

Aplicaciones telemáticas
Examen final, prueba de teoría. 15 de mayo de 2023
Grado en ingeniería telemática.
Universidad Rey Juan Carlos

Instrucciones:

- Ejecuta en un terminal `~mortuno/prepara`
- Esto dejará en el ordenador el fichero `~/at.mayo.23/teoria.TULOGIN.txt`, escribe ahí tus respuestas. TULOGIN será tu nombre de usuario en el laboratorio.

Ejercicio 1 (4 puntos)

Indica si los siguientes selectores CSS son correctos o no. Si son correctos, indica su significado. Si no son correctos, explica brevemente por qué.

- 1) `tr img`
- 2) `minimo .p`
- 3) `p .minimo`
- 4) `.minimo, #oscuro`

Respuesta

- 1) Correcto. Selecciona los elementos *img* descendentes de filas de una tabla ¹.
- 2) Incorrecto. Seleccionaría los elementos de clase *párrafo* dentro del elemento html *minimo*.
 - Lo primero, usar un nombre de clase que coincida con un elemento de HTML, es legal. Seguramente es mala idea, porque induce a confusión, pero la norma lo permite.
 - Lo segundo es incorrecto: *minimo* no es un elemento HTML.
- 3) Correcto. Selecciona los elementos de clase *minimo* descendentes de un párrafo.
- 4) Correcto. Se refiere a los elementos de clase *minimo* y al elemento con identificador *oscuro*. En otras palabras: a los elementos que tengan la clase *minimo* o el identificador *oscuro*²

Ejercicio 2 (3 puntos)

En bootstrap ¿Cuándo y por qué se presentan las columnas en disposición normal? ¿Cuándo y por qué se presentan las columnas en disposición apilada?

¹Si estuviera al revés, `img tr`, sería incorrecto porque las imágenes son de tipo void, no pueden tener contenido

²Sería un error interpretar esto como 'elementos con clase *minimo* e identificador *oscuro*, simultáneamente'

Respuesta

- Las columnas, que en la asignatura preferimos llamar *celdas*, se presentan en disposición normal, esto es, unas al lado de otras, cuando la pantalla tiene el tamaño *suficiente*. Este es el diseño previsto por el autor de la página cuando no hay restricciones de tamaño.
- Se presentan apiladas, unas encima de otras, cuando la pantalla es *pequeña*. De esta forma cada una de las celdas se puede ver completa, aunque la composición global no resulte tan vistosa.

El tamaño de la pantalla se considera *suficiente* o *pequeña* según lo indicado por el autor de la página con las clases *.col-N*, *.col-sm-N*, *.col-md-N*, *etc.*

Ejercicio 3 (3 puntos)

¿Por qué tantos programadores se quejan de la programación orientada a objetos de JavaScript?
¿Tienen fundamento esas quejas?

Respuesta

La programación orientada a objetos (POO) nativa de JavaScript es de un tipo poco habitual, la POO basada en prototipos. Es distinta a la POO basada en clases, que es con diferencia la predominante entre los lenguajes de programación. La POO basada en prototipos se considera generalmente más potente y libre de muchos de los problemas habituales de la POO basada en clases. Pero también tiene sus inconvenientes: está poco extendida y tiene una curva de aprendizaje muy pronunciada.

Cuando JavaScript solo ofrecía POO basada en prototipos, sí resultaba razonable criticar al lenguaje por esto. Actualmente esta crítica es, por lo menos, muy discutible: desde ECMAScript 6 (año 2015), JavaScript ya tiene, además, una sintaxis que permite trabajar de manera muy similar a la de cualquier lenguaje orientado a objetos tradicional.