



Frontend side

Preguntas

- HTML. Si queremos resaltar un texto, ¿qué deberíamos usar, el tag `` o ``? ¿Por qué?
- CSS. ¿Qué ventajas presenta el uso de un preprocesador de CSS? ¿Inconvenientes? ¿Es recomendable su uso en cualquier tipo de proyecto?
- JS. ¿Qué ventajas o inconvenientes presenta el uso de `promises` respecto a `callbacks`?
- Exposición. A nivel de usabilidad, presentación e interacción; por favor, desarrolla un breve análisis (no más de 15 líneas) de qué falla y qué se podría mejorar en la web actual de [Cymit](#).

Desarrollo

Los entregables para los ejercicios se espera que sean ficheros **HTML5**, **JS** y **CSS3**. El foco de la prueba será la **calidad del código (no ofuscado/minificado)**, aunque se agradecerá que a nivel visual sea presentable.

Ejercicio 1

Dado un formulario de registro de usuario con los siguientes campos:

- Email
- Password
- Nombre y apellidos
- Dirección
- Código postal

Preparar en el lado del cliente, en **vanilla JS** y **sin librerías externas**, un validador que haga las siguientes comprobaciones:

- Todos los campos (excepto password) tienen al menos 4 caracteres.
- El email tiene un formato válido.
- El password debe ser mínimamente seguro, 8 caracteres con letras y números.
- Puntos extra si el código es acompañado de tests.

Ejercicio 2

Crear una función (se pueden utilizar librerías externas y/o APIs) que convierta una cadena de texto con una dirección en la siguiente información segmentada:

- Calle y número
- Código postal
- Ciudad
- País

Ejemplos de entrada:

- *C\ Santander 42, 08020, Barcelona, Spain*
- *C\ Genova, 628004 Madrid*
- *Shützenstrasse 12 CH-4410 Liestal Suiza*

Ejercicio 3

Conociendo el esquema de datos (ver 'producto de ejemplo' en la prueba de backend, página 3), ¿cómo crees que se podría mostrar la información (propuesta con un diseño wireframe o razonamiento con texto)?



Backend side

Cymit Química es un espacio de encuentro entre oferta y demanda, entre compradores y vendedores en el sector de la química. Ofrecemos una solución integral de compra/venta para los investigadores y los fabricantes de productos dirigidos al sector químico. Trabajamos a diario para hacer la plataforma más rápida, contener más información de calidad y una oferta más amplia y competitiva.

Uno de nuestros mayores retos es mantener un catálogo de +2M productos de distintos proveedores. En esta prueba, se pide una extracción de datos típica de un proveedor. El output debe ser un archivo JSON. En éste, se representa una lista de diccionarios (o un array de arrays asociativos).

Se adjunta un ejemplo con todos los campos que potencialmente se pueden extraer.

Campos obligatorios: pid, name, url, packaging.size, packaging.price (si un producto no dispone ningún conjunto de envase/precio, debe mostrarse con valores por defecto: envase:"ne", precio:0.0).

- Campos opcionales: el resto.

Se espera que la lista alcance al menos 100 productos, aunque el listado completo alcanza ~290k. El listado completo de productos puede ser accedido desde:

- <https://www.abcr.de/shop/en/catalogsearch/advanced/result/?limit=25&mode=list&name=%25&offset=0>

La prueba se considerará válida si se muestran los campos obligatorios para todos los productos.

Se valorará más positivamente cuántos más campos opcionales hayan sido extraídos.

Detalles

- "pid" se corresponde con Product No.
- "Structure" (Formula) sólo permite tags sup o sub, o texto plano.
- MSDS significa "Material Safety Data Sheet". El valor de "pdf_msd" debe ser la URI de dicho documento en su versión en inglés, si está especificado.
- "synonims" es un campo de lista de longitud indeterminada (puede estar vacío).
- "packaging" es un array_assoc o dict, en él se enumera los conjuntos de tamaños y precios proporcionados por el proveedor.
 - "packaging.size" sin espacios y minúscula. (la key es el propio valor)
 - "packaging.price" es un float (sin mostrar divisa).
- "properties.weight" indica el peso molecular o MW.
- "properties.melting_point" (boiling/flash): debe mostrar el valor tal como lo representa el proveedor.
- Se deja un campo libre 'other' donde se puede añadir más información estructurada si lo consideras necesario.



Producto de ejemplo

```
[
  {
    "pid": "0590S",
    "name": "Bergapto1",
    "CAS": "486-60-2",
    "structure": "C<sub>11</sub>H<sub>6</sub>O<sub>4</sub>",
    "smiles": "OC1=C2C=CC(=O)OC2=CC2=C1C=C2",
    "description": "[...]",
    "url": "https://domain.com/product_page.html",
    "img": "https://domain.com/product_page/main_image.png",
    "pdf_msds": "https://domain.com/product_page/sds_document.pdf",
    "synonyms": [
      "4-Hydroxypsoralen",
      "5-Hydroxypsoralen"
    ],
    "packaging": {
      "10mg": 139.00,
      "25mg": 242.50
    },
    "properties": {
      "molecular_weight": "202.17",
      "melting_point": "276-288°C",
      "boiling_point": "675°C",
      "flash_point": "",
      "purity": "≥ 99%"
    },
    "safety": {
      "p_statements": ["P260", "P262"],
      "h_statements": ["H302", "H315", "H319", "H335"],
      "symbol": "SGH07",
      "storage": "Store at : <+8°C. Store in dry and dark place.",
      "shipping": "Ambient temperature"
    },
    "other": {
      "family": "Coumarin",
      "subfamily": "Furocoumarin",
      "...": "..."
    }
  },
  {
    "pid": "0823M",
    "...": "..."
  }
]
```