

Desarrollo de Aplicaciones Telemáticas
Examen de teoría. 8 de Mayo de 2018
Grado en Ingeniería en Tecnologías de la Telecomunicación
Universidad Rey Juan Carlos

Instrucciones:

- Entra en el puesto del laboratorio con el nombre de usuario *examen* y la contraseña *examen*.
- Ejecuta el script `prepara_dat`
- Esto dejará en el ordenador el fichero `~/dat/TULOGIN/dat.teoria.mayo.18.txt`, donde debes escribir tus respuestas. TULOGIN será tu nombre de usuario en el laboratorio.
- Encontrarás los pdf de las transparencias en `~/dat/transpas`

Ejercicio 1

La transparencia 15 del tema 8 , refiriéndose a JSONP y a la *same site policy* dice :
Se entiende que esto no es especialmente peligroso. Bancofuenla podría enviar una contraseña a una página web, como un dato, esperando que la lea el usuario. Pero en un script nunca debería haber información sensible

Explica esto.

Respuesta

La *same site policy* obliga a que un script incluido en un HTML solo pueda acceder a datos del mismo sitio web donde se descargó el HTML, no de un sitio distinto. Esta política se aplica solo a datos, no a scripts. JSONP es un *truco* que permite que el script acceda a datos de un sitio web diferente. Para ello, los datos se *disfrazan* de funciones. Aunque los datos se *maquillan* siguen siendo datos. Pero en general esto no se considera peligroso, porque si el servidor *participa en el truco*, si acepta enviar los datos de esta forma peculiar, es porque sabe que los va a manejar un script de un sitio distinto, del que no tiene por qué fiarse.

Por tanto, estos datos *disfrazados de función* nunca serán sensibles. Los datos sensibles solo se enviarán como dato *normal* y por tanto estarán protegidos por la *same site policy*.

Ejercicio 2

En JavaScript, para averiguar la longitud de una cadena escribimos `x.length`, y no `x.length()` ni `length(x)`.

¿Por qué?

Respuesta

Porque en JavaScript la longitud de una cadena es un atributo. `x.length()` sería un método, y `length(x)` sería una función.

Ejercicio 3

Explica el problema del *yo-yo* en la programación orientada a objetos *tradicional*. Explica por qué no se da en la programación orientada a objetos basada en prototipos. Ten en cuenta que si te limitas a copiar el contenido de la transparencia, tu respuesta no tendrá ningún valor.