

Tabla de contenido

Ejercicio 1	1
Fecha límite de presentación	8
Conclusiones.....	8

Ejercicio 1

Se va a desarrollar un programa en siete pasos.

Cada uno de ellos supondrá la utilización de una nueva sentencia o estructura o método...en fin un cambio de desarrollo (una mejora) sobre el anterior.

Cada una de las siete versiones numeradas deben dar lugar a una nueva rama en GitHub:
tareaContinua1 a tareaContinua7.

1.- Elabora una lista de contactos

Inicialmente tendrá los valores de la tabla adjunta:

Declara e inicializa las variables donde se almacenará toda la información (nueve variables en total).

Muestra en la consola información sobre el primer y último contacto en el formulario:
nombre/teléfono/correo.

Nombre	Teléfono	Correo
Maxwell Wright	(0191) 719 6495	Curabitur.egestas.nunc@nonummyac.co.uk
Raja Villarreal	0866 398 2895	posuere.vulputate@sed.com
Helen Richards	0800 1111	libero@convallis.edu

En este primer ejercicio , declaro las nueve variables a capón , asignándole directamente los valores .

Una vez declaradas las variables , como ya sabemos por usabilidad y eficiencia con let .

```
let nombre1 = "Maxwell Wright";
let nombre2 = "Raja Villareal";
let nombre3 = "Helen Richards ";
let telefono1 = "(0191)7196495";
let telefono2 = "0866 398 2895";
let telefono3 = "0800 1111";
let correo1 = "Curabitur.egestas.nun@nonummyac.co.uk" ;
let correo2 = "posuere.vulputate@sed.com" ;
let correo3 = "libero@convalis.edu" ;
```

Realizo un console.log de cada uno de los valores que contiene cada usuario (persona)

```
console.log("Nombre : "+nombre1+", telefono : "+telefono1+", correo : "+correo1);
console.log("Nombre : "+nombre2+", telefono : "+telefono2+", correo : "+correo2);
console.log("Nombre : "+nombre3+", telefono : "+telefono3+", correo : "+correo3);|
```

2.- Crea la lista de contactos en un array, de tal manera que cada uno de sus elementos será un objeto que describa a un solo usuario (tres propiedades).

Al final de la lista declarada de esta manera, agrega un nuevo contacto usando un método de array, apropiado. El nuevo contacto es: Maisie Haley / 0913 531 3030 / risus.Quisque@urna.ca.

Muestra el primer y último contacto, de nuevo en el formato: nombre / teléfono / correo electrónico. (length) .

En este primer ejercicio , declaro directamente en un array que recibe el nombre de listaContactos , los usuarios que voy a contemplar .

Para ello , en cada posición les asigno creando los usuarios , añadiéndoles el nombre , teléfono y correo .

Una manera de hacerlos es la siguiente :

```
let listaContactos = [ {nombre : "Maxwell Wright",telefono: "(0191)7196495",correo : "Curabitur.egestas.nun@nonummyac.co.uk"},
  {nombre : "Raja Villareal",telefono : "0866 398 2895",correo : "posuere.vulputate@sed.com"},
  {nombre : "Helen Richards ", telefono : "0800 1111", correo : "libero@convalis.edu"}
];
```

Posteriormente , el ejercicio me pide introducir un nuevo elemento al array (nuestra listaContactos) . Y como hemos visto en clase , array tiene un método propio , que es el push , que lo que realiza es introducir el nuevo elemento al array .

```
listaContactos.push ({nombre : "Maisie Haley", telefono : "0913 531 3030", correo : "risus.Quisque@urna.ca"});
```

Y para terminar , recorro el array completo para ver dichos resultados , con un for Of , que es la manera más sencilla de recorrer un array de elementos .

```
var i = 0 ;
for (const iterator of listaContactos) {
  if(i == 0 || i == listaContactos.length-1)
  {
    console.log(iterator)
  }
  i = i + 1 ;
}
```

3.- Modifica el programa para agregar, al final de la lista, el contacto que el usuario introducirá durante la ejecución del programa. (prompt). Al final, muestra el primer y último contacto de la lista.

En este tercer ejercicio , nos piden que reusemos el programa anterior , modificando y poniendo la opción de que el nuevo usuario introducido sea a través de lo que nos proporcione el usuario .

De esta manera , reusamos código anterior , y añadimos antes del anterior push , sentencias para recibir valores para nuestros atributos que usaremos para añadir a nuestro nuevo elemento , usando el prompt .

```
let nombreIntroducido = prompt("Introduce el nombre") ;
let telefonoIntroducido = prompt("Introduce el telefono");
let correoTelefono = prompt("Introduce correo");
```

Y posteriormente , lo que tenemos que hacer es únicamente añadir al array el elemento con .push y asignando estas variables .

Y recorrerlo para ver el resultado .

```
listaContactos.push({nombre : nombreIntroducido , telefono : telefonoIntroducido , correo : correoTelefono} );

var i = 0 ;
for (const iterator of listaContactos) {
  if(i == 0 || i == listaContactos.length-1)
  {
    console.log(iterator)
  }
  i = i + 1 ;
}
```

4.- Modifica el programa para que el usuario pueda elegir lo que quiere hacer con la lista. Las opciones serán:

- Mostrar el primer contacto (primero)
- Mostrar el último contacto (último)
- Añadir un nuevo contacto (nuevo)

Al agregar un nuevo contacto, verifica si el usuario ha ingresado todos los datos necesarios. Si falta al menos uno de los tres valores (nombre, teléfono o correo electrónico), no se agrega el contacto.

En este programa , nos piden que realicemos una opción en la que podamos elegir que hacer , y que dependiendo de la elección realicemos una sentencia u otra .

En primer lugar , damos la opción de elegir que hacer , usando el prompt .

En mi caso , lo he realizado con un prompt Number , para indicar que hacer con números .

```
let elegir = Number(prompt("Selecciona que quieres hacer ,\n (1) • Mostrar el primer contacto (primero), \n "+
  " (2) • Mostrar el último contacto (último), \n (3) • Añadir un nuevo contacto (nuevo). "));
```

Teniendo esto , realizo un switch de elección , en el que por cada opción realizo una función u otra .

```
switch(elegir) {
  case 1:
    console.log(listaContactos[0]);
    window.alert(listaContactos[0].nombre);
    break;
  case 2:
    console.log(listaContactos[listaContactos.length-1]);
    window.alert(listaContactos[listaContactos.length-1].nombre);
    break;
  case 3:
    let nombreIntroducido = prompt("Introduce el nombre") ;
    let telefonoIntroducido = prompt("Introduce el telefono");
    let correoTelefono = prompt("Introduce correo");

    if(nombreIntroducido.length == 0 || telefonoIntroducido.length == 0 || correoTelefono.length == 0 )
    {
      window.alert("ERROR ");
      break;
    }
    listaContactos.push({nombre : nombreIntroducido , telefono : telefonoIntroducido , correo : correoTelefono} );
    console.log(listaContactos[listaContactos.length-1]);
    window.alert(listaContactos[listaContactos.length-1].nombre);
    break;
  default:
    console.log("ERROR");
    window.alert("ERROR");
}
```

En este caso las funciones están realizadas directamente , es decir que no llaman a ningún sitio para realizarlas .

5.- Encierra todo el programa en un bucle para que al usuario se le pregunte repetidamente qué quiere hacer. El usuario puede optar por:

- Mostrar el primer contacto (primero)
- