

Aplicaciones telemáticas
Examen de teoría. 9 de mayo de 2019
Grado en ingeniería telemática. 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` y contesta sus preguntas.
- Esto dejará en el ordenador el fichero `~/at/TULOGIN/teoria.txt`, donde debes escribir tus respuestas. TULOGIN será tu nombre de usuario en el laboratorio.

Ejercicio 1 (3.3 puntos)

Este fragmento de HTML 4 no es válido en HTML 5. Explica por qué.

```
<p><font size="4" color="red" face="monospace">Hola, mundo</font>
```

Si sabes escribir una versión equivalente correcta en HTML 5, hazlo. Pero lo más importante es que describas por qué esto ya no es correcto y qué técnica se usa para escribir algo equivalente en HTML 5.

Respuesta

En este ejemplo, los atributos del texto se aplican directamente a un párrafo. En HTML 4 esto era posible, aunque se consideraba una mala técnica. El problema es que si queremos cambiar alguna de estas propiedades, tendríamos que retocar uno a uno todos los párrafos. Ya en HTML 4 se consideraba mucho más adecuado usar estilos CSS. De esa forma, asignamos a los párrafos el estilo, y si luego queremos modificar alguna propiedad del estilo, basta modificar la definición del estilo en un solo sitio, no en todos y cada uno de los párrafos. En HTML 5 ya no es algo desaconsejado, sino directamente incorrecto.

Por otro lado, el párrafo con etiqueta de apertura pero sin etiqueta de cierre tienen la misma consideración tanto en HTML 4 como en HTML 5: es correcto, pero siempre es preferible el cierre explícito. Este es un ejemplo de página similar en HTML 5 (añadiendo el cierre explícito, lo que no es imprescindible):

```
<!DOCTYPE html>
<html lang="es-ES">
<head>
  <meta charset="utf-8">
  <title>Ejemplo de estilos con CSS</title>
  <style>
    .clase01 { color: red; font-family : courier; font-size: large }
  </style>
</head>

<body>
  <p class="clase01">Hola, mundo.</p>
</body>
</html>
```

Ejercicio 2 (3.4 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. `div.ayuda`
2. `div .ayuda`
3. `div #ayuda`
4. `div ayuda`

Respuesta

1. Correcto. Selecciona los elementos *div* que sean de clase *ayuda*.
2. Correcto. Selecciona los elementos de clase *ayuda* que estén contenidos dentro un elemento *div*. (Contenidos directamente como *hijos* o como *nietos*, *bisnietos*, etc).
3. Correcto. Selecciona el elemento con el identificador *ayuda*, si está contenido dentro de un elemento *div*.
4. Incorrecto. Sería los elementos *ayuda* que estén dentro de elementos *div*, pero en HTML no existen los elementos *ayuda* (pueden existir elementos con esta clase o con este identificador, pero no es un tipo de elemento).

Ejercicio 3 (3.3 puntos)

¿Qué significa *fuertemente acoplado*? ¿Qué significa *débilmente acoplado*? ¿Cuál es la relación entre estos conceptos y la programación orientada a objetos basada en clases y la programación orientada a objetos basada en prototipos?

Respuesta

Fuertemente acoplado y *débilmente acoplado* son conceptos que se aplican a cualquier sistema modular. Si los módulos necesitan mucha información unos de otros para poder ensamblarse, decimos que son *fuertemente acoplados*. Si necesitan poca información unos de otros, son *débilmente acoplados*. Y por supuesto, hay todo tipo de valores posibles intermedios. Así podríamos decir que los componentes de un ordenador tipo torre tienen un acoplamiento mucho más débil que los de un portátil: en los primeros es fácil por ejemplo quitar una tarjeta de vídeo y poner otra, en los segundos, no.

La programación orientada a objetos basada en clases tiene un acoplamiento más fuerte que la basada en prototipos, porque un objeto instanciado a partir de una clase normalmente hereda todas sus propiedades, y las de sus clases padre, abuelo, etc. Mientras que con las técnicas de programación orientada a objetos basadas en prototipos, los objetos se crean a partir de objetos, controlando qué elementos se reaprovechan y cuáles no.