PEC 1

Introducción al desarrollo frontend…

Universitat Oberta de Catalunya

Información relevante:

- Fecha límite de entrega: 13 de octubre.
- Peso en la nota de FC: 12%.

Contenido

Información docente

Presentación

Esta práctica se centra en conocer las tecnologías Web actuales, así como en aprender a configurar un entorno de trabajo para comenzar a desarrollar aplicaciones web. Además, se repasarán conceptos de programación aplicados a `JavaScript` utilizando la versión ES5 (2009) del estándar de `JavaScript`.

Objetivos

Los objetivos que se desean lograr con el desarrollo de esta PEC son:

- **Conozcas las tecnologías web actuales**, y dónde se ubica cada una de ellas en el stack tecnológico.
- Instales y configures un **entorno de desarrollo** completo**.**
- **Repases conceptos de `JavaScript (ES5)`** a través de prácticas guiadas.
- Te familiarices con la **dinámica de la asignatura**: metodología, tipología de enunciados, documentación a realizar, nomenclaturas, etc.

Enunciado

Esta PEC contiene 2 tareas (actividades no evaluables) y 2 ejercicios (sí evaluables). Debes entregar tu solución de los 2 ejercicios evaluables (ver el último apartado).

Debido a que las actividades están encadenadas (i.e. para hacer una se debe haber comprendido la anterior), **es altamente recomendable hacer las tareas y ejercicios en el orden en que aparecen en este enunciado**.

Tarea 1 – Configuración del entorno desarrollo

En esta primera PEC se va a configurar un entorno de desarrollo básico para comenzar a trabajar con `JavaScript`.

- **`Visual Studio Code`**
- **`Node.js y npm.`**

Para realizar esta configuración, se facilita el siguiente recurso multimedia que puedes seguir si tienes alguna duda con la instalación de `Visual Studio Code` o `node.js`:

- https://learning.oreilly.com/videos/the-complete-node-js/9781789955071/9781789955071-video2_2

Recuerda que como miembro de la Comunidad UOC, tienes disponible el catálogo de recursos de la editorial O'Reilly. Para acceder debes seguir los siguientes pasos:

* Acceder a *https://*learning.*oreilly.com*

* Cuando te pida el *login* y *password*, escribir en el campo *Email Address or Username vuestro* correo electrónico de la UOC, p.ej. dgarciaso@uoc.edu

* Después pulsa fuera con el ratón y desaparecerá el campo *password*. Además, el botón de *Sign In* cambiará por un botón rojo con el texto *Sign In with Single Sign On*.

* A partir de aquí sigue los pasos que te vaya pidiendo, seguramente te llevará a una página de autenticación con el logo de la UOC. Además, te llegará un e-mail a tu cuenta de correo electrónico de la UOC avisando de que te has suscrito a este catálogo.

* Una vez tengas acceso al catálogo de O'Reilly, para ver el contenido de los enlaces que os proporcionamos, sólo tienes que estar autenticados en O'Reilly, abrir una pestaña nueva en vuestro navegador y copiar el enlace que te facilitamos.

En los vídeos de las prácticas guiadas del ejercicio 2, el autor usa algunas extensiones de `Visual Studio` como:

- `Auto-rename-tag`
- `Bracket pair colorizer 2`
- `Javascript (ES6) Code Snippets`
- `Live Server`

Todas son opcionales, pero esta última de -`Live Server`- la recomiendo instalar ya que permitirá ir viendo el resultado de vuestro `Html, Css, Javascript` según vayas codificando y guardando sin tener que recargar el navegador.

El autor de los videos del ejercicio 2, también se ayuda mucho de `Emmet`, esta extensión ya forma parte de `Visual Studio Code` y no es necesario instalar nada, podéis consultar cómo funciona en `Visual Studio Code` en el siguiente enlace a la documentación:

https://code.visualstudio.com/docs/editor/emmet

La veréis en uso en las prácticas guiadas del ejercicio 2, es bastante útil y ahorra bastante tiempo.

Tarea 2 – Configuración del entorno de git y creación del fichero README.md

Antes de continuar debes:

> Haber leído los recursos teóricos disponibles en la PEC sobre `git` y `Markdown` y haber realizado todos los pasos de la tarea 1.

