

Entrega 1- Calculadora

Versión: 27 de Enero de 2020

Objetivo

Practicar con expresiones básicas, variables, asignaciones, if...else, switch...case, bucles, los tipos number, string, boolean, eventos sencillos, arrays, así como con objetos DOM y Date.

Descripción de la práctica

En esta entrega vamos a desarrollar una aplicación web de calculadora. Para comenzar el desarrollo partimos de la versión básica de la calculadora, cuyo se proporciona al alumno. Las instrucciones para obtener este código están en el siguiente apartado. Sobre éste, el alumno tendrá que implementar las operaciones que faltan, las cuales se indican en el apartado "Tareas".

Calculadora de CORE

Número:

Info sobre el número

Operaciones unitarias

Operaciones binarias

Operaciones en formato CSV

Descargar el código del proyecto

El proyecto se descarga en el ordenador local con estos comandos:

El proyecto debe clonarse en el ordenador desde el que se está trabajando

```
$ git clone https://github.com/CORE-2020/Entrega1_Calculadora
```

Entrar en el directorio de trabajo

```
$ cd Entrega1_Calculadora
```

Tareas

Para superar esta entrega, el alumno tendrá que implementar las siguientes funcionalidades:

1. Campo informativo:

Añada al documento HTML un nuevo elemento mediante la etiqueta `<h2>` con identificador "info", clase "grande", un atributo "title" con valor "Info sobre el número" y contenido inicialmente "Info sobre el número".

El contenido de dicho elemento `<h2>` se debe actualizar cada vez que se hace un cálculo en la calculadora. En este primer paso tan solo tenemos la función "cuadrado", así que solo se actualizará el campo informativo cuando el usuario haga click en el botón "cuadrado". Su contenido debe ser "Info: El resultado es menor que 100", "Info: El resultado está entre 100 y 200" o "Info: El resultado es superior a 200" según sea el resultado del cálculo que muestre el input.

Aunque en esta primera funcionalidad solo hay una funcionalidad en la calculadora, el resto de apartados pedirá que se realicen más botones en la calculadora y el campo informativo debe seguir funcionando, esto es, mostrando un mensaje u otro según el resultado del cálculo ejecutado. Así que recomendamos hacer esta funcionalidad en una función específica (de nombre por ejemplo `rellenar_info`) que llamemos cada vez que sea necesario.

2. Operaciones unitarias:

- Añada un nuevo botón a la calculadora junto al que ya tiene. Este botón tendrá un id "modulo" y llamará a una función `mod()` que debe crear usted. Dicha función calcula el módulo del número X introducido en el input, es decir si es número es positivo lo deja como está y si es negativo devuelve como resultado $-X$
- Añada un nuevo botón a la calculadora junto al que ya tiene. Este botón tendrá un id "factorial" y llamará a una función `fact()` que debe crear usted. Dicha función calcula el factorial del número X introducido en el input. $\text{Factorial} = X \cdot (X-1) \cdot (X-2) \cdot \dots \cdot 3 \cdot 2 \cdot 1$, (calcularlo con un bucle for)

3. Operaciones binarias:

Las operaciones binarias (suma, resta, multiplicación, división, resto y potencia) consisten en dos operandos, un operador y se debe pulsar el signo igual para obtener el resultado. El funcionamiento sería como el de cualquier calculadora. Se introduce un primer número, se pulsa la operación, se introduce un segundo número y se pulsa el signo igual.

Consejos para el desarrollo: Tenemos dos pulsaciones a botones. Primera, al pulsar cualquier operador binario, debe invocarse una función, que guarde en variables globales, tanto el número

teclado en el input, como un string indicando el operador pulsado (las variables globales son visibles dentro de todas las funciones del script y puedan utilizarse para pasar valores de una función a otra). Al pulsar el botón con el signo "=" debe invocarse una nueva función que calcule el resultado, realizando la operación indicada por el operador guardado, utilizando el primer número (guardado en la variable global) y el segundo número que debe estar en el input.

- Añada un nuevo operador binario para hacer una multiplicación y otro para hacer una suma (el resto de operadores binarios se dejan opcionales). Con esto aparecen 3 botones nuevos. El primero debe tener id "multiplicacion" y llamar a una función mul() que como se ha explicado guarda el valor del input y el operador (en este caso multiplicación) en variables globales. El segundo debe tener el id "suma" y llamar a la función add() que guardará también lo necesario. El tercero debe tener el id "igual" y llamar a la función eq() que hará el cálculo adecuado según se haya pulsado antes en multiplicar o sumar y mostrará el resultado calculado en el input. La función eq() si se pulsa repetidamente no hace nada, es decir solo hace el cálculo si se ha usado antes la multiplicación o la suma.

