

Tarefa 3.1

Axenda de contactos v3

Nome d@ alumn@ *[Nome apelido1 apelido2]*

Grupo *2º DAM/DAW Ordinario*

Enunciado

Partindo do ficheiro `tarefa3.1.html` e do código implementado para a Tarefa 2.12 completa o código JS e CSS para obter o código da **Axenda de Contactos** tal e como se establece nos seguintes apartados.

IMPORTANTE: só poden modificarse os ficheiros `tarefa3.1.css` e `tarefa3.1.js`, e ningún outro.

Apartado 1: vista CSS (0,5 puntos)

Crea o ficheiro `tarefa3.1.css` a partir de `tarefa2.2.css` de xeito que manteña os estilos, incorpore as correccións que te puideran ter sido indicadas e ademais:

- Aplique estilos aos novos elementos `Código Postal` e `Provincia` da vista mantendo un estilo homoxéneo e estruturado. **(0,5 puntos)**

Apartado 2: os campos código postal e provincia (1,5 puntos)

Crea o ficheiro `tarefa3.1.js` a partir de `tarefa2.2.js` no que implementarás a funcionalidade necesaria para que:

- O select `provincia` se encha cos datos das provincias de España tomadas directamente dun `array` e deixando como opción por defecto o campo en branco. **(0,5 puntos)**
- O select `provincia` non sexa editable namentres haxa algo escrito no campo `código postal`. **(0,5 puntos)**
- Ao escribir un código postal válido se escolla automaticamente a provincia correspondente no select. **(0,5 puntos)**

Apartado 3: manipulación de DOM (0,75 puntos)

Fai os cambios necesarios no ficheiro `tarefa3.1.js` para que toda operación de manipulación de DOM se faga conforme aos estándares, isto é:

- Usando as propiedades e métodos do obxecto `document` (evitando na medida do posible traballar con `innerHTML` e **non** usando baixo ningún concepto os métodos `write()` ou `writeln()`). **(0,25 puntos)**
- **Non** usando elementos tipo `alert` ou `prompt`, considerando que a nosa interface ten que ser única. **(0,25 puntos)**

Apartado 4: tratamento de datos (3 puntos)

Fai os cambios necesarios no ficheiro `tarefa3.1.js` para que o campo clave actúe como clave primaria da nosa axenda, de tal modo que:

- Non se admitan insercións cun nome xa en uso (sen distinguir maiúsculas de minúsculas). **(0,75 puntos)**

- Se amose unha mensaxe de erro na parte superior do formulario indicando que ese nome está xa en uso. **(0,25 puntos)**
- Non se admitan modificacións de nomes inexistentes na axenda. **(0,75 puntos)**
- Se amose unha mensaxe de erro na parte superior do formulario indicando que non se pode modificar un contacto que non existe. **(0,25 puntos)**
- Non se admitan eliminacións de nomes inexistentes na axenda. **(0,75 puntos)**
- Se amose unha mensaxe de erro na parte superior do formulario indicando que non se pode eliminar un contacto que non existe. **(0,25 puntos)**

Apartado 5: validación (4 puntos)

Fai os cambios necesarios no ficheiro **tarefa3.1.js** para que haxa validación no noso formulario, tal e como segue:

- As validacións aplicaranse no intre de premer os botóns correspondentes. **(0,25 puntos)**
- Serán campos requiridos para as operacións de engadir e modificar: nome, apelidos e teléfono. **(1 punto)**
- Será un campo requirido para as operacións de eliminar e buscar: nome. **(0,5 puntos)**
- Os campos nome e apelidos só admitirán: letras (con ou sen acentos gráficos), apóstrofos, espazos e guións. **(1 punto)**
- O campo teléfono só admitirá números. **(0,5 puntos)**
- O campo e-mail (no caso de estar cuberto) só admitirá formatos de e-mail válido. **(0,5 puntos)**
- Mensaxes de validación:
 - Para cada erro de validación amosarase unha mensaxe mediante un texto en letra pequena e cor vermella que aparecerá xusto debaixo da etiqueta do campo afectado, cunha mensaxe significativa. **(0,75 puntos)**
 - As mensaxes de validación deberán desaparecer cada vez que se prema un botón (xa que se recalcularán as validacións). **(0,25 puntos)**

Outras condicións de avaliación

- Todo o código CSS terá que estar correctamente estruturado.
- Todo o código CSS terá que estar correctamente comentado, de xeito que os comentarios resulten explicativos e abondo detallados para entender en que consisten as regras aplicadas e a que elementos aplican.
- Todo o código JS terá que estar correctamente estruturado.
- Todo o código JS terá que estar correctamente comentado, de xeito que os comentarios resulten explicativos e abondo detallados para entender que se está a facer nos distintos bloques de código e para que serven.
- Non podes modificar o ficheiro **tarefa3.1.html**, xa que non se usará a túa versión para a corrección da tarefa.

O incumprimento dalgunha destas condicións de avaliación conlevará unha penalización máxima de -1 punto por cada unha delas.

Indicacións de entrega

Achega un ficheiro comprimido **ZIP** de xeito que só conteña:

- O ficheiro **tarefa3.1.js**.
- O ficheiro **tarefa3.1.css**.

Entrega a través desta plataforma, no xeito establecido para tal fin. O arquivo nomearase seguindo estas pautas:

apelido1_apelido2_nome – tarefa3.1.zip

Asegúrate que o nome non conteña caracteres especiais (como a letra ñ ou acentos). Así, por exemplo, unha alumna cuxo nome sexa María Begoña Sánchez Mañas debería nomear esta tarefa:

sanchez_manas_maria_begona – tarefa3.1.zip

CALQUERA TAREFA QUE NON CUMpra AS INDICACIÓNS DE NOMENCLATURA E FORMATO ANTERIORES CONSIDERARASE COMA NON ENTREGADA.

NON SE ADMITIRÁ NINGUNHA ENTREGA FORA DE PRAZO. SE O PRAZO ESTÁ PRÓXIMO A FINALIZAR E TES ALGÚN PROBLEMA COA PLATAFORMA, FAI ENTREGA VÍA E-MAIL