Si no tienes instalado `GIT`, Desde la web oficial http://git-scm.com , debes descargar e instalar la última versión disponible para tu sistema operativo. Sigue los pasos por defecto de la instalación, no es necesario ninguna configuración especial.

Una vez instalado, abre un terminal del sistema operativo o de `git` y ejecuta los siguientes comandos de configuración inicial:

```
git config --global user.name “[nombre apellido1 apellido2]”
```

Este comando servirá para configurar tu nombre en el repositorio.

```
git config --global user.email “[email-UOC]”
```

Este comando servirá para configurar tu email de alumno UOC en el repositorio.

Podéis comprobar que se han realizado correctamente esta configuración con este otro comando:

```
git config –list
```

Ahora en el directorio de trabajo, crea una carpeta PEC1, abre una terminal, y muévete al directorio creado ejecutando `git init` en esa ruta.

Si todo ha funcionado correctamente tendrás un directorio oculto `.git` en esa ruta. Este directorio contendrá el repositorio de todo el trabajo y commits realizados para la PEC1 y debe entregarse comprimido para la evaluación de la PEC.

Ahora en la raíz del directorio PEC1, crea un fichero `README.md` que contendrá al menos esta información:

- Login UOC
- Nombre y apellidos del alumno.
- Breve descripción de lo realizado en esta PEC, dificultades, mejoras realizadas, si hay que tener algo en cuenta a la hora de corregir/ejecutar la practica o cualquier aspecto que queráis destacar.

Una vez creado el fichero `README.md` debes añadir este fichero al repositorio `git` de la PEC. Lo puedes hacer abriendo una terminal en la ruta de la PEC y ejecutando los comandos:

```
git add README.md
git commit -m “Added README.md file”
```

Cada vez que se actualice este u otro documento se deberá actualizar y realizar commit correspondiente en el repositorio local de `git` de la PEC.

Ejercicio 1 – Preguntas teóricas (2 puntos)

Antes de continuar debes:

Haber leído los recursos teóricos disponibles en la PEC y haber realizado todos los pasos de la tarea 1 y 2.

En la raíz del directorio `PEC1` crea una nueva carpeta `PEC1_Ej1`, después crea en su interior un fichero `S01_PEC1_Solucion_Ejercicio_1.md` que contenga los enunciados y las respuestas a estas preguntas:

La aparición de `HTML5/CSS3/JS` ha supuesto el nacimiento del desarrollo front-end moderno. (0.75 puntos)

- ¿Cuál es la ventaja del uso de etiquetas semánticas? Nombra y explica al menos 3 de estas ventajas.
- Cita al menos 3 `APIs HTML5` y explica brevemente su funcionalidad.
- Cita qué opción ofrece `CSS3` para conseguir que se apliquen diferentes estilos `CSS` sobre el mismo elemento en su visualización en diferentes dispositivos (diferentes tamaños de pantalla).
- Cita al menos 4 de las características principales de `TypeScript` (importante superset de `JavaScript` que trataremos en el siguiente capítulo).

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 (0.5 puntos)
    - Cita al menos 2 de estos preprocesadores.
    - Cita al menos 4 ventajas que ofrecen estos preprocesadores.
    - Explica brevemente en qué consisten los sourcemaps.
    - Explica qué es un transpilador.

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 (0.75 puntos).

    - Cita al menos dos sistemas de control de versiones y dos herramientas de gestión de módulos.
    - Cita y explica al menos 3 comandos de `Git`.
    - Cita y explica brevemente las características más definitorias de `WebPack`.

Ejecuta los comandos `git` necesarios para añadir el fichero `S01_PEC1_Solucion_Ejercicio_1.md` y los cambios que realices en él al repositorio local `git` de la PEC.

Ejercicio 2 – Prácticas guiadas (8 puntos)

Estas prácticas son guiadas mediante la visualización de vídeos, pero se valora positivamente cuando se hagan mejoras sobre el ejercicio propuesto.

