

Desarrollo de Aplicaciones Telemáticas
Examen final, prueba de teoría. 18 de mayo de 2026.
Grado en Ingeniería en Tecnologías de la Telecomunicación
Grado en Ingeniería en Sistemas de la Telecomunicación
Universidad Rey Juan Carlos.

Instrucciones:

- Rellena el fichero `teoria.txt` con tus datos personales y la respuesta a las cuatro preguntas siguientes. No cambies el nombre del fichero.

Ejercicio 1 (3.5 puntos)

Indica si los siguientes selectores CSS son correctos o no. Si son correctos, describe su funcionamiento. Si no son correctos, explica brevemente por qué. Un apartado erróneo o en blanco tendrá una penalización del 50 %. Dos o más apartados erróneos o en blanco, tendrán puntuación nula.

- 1) `table spam`
- 2) `table.spam`
- 3) `div, span`
- 4) `div #spam`

Respuesta

- 1) Sintácticamente es correcto, seleccionaría todos los elementos *spam* descendentes de elementos *table*. Pero *spam* (correo basura) no es un elemento HTML estandar. Por tanto es un selector que no tiene sentido, nunca seleccionaría nada.
- 2) Correcto. Selecciona todos los elementos *table* con la clase *spam*.
- 3) Correcto. Selecciona un elemento si es un *div* o si es un *span*. Esto es, selecciona los *div* y los *span*.
- 4) Correcto. Selecciona el elemento con el identificador *spam*, si es descendente de un *div*.

Ejercicio 2 (2.5 puntos)

De cada una de estas afirmaciones, indica si es verdadera o falsa. Si es verdadera justifícala brevemente. Si es falsa, explica todos los motivos por los que lo es.

1. Cuando el ratón abandona un elemento, o cualquiera de sus descendentes (hijos, nietos), este elemento recibe el evento *mouseout*.
2. Cuando un elemento recibe un evento, todos sus descendientes (hijos, nietos...) lo reciben también.
3. Cuando el ratón llega a un elemento, este recibe el evento *mouseover*.
4. Cuando un elemento recibe un evento, todos sus ascendentes (padre, abuelo...) lo reciben también.

Respuesta

1. Cierto.
2. Falso. Un evento se propaga hacia los ascendentes, no hacia los descendentes. Ejemplo: una tabla contiene una fila, la fila contiene varias imágenes. Un evento sobre la fila se propaga a la tabla. Pero no a las imágenes.
3. Cierto.
4. Cierto.

Importante: aquí hablamos de elementos HTML anidados unos dentro de otros, no a las jerarquías de clases en JavaScript. Cualquier explicación basada en herencia de clases es errónea.

Un elemento HTML se corresponde con un objeto en el DOM. Pero esto no tiene nada que ver con ninguna clase en JavaScript.

Ejercicio 3 (2 puntos)

En diseño web, en *frameworks* como Bootstrap y similares ¿qué se entiende por *grid*?

Respuesta

Es la manera habitual en la actualidad de conseguir que una página web se vea bien en cualquier tamaño de pantalla. El diseñador crea el contenido dentro de una rejilla (grid) formada por *cajas* o *celdas*, denominadas realmente *columnas*. De forma similar a las columnas de un periódico de papel. Cada *caja* es una unidad de contenido.

- Si la pantalla es lo bastante grande, las cajas se muestran en disposición *normal*, esto es en horizontal, de forma que se ven varias cajas por fila.
- Si la pantalla es más pequeña, las cajas se muestran *apiladas*, esto es, en cada fila solo se ve una caja, pero la caja se ve completa.

Ejercicio 4 (2 puntos)

1. Supongamos que alguien dice *el TypeScript se transpila a JavaScript*. ¿Qué significa? ¿Tiene sentido?, ¿Se hace realmente? ¿Por qué?
2. Supongamos que alguien dice *el JavaScript se transpila a TypeScript*. ¿Qué significa? ¿Tiene sentido?, ¿Se hace realmente? ¿Por qué?

Respuesta

1. Esto significa convertir el código TypeScript en código JavaScript. Tiene mucho sentido y es la manera habitual de trabajar en TypeScript. TypeScript es un superconjunto de JavaScript, esto es, JavaScript con ampliaciones. Estas ampliaciones no están soportadas por los navegadores, por tanto a partir de TypeScript se genera código JavaScript estandar que implementa esta funcionalidad y que será el que ejecute el navegador.
2. Esto sería convertir el código JavaScript en código TypeScript. No suele hacerse como proceso de compilación normal, no es necesario. JavaScript es un subconjunto de TypeScript. En otras palabras, todo código JavaScript válido ya es TypeScript válido.