

UD 03. MÓDULOS Y HERRAMIENTAS DE DESARROLLO

Desarrollo Web en entorno cliente
CFGS DAW

Ejercicios

Álvaro Maceda Arranz
alvaro.maceda@ceedcv.es

2024/2024

Versión:241013.2011


Licencia




Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante

ÍNDICE DE CONTENIDO

1. Ejercicio 1: Incrementar contador.....	3
2. Ejercicio 2: Puntuación de tenis.....	3
3. Ejercicio 3: Publicación de un módulo npm.....	3

UD03. MÓDULOS Y HERRAMIENTAS DE DESARROLLO

Para cada uno de los ejercicios de este módulo deberás crear un entorno de desarrollo con Vite. Los fuentes estarán en el directorio `src` excepto `index.html` que estará en el raíz.

1. EJERCICIO 1: INCREMENTAR CONTADOR

Crea un módulo que implemente una función `incrementar()` que devuelva un número empezando por 1 y que se vaya incrementando en cada llamada, y otra función `reset()` para reinicializarlo.

Crea una página web que tenga un único `<p>` con el contenido inicial 0, un botón “Incrementar” y otro botón “Reset”.

Crea otro módulo que implemente la funcionalidad necesaria para hacer que, al pulsar el botón, el párrafo se incremente o se resetee a cero.

2. EJERCICIO 2: PUNTUACIÓN DE TENIS

Construye un módulo para mantener la puntuación de un juego en un partido de tenis.

El módulo exportará tres funciones:

- `iniciarJuego()`: Inicializa la puntuación del juego.
- `puntoJugador(jugador)`: Indica que uno de los jugadores ha ganado el punto. Si el juego había finalizado, lanzará una excepción. El parámetro `jugador` será `1` para el primer jugador y `2` para el segundo. Si se pasa un número de jugador diferente lanzará una excepción con el mensaje `'Jugador inválido'`. Si el juego había finalizado se lanzará la excepción `'El juego ya ha finalizado'`.
- `resultado()`: Devuelve una cadena indicando el resultado actual del juego. Según el caso debe devolver (atención a las mayúsculas y espacios):
 - Deuce: `Deuce`
 - Ventaja de un jugador: `Ventaja jugador 1`
 - En otro caso: `Jugador 1: 15 Jugador 2: 40`

La puntuación de un jugador se incrementa cada vez que gana un punto siguiendo la siguiente secuencia: 0, 15, 30, 40. Si el jugador tiene una puntuación de 40 y gana el punto, gana el juego.

Si los dos jugadores tienen 40 puntos se pasa a “deuce”. Quien gane el punto tiene “ventaja”. Si el jugador que no tiene ventaja gana el punto, se pasa de nuevo a “deuce”. Si el jugador que tiene ventaja gana el punto, ganará el juego.

Crea dos versiones: una versión que utilice CommonJS y otra versión que utilice módulos ES6 (las dos versiones deben estar en el mismo proyecto)

3.EJERCICIO 3: PUBLICACIÓN DE UN MÓDULO NPM

Crea un módulo que exporte una función. La función debe admitir un array como entrada y transformar todos sus elementos en la cadena `'banana'`.

Sigue la guía que hay en los enlaces para publicar dicho módulo en el repositorio de npm. Utiliza scopes para publicarlo como `@[TU USUARIO]/bananize`.

Una vez publicado, crea un proyecto nuevo y comprueba que puedes utilizarlo. Mira qué es lo que se ha descargado en `node_modules` cuando instalas el módulo en otro proyecto.

A continuación ofrecemos en orden alfabético el listado de autores que han hecho aportaciones a este documento:

- Álvaro Maceda Arranz