



Un recurso que se usa en la mayoría de los programas, para facilitar su redacción son las librerías.

Son un conjunto de instrucciones que realiza funciones básicas como leer, escribir en pantalla, trabajar números y funciones matemáticas, manejo de memoria, entre otros. Reducen la cantidad de instrucciones de un programa.

Un ejemplo simple Class.

```
var Animal = new Class({
  initialize: function(age){
    this.age = age;
  }
});

var Cat = new Class({
  Extends: Animal,
  initialize: function(name, age){
    // calls initialize method of Animal class
    this.parent(age);
    this.name = name;
  }
});

var myCat = new Cat('Micia', 20);
alert(myCat.name); // alerts 'Micia'.
alert(myCat.age); // alerts 20.
```

Cada librería está delimitada por el trabajo que desempeñará y cada lenguaje de programación tiene sus propias librerías que deben ser escritas al inicio de la redacción de un programa. Algunas están ligadas, por lo que es necesario conocer las más relevantes para el lenguaje elegido.

Librerías estáticas

Se caracterizan por **unirse al programa** una vez que se escriben y se compilan en un lenguaje de programación, por lo que se puede usar en cualquier máquina, sin embargo, **vuelve al programa ejecutable mucho más pesado**.

Ajax.

```
// create a new Class instance
var myRequest = new Request({
  url: 'getMyText.php',
  method: 'get',
  onRequest: function(){
    myElement.set('text', 'loading...');
  },
  onSuccess: function(responseText){
    myElement.set('text', responseText);
  },
  onFailure: function(){
    myElement.set('text', 'Sorry, your request failed :(');
  }
});

// and to send it:
myRequest.send(data);
```

Librerías dinámicas

Se deben compilar cada vez que sean llamadas y, si se pretenden usar en otra máquina, se debe cargar y compilar nuevamente; pero su programa ejecutable será mucho más pequeño.

Sintaxis.

```
$('#myInput').tidy();
```