PEC 1. Desarrollo front-end con framew. JavaScript

Ejercicio 1

Ignacio Casares Ruiz

Marzo de 2023

1

¿Cuál es la ventaja del uso de etiquetas semánticas? Nombra y explica al menos 3 de estas ventajas.

Las etiquetas en HTML permiten establecer los elementos que conformarán la página. Estas permiten:

- Estructurar. Dividir los elementos en etiquetas permiten que la página adquiera una estructura coherente y lógica a partir de la cual se desarrollará la misma.
- Identificar. Las etiquetas poseen nombres específicos que ayudarán a identificar la funcionalidad de cada elemento. Esto permitirá, tanto a desarrolladores como a revisores, tener un índice visual a partir del cual podrán reconocer la utilidad que tiene cada elemento.
- Configurar. Las etiquetas pueden ser configuradas de forma que se les puede otorgar clases o atributos que permiten su modificación y configuración, con el objetivo de obtener un mayor nivel de personalización del elemento.

Cita al menos 3 `APIs HTML5` y explica brevemente su funcionalidad.

- La API de geolocalización (Geolocation API) permite al usuario proporcionar información sobre su localización a aplicaciones web.
- HTML Drag and Drop API. Esta permite poder arrastrar elementos y colocarlos en un elemento diferente utilizando el ratón.
- HTML Sanitizer API. Esta permite convertir `strings` no fiables en HTML y modificarlos para que se puedan insertar de forma segura en el DOM de un documento.

Cita qué opción `CSS3` para conseguir que se apliquen diferentes estilos `CSS` sobre el mismo elemento en su visualización en diferentes dispositivos (diferentes tamaños de pantalla).

Esto puede realizarse a través de la inclusión de `media` *queries*. Estas pueden configurarse de forma que a un elemento se le apliquen diferentes reglas en función de lo establecido en la *query*. Si queremos, por ejemplo, ajustar el peso (propiedad *font-weight*) de la letra en función de la tamaño de la pantalla realizaríamos la siguiente `media` *query*:

```
@media only screen and (min-width: 641px) {
    h1 {
        font-weight: bold;
    }
}
```

Esta regla se aplicará a aquellos dispositivos con una anchura mínima de 641 píxeles, y aplicará la declaración `font-weight: bold;` a los elementos `h1`.

Cita al menos 4 de las características principales de `TypeScript` (importante superset de `Javascript` que trataremos en el siguiente capítulo)

- TypeScript es simplemente JavaScript. Esto quiere decir que TypeScript proporciona la estructura básica para poder construir algo que eventualmente se definirá en JS. Por ello, todo el código TS acabará siendo convertido a JS a la hora de su ejecución.
- TypeScript proporciona tipos. JavaScript no tiene tipos por defecto, lo cual puede dar problemas en proyectos de gran complejidad. Esta desventaja es compensada a través de JavaScript, que permite la utilización de esta característica.
- TypeScript permite el uso de librerías de JS. Al utilizar JS como fundamento, todas las librerías utilizables en JS pueden ser reusadas en TS.
- Javascript también es TypeScript. Esto quiere decir que todo código escrito en JS puede ser renombrado como TS y compilado junto con el resto de código TS.

2

El lenguaje CSS es muy rígido, poco práctico y ordenado a la hora de programar. Para evitar este problema se han creado los preprocesadores `CSS`, que ofrecen evidentes ventajas.

- **Cita al menos 2 de estos preprocesadores**.

Dos de los preprocesadores CSS más utilizados y reconocidos pueden ser Sass y Less.

- **Cita al menos 4 ventajas que ofrecen estos preprocesadores**

Sass permite crear funciones, algo que no es posible en vanilla CSS. Esto habitúa la posibilidad de crear estilos complejos definidos a través de lo descrito en la función. Un ejemplo sería:

```
@function pow($base, $exponent) {
  $result: 1;
  @for $_ from 1 through $exponent {
    $result: $result * $base;
  }
  @return $result;
}

.sidebar {
  float: left;
  margin-left: pow(4, 3) * 1px;
}
```

Sass permite aplicar las mismas propiedades a diferentes reglas a través de la opción de `mixins`. Estos, permiten, además, la utilización de argumentos. Ejemplo:

```
=left($dist)
  float: left;
  margin-left: $dist;

#data
  +left(10px);
```

Sass permite también la utilización de *nesting*, lo cual no es posible en CSS. Ejemplo:

```
table.hl {
  margin: 2em 0;
  td.ln {
    text-align: right;
  }
}
```

Adicionalmente, los preprocesadores CSS permiten mantener el código a largo plazo de forma más sencilla. Al ser este compilado antes de ser convertido a CSS, estos aplicarán cualquier cambio que haya sido introducido al estándar CSS y eliminará o adaptará el código que se encuentre obsoleto durante el proceso de compilación.

- **Explica brevemente en qué consisten los sourcemaps**

Un sourcemap se usa en el contexto de utilización de aplicaciones que realicen una transformación de nuestro código. Cuando esto ocurre, es difícil realizar debugging para saber cómo funciona. Los sourcemaps permiten establecer un vínculo entre el origen del código y su expresión en el código ya transformado. Por ejemplo, al utilizar el module bundler Webpack, el código final generado dista mucho del original en cuanto a que ha realizado una serie de aplicaciones sobre este. Sin embargo, Webpack ofrece archivos de sourcemap de forma que este código pueda identificarse en caso de necesitar debugging.

- **Explica qué es un transpilador**

Un transpilador es un programa que toma como origen código escrito en un lenguaje y realiza modificaciones sobre este o lo traduce a un lenguaje distinto, siempre dentro de lenguajes que operen en el mismo nivel de abstracción aproximado.

3

El flujo de trabajo profesional en front-end hace indispensable el uso de herramientas como controles de versiones y herramientas de gestión de módulos

- **Cita al menos dos sistemas de control de versiones y dos herramientas de gestión de módulos**

Entre los sistemas de control de versiones podemos mencionar Git y Mercurial como dos ejemplos.

Dos herramientas de gestión de módulos serían Parcel y Webpack.

- **Cita y explica al menos 3 comandos de `Git`**

`git add` añade los últimos cambios realizados a la fase *staged*. `git commit` guarda los cambios en tu repositorio local, incluyendo una breve descripción de los cambios que se han hecho. `git push` sube los cambios realizados en el repositorio local al repositorio remoto.

- **Cita y explica brevemente las características más definitorias de `WebPack`.**

Webpack es un *module bundler* que permite organizar tu proyecto en módulos. Webpack toma los módulos con dependencias y genera *assets* estáticos con la información que se encuentra en esos módulos.