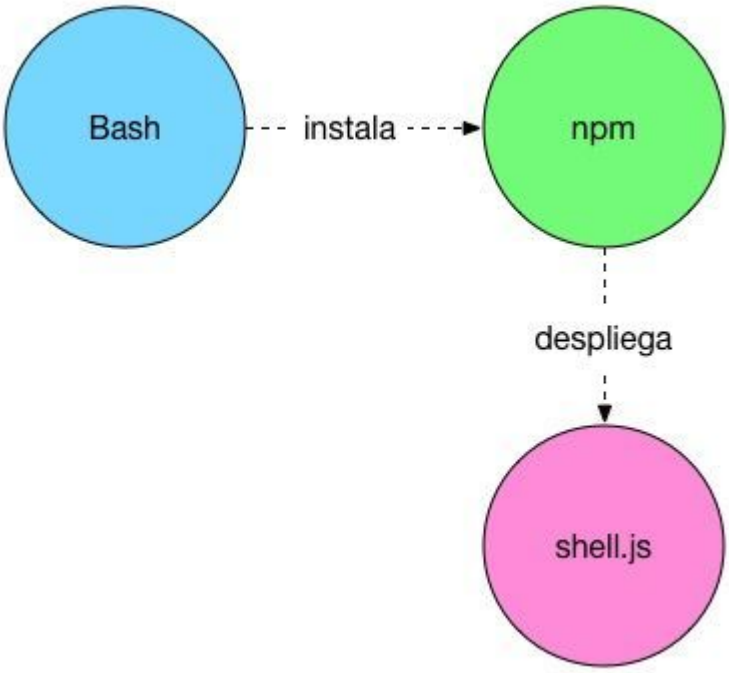
¿Qué es Shell.js?. En muchas ocasiones los desarrolladores nos encontramos con la necesidad de crear un script que realice alguna tarea tediosa. Es común usar herramientas como ANT para abordar estas tareas desde el mundo Java. Lamentablemente no siempre disponemos de la herramienta más cercana a nuestra tecnología para abordar la creación de un script. De hecho en muchas ocasiones nos encontramos en un entorno linux y lo más natural es crear un script de bash. El problema casi siempre es que no tenemos en ese momento tiempo para aprender la sintaxis básica de bash a nivel de script . ¿Cómo podemos solventarlo de una forma sencilla? . Una solución rápida es instalarnos node.js y Shell.js.



Vamos a ver un ejemplo.

```
node install shell.js -g
```

Esto nos instalara para Node.js una shell , a partir de este momento podemos construir

nuestros propios scripts de bash pero usando sintaxis de JavaScript. Vamos con el primero.

```
require("shelljs/global");
echo ("vamos a empezar");
echo ("listando el directorio");
var ficheros= ls(".").forEach(function(elemento) {

console.log(elemento);
});
```

En este caso acabamos de listar la carpeta actual utilizando el comando ls() de la shell , para ejecutarlo utlizamos node:

```
node miscript.js
```

El resultado aparecerá por la consola:



En este caso muestra algunos de los ficheros js con los que estoy trabajando. Es momento de crear una carpeta y copiar todos los ficheros a la carpeta “copia”.

```
require("shelljs/global");

echo ("creando carpeta");
mkdir("copia/");
echo ("copiando ficheros");

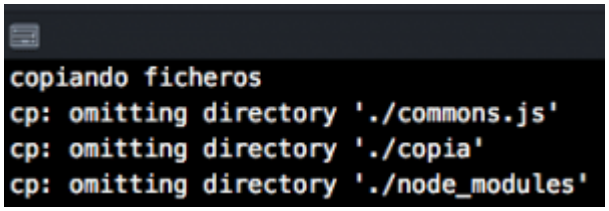
cp("./*","copia/");

cd("copia/");

var ficheros= ls().forEach(function(elemento) {

console.log(elemento);
})
```

Este código nos copia los ficheros de la carpeta actual a “copia”.



Al no ser recursivo se omiten algunas carpetas. Podemos usar el API clásico de JavaScript para realizar un filtrado y sacar por pantalla solo los ficheros con extensión html.

```
var ficheros= ls();

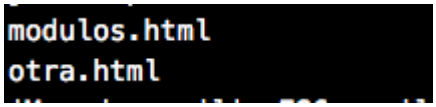
ficheros.filter(function(elemento) {
```

```
return elemento.includes(".html");

}).forEach(function(elemento) {

console.log(elemento);
});
```

En el resultado solo veremos los html:



Como siempre Node.js es capaz de hacer casi cualquier cosa y en este caso evitarnos el uso de el script de bash al cambiarlo por algo más intuitivo para los desarrolladores web . Shell.js es una gran ayuda.

Otros artículos relacionados: JavaScript ES6 fetch API , JavaScript Bundle