1. Validador de formularios (2 puntos).

   En la raíz del directorio `PEC1` crea una nueva carpeta `PEC1_Ej2_1` e incluye los ficheros con el código, imágenes, … necesarios para las siguientes tareas:

   - Construir un formulario HTML con varios campos a completar (Username, Email, Password, Confirm Password), sobre estos campos debes realizar las diferentes validaciones del vídeo. (0,75 punto)

Vídeo ejemplo para seguir:

https://learning.oreilly.com/videos/20-web-projects/9781800563049/9781800563049-video2_1

   - Añade un nuevo campo para edad, se debe comprobar que la edad es mayor o igual a cero e inferior a 1000. (0,25 puntos)

   - Añade las siguientes validaciones sobre el Password:

     - Usa 8 caracteres como mínimo.
     - Debe contener mayúsculas, signos, minúsculas, cifras.

     Signos permitidos: ` ~ ! @ # $ % ^ & * ( ) _ + - = { } | [ ] \ : " ; ' < > ? , . /

     (1 punto)

Durante el desarrollo del ejercicio ejecuta los comandos `git` necesarios para añadir esta carpeta, ficheros y los cambios que realices en ellos al repositorio local `git` de la PEC.

2. Calculadora de tipos de cambio de moneda (2.5 puntos)

En la raíz del directorio `PEC1` crea una nueva carpeta `PEC1_Ej2_2` e incluye los ficheros con el código, imágenes, … necesarios para las siguientes tareas:

Hacer uso de código `JavaScript` para llamar a una API de terceros. Esta Api nos servirá para que facilitándole una moneda base, nos devuelva el cambio al que cotiza esa moneda base respecto a otras muchas. (1 punto)

A parte de lo que se realiza en el vídeo debéis introducir las siguientes mejoras (1.5 puntos):

- Cuando se realiza la consulta al API, indicar estado de espera mediante logo o mensaje por pantalla.
- Si se produce un error al hacer la consulta al API mostrar un mensaje con el error.
- No permitir números negativos.

El objetivo de esta parte práctica es realizar peticiones asíncronas con `fetch` a un API de terceros).

En el vídeo se usa la versión 4 de la API de api.exchangerate-api.com, actualmente van por la Versión 6, no obstante, la versión 4 todavía funciona y es válida, podéis hacerlo con cualquiera de estas versiones del API o incluso con otra API que ofrezca este servicio.

Vídeo ejemplo para seguir:

https://learning.oreilly.com/videos/20-web-projects/9781800563049/9781800563049-video5_1

Al igual que en el resto de los ejercicios, durante el desarrollo de este ejecuta los comandos `git` necesarios para añadir la carpeta, ficheros y los cambios que realices en ellos al repositorio local `git` de la PEC.

3. Reserva de asientos de cine con conversor de tipo de moneda (3.5 puntos).

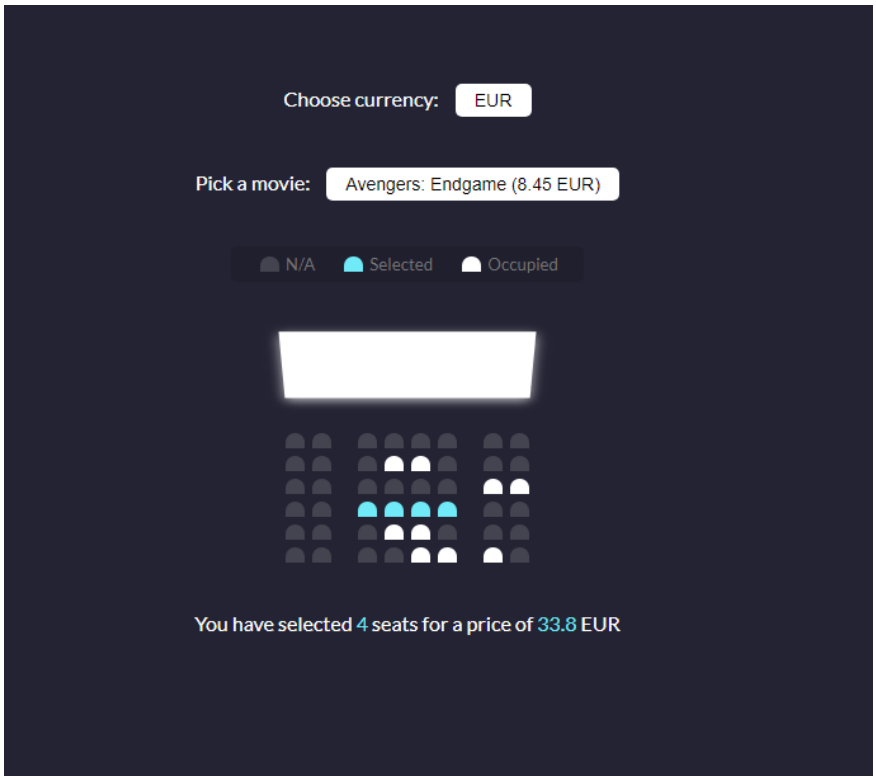
   En la raíz del directorio `PEC1` crea una nueva carpeta `PEC1_Ej2_3` e incluye los ficheros con el código, imágenes, … necesarios para las siguientes tareas:

