

Desarrollo Web en Entorno Cliente

UD4-Programación con arrays, funciones y objetos definidos por el usuario - Actividades 02

Actualizado Septiembre 2024

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

Práctica 1	3
Práctica 2	4
Práctica 3	4
Práctica 4	4

UD4-PROGRAMACIÓN CON ARRAYS, FUNCIONES Y OBJETOS DEFINIDOS POR EL USUARIO - ACTIVIDADES 02

 **Importante 1:** no intentes copiar ejercicios ni tan siquiera “ver un poco” código de otros compañeros. Es el mayor error de quien empieza a programar, ya que luego no sabe resolver problemas por sí mismo y da una falsa sensación de aprendizaje.

 **Importante 2:** si en programación algo no sale a la primera... es totalmente normal. Es parte del aprendizaje. ¿Cómo crees que aprendieron los mejores programadores?

 **Importante 3:** Se deben comentar todos los códigos con las acciones realizadas en cada instrucción o grupo de instrucciones. Se deben guardar los ejercicios en una carpeta con tus iniciales seguidas por ud02pr1. Dentro tendrás el ejercicio resuelto con nombres practica.html etc....

PRÁCTICA 1

a) Crear un objeto llamado Hospital con las siguientes propiedades y métodos:

Propiedades:

- | | |
|-------------------------|--------------------|
| • codHospital | • telefono |
| • nombreHospital | • poblacion |
| • direccion | • codPostal |

Método:

- **Hospital:** Permite crear una instancia con los datos de un hospital.

b) Crear un objeto llamado Médico con las siguientes propiedades y métodos:

Propiedades:

- | | |
|--------------------|----------------------------|
| • codMedico | • poblacion |
| • nombre | • codPostal |
| • apellidos | • fechaNacimiento |
| • dni | • especialidad |
| • direccion | • sueldo |
| • telefono | • Hospital (objeto) |

Métodos:

- **Medico:** Permite crear una instancia con los datos de un médico.
- **cambiaSueldo:** Permite cambiar el sueldo del médico.
- **retencionMedico:** Permite calcular la retención aplicada al sueldo del médico. Se le pasan los datos sueldo y porcentaje de retención.
- **mostrarDatos:** Muestra los datos del médico y del hospital en el que trabaja.

- c) Crear una instancia del objeto Hospital llamado hospitalRibera.
- d) Crear dos instancias del objeto llamado Medico llamados medicoDigestivo y medicoTraumatologo que pertenecen al hospitalRibera.
- e) Cambia el sueldo del medicoTraumatologo por 2.300,00 €
- f) Mostrar los datos de la instancia hospitalRibera del objeto Hospital.
- g) Calcula la retención de las dos instancias del objeto Medico que hemos creado.
- h) Mostrar los datos de las dos instancias del objeto Medico que hemos creado, así como las retenciones y los sueldos finales de cada una de las instancias del objeto.

PRÁCTICA 2

Realizar un script que permita:

- Introducir por teclado números enteros, hasta leer el valor cero.
- Escribir la longitud del array.
- Escriba en el documento los elementos en orden inverso y eliminándoles.
- Por último, escribir la longitud del array que deberá ser cero

PRÁCTICA 3

Implementar una página que facilite las citas médicas, basada en arrays paralelos vistos anteriormente y que aplique funciones para organizar las tareas a realizar.

- Presentaremos una pantalla, con los siguientes elementos:
 - Citarse: Botón para que el paciente pulse y dé su nombre y número de consulta.
 - Siguiendo a consulta N: Un botón por consulta, donde el doctor pulsa para siguiente paciente.
 - La lista de pacientes llamados a cada consulta se visualizará permanentemente en pantalla.
- Crearemos una estructura de datos como el siguiente:
 - Consultas: array con el nombre de los doctores (se inicializará por código en el script)
 - Pacientes: array con el nombre de los pacientes (inicialmente vacío).
 - Citas: array bidimensional con una fila por consulta y una columna por paciente citado (se creará dinámicamente)
- Diseñar una pantalla de gran tamaño, que resulte accesible y usable basada en CSS.

PRÁCTICA 4

Definir la estructura de un objeto que implemente el documento de factura.

