Grado en Ingeniería Informática del Software

Software y estándares para la Web

P4. COMPUTACIÓN EN EL CLIENTE WEB (SESIÓN 1 ECMASCRIPT)

Escuela de Ingeniería Informática
Universidad de Oviedo

Contenido

RECUERDA:

La práctica se entrega en un único archivo empaquetado, con una estructura Ejercicio1, Ejercicio2, etc.

Cada ejercicio puede dividirse en tareas, que se presentará en carpetas separadas: tarea1, tarea2, etc.

Objetivos y requisitos comunes a todos los ejercicios

En esta práctica el objetivo es hacer computación en el cliente Web usando el estándar ECMAScript.

En esta práctica se usará ECMAScript "puro" sin bibliotecas ni extensiones fuera del estándar ECMAScript.

Se usarán objetos obligatoriamente, con clases o sin clases, para todos los ejercicios. Los programas desarrollados deberán siempre usar objetos y no se permitirá código no orientado a objetos (por ejemplo: funciones libres no ligadas a objetos o clases, paradigma procedimental).

Los archivos HTML, CSS y ECMAScript deben estar siempre separados y no incrustados en el archivo HTML. El objetivo es tener en archivos separados el contenido, la presentación y la computación.

Se debe comprobar la validez de los archivos construidos HTML5 y CSS con los validadores del W3C.

Los ejercicios desarrollados deben funcionar en todos los navegadores de referencia: Firefox, Edge, Chrome, Opera, Safari en los Mac e IOS, Chrome en Android, etc.

Esta práctica se corresponde con los temas de teoría:

- Computación Web
- Lenguajes de Script
- El lenguaje JavaScript

Ejercicio 1

Tarea 1. Escribir un archivo en HTML5 denominado **Ejercicio1** (con extensión **.html**) que referencie a un archivo CSS denominado **Ejercicio1** (con extensión **.css**) y también debe referenciar a varios archivos en ECMAScript (extensión **.js**). El archivo en HTML5 deberá contener:

- Incluir en cabeza (head) un archivo ECMAScript denominado **Cabecera.js**
- En el cuerpo (body) incluir un archivo ECMAScript denominado **Titulo1.js**
- En el cuerpo (body) incluir un archivo ECMAScript denominado **Titulo2.js**
- En el cuerpo (body) incluir un archivo ECMAScript denominado **Titulo3.js**
- En el cuerpo (body) incluir un archivo ECMAScript denominado **Titulo4.js**
- En el cuerpo (body) incluir un archivo ECMAScript denominado **Parrafos.js**

Tarea 2. Escribir un archivo CSS para el HTML de la tarea anterior que especifique el estilo de h1, h2, h3, h4, p y el fondo. Se deja libre al estudiante elegir el diseño de la hoja de estilo. El archivo se denominará **Ejercicio1.css**

Tarea 3. Escribir un archivo ECMAScript denominado **Cabecera.js** que contiene un objeto con información del nombre de la asignatura, nombre de la titulación, nombre del centro donde se imparte, nombre de la Universidad, curso actual, nombre del estudiante y e-mail.

Tarea 4. Escribir un archivo ECMAScript denominado **Titulo1.js** que llame a un método que escriba en el objeto **document** el nombre de la asignatura en un encabezado de nivel 1 (h1).

Tarea 5. Escribir un archivo ECMAScript denominado **Titulo2.js** que llame a un método que escriba en el objeto **document** el nombre de la titulación en un encabezado de nivel 2 (h2).

Tarea 6. Escribir un archivo ECMAScript denominado **Titulo3.js** que llame a un método que escriba en el objeto **document** el nombre del centro donde se imparte la titulación en un encabezado de nivel 3 (h3).

Tarea 7. Escribir un archivo ECMAScript denominado **Titulo4.js** que llame a un método que escriba en el objeto **document** el nombre de la universidad donde se imparte la titulación en un encabezado de nivel 4 (h4).

Tarea 8. Escribir un archivo ECMAScript denominado **Parrafos.js** que llame a un método que escriba en el objeto **document** el resto de información disponible en **Cabecera.js** en párrafos (p).

Epílogo. Todas las tareas se presentan en la misma sub-carpeta de **P4** denominada **Ejercicio1** que debe contener los archivos:

- Ejercicio1.html
- Ejercicio1.css
- Cabecera.js
- Titulo1.js
- Titulo2.js
- Titulo3.js
- Titulo4.js
- Parrafos.js

Ejercicio 2

Tarea 1. Escribir un archivo en HTML5 denominado **Ejercicio2** (con extensión **.html**) que referencie a un archivo CSS denominado **Ejercicio2** (con extensión **.css**) y también debe referenciar a varios archivos en ECMAScript (extensión **.js**). El archivo en HTML5 deberá contener:

- Incluir en cabeza (head) un archivo ECMAScript denominado **InfoNavegador.js**
- Un encabezado de nivel 1 (h1), con llamada en su interior a un archivo ECMAScript denominado **NombreNavegador.js**
- Un encabezado de nivel 2 (h2), con llamada en su interior a un archivo ECMAScript denominado **IdiomaNavegador.js**
- Un párrafo (p), con llamada en su interior a un archivo ECMAScript denominado **MasInfoNavegador.js**

Tarea 2. Escribir un archivo CSS para el HTML de la tarea anterior que especifique el estilo de h1, h2, p y el fondo. Se deja libre al estudiante elegir el diseño de la hoja de estilo. El archivo se denominará **Ejercicio2.css**

Tarea 3. Escribir un archivo ECMAScript denominado **InfoNavegador.js** que contiene un objeto con información del navegador que utiliza el cliente Web.

Tarea 4. Escribir un archivo ECMAScript denominado **NombreNavegador.js** que llame a un método que escriba en el objeto **document** el nombre del navegador.

Tarea 5. Escribir un archivo ECMAScript denominado **IdiomaNavegador.js** que llame a un método que escriba en el objeto **document** el idioma del navegador.

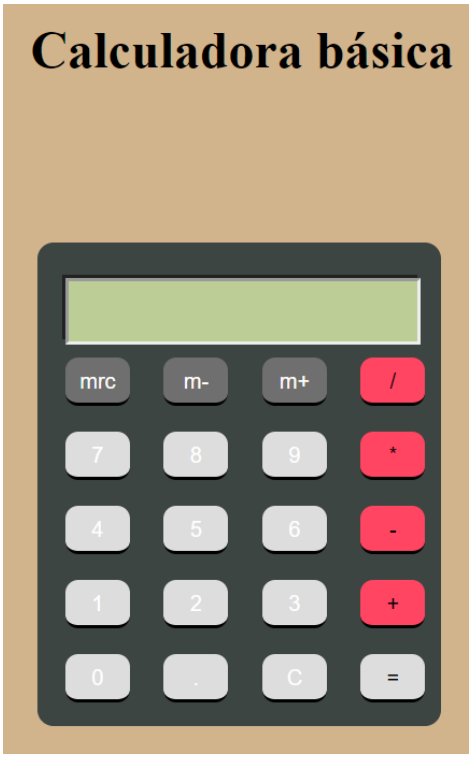
Tarea 6. Escribir un archivo ECMAScript denominado **MasInfoNavegador.js** que llame a un método que escriba en el objeto **document** el resto de información disponible del navegador.

Epílogo. Todas las tareas se presentan en la misma sub-carpeta de **P4** denominada **Ejercicio2** que debe contener los archivos:

- Ejercicio2.html
- Ejercicio2.css
- InfoNavegador.js
- NombreNavegador.js
- IdiomaNavegador.js
- MasInfoNavegador.js

Ejercicio 3

Tarea 1. Escribir un archivo en HTML5 denominado **CalculadoraBasica** (con extensión **.html**) que referencie a un archivo CSS denominado **CalculadoraBasica** (con extensión **.css**) y también debe referenciar a un archivo en ECMAScript denominado **CalculadoraBasica** (extensión **.js**). El archivo en HTML5 deberá usar la metáfora de una calculadora para contener la interfaz web de una calculadora básica. La interfaz puede ser de la forma siguiente:



Tarea 2. Escribir un archivo CSS denominado **CalculadoraBasica** (con extensión **.css**) con la hoja de estilo de la interfaz web de la calculadora básica.

Tarea 3. Escribir un archivo ECMAScript denominado **CalculadoraBasica** (con extensión **.js**) con una clase denominada Calculadora con los atributos y métodos necesarios para realizar las operaciones de la calculadora.

Epílogo. Todas las tareas se presentan en la misma sub-carpeta de **P4** denominada **Ejercicio3** que debe contener los archivos:

- CalculadoraBasica.html
- CalculadoraBasica.css
- CalculadoraBasica.js

Ejercicio 4

Tarea 1. Escribir un archivo en HTML5 denominado **CalculadoraCientífica** (con extensión **.html**) que referencie a un archivo CSS denominado **CalculadoraCientifica** (con extensión **.css**) y también referencia a un archivo en ECMAScript denominado **CalculadoraCientifica** (extensión **.js**).

Debe utilizarse la herencia de la clase calculadora básica del ejercicio anterior.

El archivo en HTML5 deberá usar la metáfora de una calculadora para contener la interfaz web de una calculadora científica. La interfaz puede emular a la calculadora científica de Windows 10:

Tarea 2. Escribir un archivo CSS denominado **CalculadoraCientífica** (con extensión **.css**) con la hoja de estilo de la interfaz web de la calculadora científica.

Tarea 3. Escribir un archivo ECMAScript denominado **CalculadoraCientifica** (con extensión **.js**) con una clase denominada Calculadora con los atributos y métodos necesarios para realizar las operaciones de la calculadora.

Epílogo. Todas las tareas se presentan en la misma sub-carpeta de **P4** denominada **Ejercicio4** que debe contener los archivos:

- CalculadoraCientifica.html
- CalculadoraCientifica.css
- CalculadoraCientifica.js

Ejercicio 5

Tarea 1. Escribir un archivo en HTML5 denominado **CalculadoraRPN** (con extensión **.html**) que referencie a un archivo CSS denominado **CalculadoraRPN** (con extensión **.css**) y también referencia a un archivo en ECMAScript denominado **CalculadoraRPN** (extensión .**js**).

Las calculadoras RPN utilizan la notación postfija o notación polaca inversa (reverse polish notation). Evalúan las expresiones utilizando una pila. No tienen el botón igual.

La calculadora debe tener además de las funciones básicas (+,-,*,/), funciones científicas (sin, cos, tan,…). Se deja libre al estudiante el diseño de las funciones incorporadas.

Epílogo. Todas las tareas se presentan en la misma sub-carpeta de **P4** denominada **Ejercicio5** que debe contener los archivos:

- CalculadoraRPN.html
- Calculadora.css
- Calculadora.js

Ejercicio 6

Tarea 1. Realización de una aplicación web, utilizando HTML5, CSS y ECMAScript. La aplicación web es de **temática libre**.

Se valorará la presentación, la complejidad de la aplicación, la originalidad, la creatividad y los elementos usados de ECMAScript.

En este ejercicio **no** pueden utilizarse bibliotecas como *jQuery* y otras.

Debe usarse ECMAScript "puro".

Debe ser totalmente orientada a objetos.

Epílogo. Todos los archivos se presentan en la misma sub-carpeta de **P4** denominada **Ejercicio6**