objects they dont override each other

spanish.greet = "Hola ;";

// console.log(english);

/\* english.greetings.greet = "Hello !"; \*/

// remember the dot operator works left to right and greetings will return as undefined

/\* until you define the greetings inside objects literals. if you console.log english.greetings.greet you will find greetings is undefined. Literally, we are saying englihs (object) . greetings (undefined) . greet (function). \*/

// However to ensure we have an object called greetings inside object english, we need to define it.

english.greetings = {}; // now we have a sub object ready called greetings inside object english

console.log(english.greetings); // it can be done like this but not recommended

english = {

greetings: {

greet: "Hello !"

}

} // this is a better way of doing it.

console.log(english.greetings.greet);