

Desarrollo de Aplicaciones telemáticas
Examen final, prueba de teoría. 2 de junio de 2021
Grado en Ingeniería en Tecnologías de la Telecomunicación
Grado en Ingeniería en Sistemas de la Telecomunicación
Universidad Rey Juan Carlos

Instrucciones:

- Ejecuta en un terminal `~mortuno/prepara`
- Esto dejará en el ordenador el fichero `~/dat.junio.21/examen.TULOGIN.txt`, donde debes escribir tus respuestas. TULOGIN será tu nombre de usuario en el laboratorio.

Ejercicio 1 (4 puntos)

¿Cuál es la diferencia entre `=`, `==` y `===` en JavaScript? ¿Para qué se usa cada uno?

Respuesta

- El igual simple sirve para asignar un valor a una variable.
- El igual doble es el comparador de igualdad *permissivo* que implementaba inicialmente JavaScript. Su propósito es considerar iguales dos valores aunque su tipo de datos sea distinto. Demostró ser muy problemático, es preferible no usarlo nunca.
- El igual triple, que JavaScript llama *estricto* es el que deberíamos usar siempre, es como el de cualquier otro lenguaje de programación: para que dos valores sean iguales, en condición necesaria (pero no suficiente) que sean del mismo tipo.

Ejercicio 2 (3 puntos)

Explica brevemente la siguiente afirmación: En JavaScript 6 la programación orientada a objetos basada en herencia es *azúcar sintáctico*.

Respuesta

La programación orientada a objetos basada en prototipos es el tipo de programación nativo en JavaScript. Pero muchos programadores prefieren la programación orientada a objetos basada en clases, porque está mucho más extendida y porque funciona bien para problemas de complejidad baja o media.

Así que en JavaScript 6 aparece, como opción adicional, una sintaxis similar a la de cualquier POO basada en clases. Es solo un recubrimiento externo, una apariencia. Internamente sigue siendo POO basada en prototipos, pero *endulzada* para que sea *más fácil de digerir* por quienes prefieren la programación tradicional.

Ejercicio 2 (3 puntos)

¿Para qué sirven los *web workers*?

Respuesta

Se trata de un API de JavaScript que permite ejecutar un hilo de código en segundo plano. Son especialmente útiles para tareas que consuman mucho tiempo, normalmente por ser intensivas en uso de CPU o de comunicaciones, porque así no interfieren a la tarea en primer plano donde está el interfaz de usuario.