

Cada día tenemos más código de JavaScript y necesitamos organizarlo. Uno de los problemas más habituales de JavaScript es su falta total de estructura y aislamiento. Muchas veces acabamos con un montón de variables definidas a nivel global. Para evitar este tipo de problemática podemos definir Namespaces que nos aislen unos elementos de otros. Vamos a ver un ejemplo sencillo.

```
<script type="text/javascript">

/*
var MIESPACIOA= MIESPACIOA || {};

MIESPACIOA.persona= {nombre:"pedro",edad:20};

console.log(MIESPACIOA.persona.nombre);

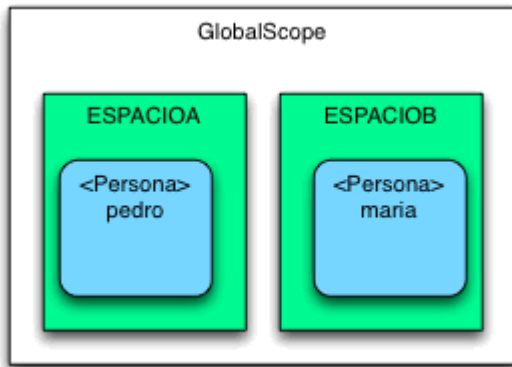
var MIESPACIOB= MIESPACIOB || {};

MIESPACIOB.persona= {nombre:"maria",edad:15};

console.log(MIESPACIOB.persona.nombre);

</script>
```

Acabamos de crear dos javascript namespace en cada uno de los cuales hemos almacenado un objeto. Eso sí ambos objetos tienen el mismo nombre “persona” aunque almacenan datos diferentes. Si esto lo hubiéramos realizado a nivel global el programa no funcionaría. Sin embargo al almacenarlo en “namespaces” todo funcionará correctamente.



Si imprimimos los datos por la consola el resultado será:

```
pedro  
maria
```

Trabajar con Namespaces es crítico en las aplicaciones JavaScript actuales . Muchos desarrolladores Java se encuentran más cómodos con una sintaxis de Namespace más cercana a su lenguaje. Para ello podemos diseñar la siguiente función que nos generará los Namespaces de una forma más sencilla.

```
var MIAPPLICACION = {};  
MIAPPLICACION.namespace = function (espacios) {  
    var listaEspacios = espacios.split('.');  
  
    var espacioPadre = this;  
  
    for (var i = 0; i < listaEspacios.length; i++) {  
  
        var nuevoEspacio=listaEspacios[i];
```

```
console.log(nuevoEspacio);

if (!espacioPadre[nuevoEspacio]) {

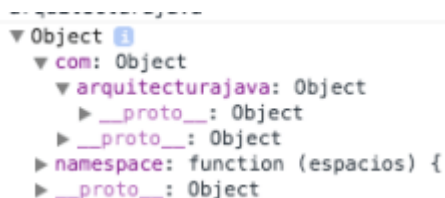
    espacioPadre[nuevoEspacio] = {};
}

espacioPadre = espacioPadre[nuevoEspacio];
}

};
//creación namespace
MIAPPLICACION.namespace("com.arquitecturajava");

console.log(MIAPPLICACION);
```

El resultado que nos imprimirá la consola será:



```
▼ Object
  com: Object
    arquitecturajava: Object
      __proto__: Object
    namespace: function (espacios) {
    }
    __proto__: Object
```

Como vemos hemos generado la estructura de Namespaces que deseábamos.

Otros artículos relacionados: [Javascript Templates](#) , [jQuery Context](#) , [Javascript JPerf](#)