- Realiza una pequeña aplicación para la reserva de asientos de películas, se utilizará almacenamiento local del navegador (es similar a utilizar un fichero de texto, pero almacenado en el navegador). (1.5 puntos)

Vídeo ejemplo para seguir:
https://learning.oreilly.com/videos/20-web-projects/9781800563049/9781800563049-video3_1

- Integra esta parte con la Calculadora de tipos de cambio de moneda de forma que la aplicación realizada para reserva de asientos de cine permita seleccionar la moneda en la que trabajara la aplicación y se ofrezcan tanto los precios de las películas, como los precios finales en la moneda seleccionada. (2 puntos)

  Por ejemplo, sería algo como esto:



Debes incluir, un desplegable (o similar) que permita seleccionar la moneda y al cambiar la moneda seleccionada debes hacer:

- Que cambie el importe de la entrada, si antes la entrada de *Avengers: Endgame*, eran 10 dólares, ahora al seleccionar la moneda EUR se aplicara la conversión. En el ejemplo he supuesto que 10 dólares equivalen a 8.45 EUR, esta conversión debe realizarla online el programa consultando el API usada en el ejercicio 2.
- También debe cambiar el importe total de las entradas. Si antes 4 entradas tenían un precio de 40 dólares, ahora debe mostrar el equivalente en la moneda seleccionada, en el ejemplo 33.8 EUR.

Al igual que en el resto de los ejercicios, durante el desarrollo de este ejecuta los comandos `git` necesarios para añadir la carpeta, ficheros y los cambios que realices en ellos al repositorio local `git` de la PEC.

Formato y fecha de entrega

Tienes que entregar un fichero `*.zip`, cuyo nombre tiene que seguir este patrón: `loginUOC_PEC1.zip`. Por ejemplo: `dgarciaso_PEC1.zip`. Este fichero comprimido tiene que incluir los siguientes elementos:

La **carpeta PEC1**, y en su interior:

- La **carpeta oculta `.git`** con el contenido del repositorio local `git`.
- Una **carpeta `PEC1_Ej1`** con el documento **`S01_PEC1_Solucion_Ejercicio_1.md`** que contenga el enunciado y las respuestas a las preguntas de los tres apartados del ejercicio 1
- Una **carpeta `PEC1_Ej2_1`** con el código completado siguiendo las peticiones y especificaciones del Ejercicio 2.1.
- Una **carpeta `PEC1_Ej2_2`** con el código completado siguiendo las peticiones y especificaciones del Ejercicio 2.2.
- Una **carpeta `PEC1_Ej2_3`** con el código completado siguiendo las peticiones y especificaciones del Ejercicio 2.3.

Penalizaciones

- Entrega en otro formato que no sea el especificado (ej. en `zip`): **-0.75 puntos**
- Comprimir archivos dentro del `zip`: -**1 puntos**.
- Para cada ejercicio/apartado donde no se respete la nomenclatura exacta de las carpetas o ficheros indicados (símbolos, minúsculas, mayúsculas, etc.): **-0.75 puntos.**
- La no entrega del repositório local `git` **-3 puntos**

El último día para entregar esta PEC es el 13 de octubre de 2024 hasta las 23:59.