4. Operaciones en formato CSV (Comma Separated Values)

La calculadora debe incluir operaciones con varios operandos en formato CSV, es decir valores separados por comas, por ejemplo "5,7,9,1". Añada cuatro nuevos botones, uno para hacer el sumatorio, otro que ordena la lista de números provista, otro que invierte la lista y un último que quita los dos últimos elementos.

- El botón 'sumatorio' tiene que tener un id "sumatorio" y llamar a una función sumatorio() que calcula la suma de los elementos introducidos en el input. Para un input relleno con "5,7,9,1" al hacer click en el botón "sumatorio" el input pasaría a tener el valor 22 (la suma de todos los números $5+7+9+1$).
- El botón 'ordenar' tiene que tener un id "ordenar" y llamar a una función ordenar() que ordena (en orden ascendente) los elementos introducidos en el input. Para un input relleno con "5,7,9,1" al hacer click en el botón "ordenar" el input pasaría a tener el valor "1,5,7,9". Cuidado en esta función ya que el método sort() de los arrays en JavaScript ordena strings con lo que 100 será menor que 2. Es necesario pasarle a la función sort() como callback una función que haga la comparación con números (ver: <https://alligator.io/js/array-sort-numbers/>).
- El botón 'revertir' tiene que tener un id "revertir" y llamar a una función revertir() que invierte el orden de los elementos introducidos en el input. Para un input relleno con "5,7,9,1" al hacer click en el botón "revertir" el input pasaría a tener el valor "1,9,7,5".
- El botón 'quitar' tiene que tener un id "quitar" y llamar a una función quitar() que quita el último número y la coma de los introducidos en el input. Para un input relleno con "5,7,9,1" al hacer click en el botón "quitar" el input pasaría a tener el valor "5,7,9", para un input "43,2,256,4543" quedaría "43,2,256".

5. Tratamiento de errores

Queremos que la calculadora sea un poco más robusta y funcione independientemente de lo que introduzca el usuario. Para ello tendremos que validar la entrada antes de hacer las operaciones y mostrar un error en caso de que lo introducido por el usuario no sea válido.

- Implemente la funcionalidad de tratamiento de errores. Para ello debe validar la entrada del usuario antes de hacer cualquier operación. Cree una función `validar()` y utilícela en todas las funciones que realizan operaciones. Dicha función debe coger el valor del input y comprobar que es algo soportado por la calculadora. Si es así realiza la operación, si no es algo soportado muestra "Error" en el input.

La calculadora debe soportar números enteros y decimales con el punto (positivos y negativos), y también listas de dichos números en formato CSV (separados por comas).

Consideraciones adicionales y recomendaciones

Todas las funciones que se definan y utilicen deben utilizar la notación arrow.

Se recomienda añadir algún separador adicional a los grupos de botones como muestra la imagen de ejemplo de la calculadora, así como un poco de CSS para darle algo de estilo.

El texto que contienen los botones queda a elección del alumno. Es decir el botón suma podrá contener la palabra suma o el signo +, etc.

Prueba de la práctica

Para ayudar al desarrollo, se provee una herramienta de autocorrección que prueba las distintas funcionalidades que se piden en el enunciado. Para utilizar esta herramienta debes tener node.js (y npm) (<https://nodejs.org/es/>) y Git instalados.

Para instalar y hacer uso de la [herramienta de autocorrección](#) en el ordenador local, ejecuta los siguientes comandos en el directorio del proyecto:

```
$ npm install -g autocorector      ## Instala el programa de test
$ autocorector                     ## Pasa los tests al fichero a entregar
.....                           ## en el directorio de trabajo
... (resultado de los tests)
```

También se puede instalar como paquete local, en el caso de que no se dispongas de permisos en el ordenador desde el que estás trabajando:

```
$ npm install autocorector         ## Instala el programa de test
$ npx autocorector                 ## Pasa los tests al fichero a entregar
.....                           ## en el directorio de trabajo
... (resultado de los tests)
```

Se puede pasar la herramienta de autoorrección tantas veces como se desee sin ninguna repercusión en la calificación.

Instrucciones para la Entrega y Evaluación.

Una vez satisfecho con su calificación, el alumno puede subir su entrega a Moodle con el siguiente comando:

```
$ autocorector upload
```

o, si se ha instalado como paquete local:

```
$ npx autocorector upload
```

La herramienta de autocorrección preguntará por el correo del alumno y el token de Moodle. En el enlace <https://www.npmjs.com/package/autocorector> se proveen instrucciones para encontrar dicho token.

RÚBRICA: Se puntuará el ejercicio a corregir sumando el % indicado a la nota total si la parte indicada es correcta:

- 10%:
- 10%:
- 30%:
- 30%:
- 20%:

Si pasa todos los tests se dará la máxima puntuación.

¡Cuidado! Una vez enviadas, tanto la entrega, como la evaluación, no se pueden cambiar.