- Estará formada por información que dispondremos en las siguientes propiedades

del objeto:

- Empresa (nombre de la empresa, dirección, teléfono, NIF),
- Cliente (nombre del cliente, dirección, teléfono, NIF),
- Detalle de factura como lista de elementos (descripción, precio, cantidad)
- Totales de la factura (base Imponible, importe total, tipo de iva, forma de pago).
- Una vez definidas las propiedades del objeto, añadir los siguientes métodos:
 - El constructor, que tendrá el siguiente prototipo de función:
function Factura (empresa, cliente, detalle, tiva, formaPago)
 - calculaTotal calculará el importe total de la factura y actualizará las propiedades correspondientes.
 - muestraTotal mostrará el importe total de la factura, el iva aplicado y la forma de pago.
 - imprimirFactura la imprimirá formateándola adecuadamente en el documento actual (document)

PRÁCTICA 5

Queremos hacer una aplicación en JavaScript para gestionar edificios con la información de la situación del edificio y de los propietarios de cada piso. Para ello queremos almacenar la siguiente información de cada edificio:

- calle.
- número.
- código postal.
- plantas del edificio (dentro de cada planta tendremos un número de puertas y para cada puerta almacenaremos el nombre del propietario).

Se pide:

- Crear un objeto que nos permita instanciar edificios. Cada vez que instanciamos un edificio le pasaremos la calle, número y código postal como parámetros. Se pide además crear los siguientes métodos para el objeto Edificio:
 - agregarPlantasYPuertas(numplantas, puertas) // Se le pasa el número de plantas que queremos crear en el piso y el número de puertas por planta. Cada vez que se llame a este método, añadirá el número de plantas y puertas indicadas en los parámetros, a las que ya están creadas en el edificio.
 - modificarNumero(numero) // Se le pasa el nuevo número del edificio para que lo actualice.
 - modificarCalle(calle) // Se le pasa el nuevo nombre de la calle para que lo actualice.
 - modificarCodigoPostal(codigo) // Se le pasa el nuevo número de código postal del edificio.
 - imprimeCalle // Devuelve el nombre de la calle del edificio.
 - imprimeNumero // Devuelve el número del edificio.
 - imprimeCodigoPostal // Devuelve el código postal del edificio.

- agregarPropietario(nombre,planta,puerta) // Se le pasa un nombre de propietario, un número de planta y un número de puerta y lo asignará como propietario de ese piso.
 - imprimePlantas // Recorrerá el edificio e imprimirá todos los propietarios de cada puerta.
- Cada vez que se crea un edificio que muestre automáticamente un mensaje del estilo:
 - construido nuevo edificio en calle: xxxxxx, nº: xx, CP: xxxxx.
- Cada vez que se añada un propietario a un piso de un edificio que muestre un mensaje del estilo:
 - xxxxxxxx es ahora el propietario de la puerta x de la planta x.

Aquí se muestra un **ejemplo de lo que tendría que mostrar la aplicación**:

Trabajando con objetos Javascript:

Instanciamos 3 objetos edificioA, edificioB y edificioC con estos datos:

- Construido nuevo edificio en calle: Garcia Prieto, nº: 58, CP: 15706.
- Construido nuevo edificio en calle: Camino Caneiro, nº: 29, CP: 32004.
- Construido nuevo edificio en calle: San Clemente, nº: s/n, CP: 15705.
- El código postal del edificio A es: 15706.
- La calle del edificio C es: San Clemente.
- El edificio B está situado en la calle Camino Caneiro número 29.

Agregamos 2 plantas y 3 puertas por planta al edificio A...

Agregamos 4 propietarios al edificio A...

- Jose Antonio Lopez es ahora el propietario de la puerta 1 de la planta 1.
- Luisa Martinez es ahora el propietario de la puerta 2 de la planta 1.
- Marta Castellón es ahora el propietario de la puerta 3 de la planta 1.
- Antonio Pereira es ahora el propietario de la puerta 2 de la planta 2.

Listado de propietarios del edificio calle García Prieto número 58

- Propietario del piso 1 de la planta 1: Jose Antonio Lopez.
- Propietario del piso 2 de la planta 1: Luisa Martinez.
- Propietario del piso 3 de la planta 1: Marta Castellón.
- Propietario del piso 1 de la planta 2:
- Propietario del piso 2 de la planta 2: Antonio Pereira.
- Propietario del piso 3 de la planta 2:

Agregamos 1 planta más al edificio A...

Agregamos 1 propietario más al edificio A planta 3, puerta 2...

- Pedro Meijide es ahora el propietario de la puerta 2 de la planta 3.

Listado de propietarios del edificio calle García Prieto número 58

- Propietario del piso 1 de la planta 1: Jose Antonio Lopez.
- Propietario del piso 2 de la planta 1: Luisa Martinez.
- Propietario del piso 3 de la planta 1: Marta Castellón.
- Propietario del piso 1 de la planta 2:

- Propietario del piso 2 de la planta 2:
- Propietario del piso 1 de la planta 3:
- Propietario del piso 2 de la planta 3: Pedro Meijide.

Consejos y recomendaciones.

Intenta crear todos los métodos dentro del objeto Edificio y utiliza los métodos programados para imprimir el valor de las propiedades que se piden en el enunciado.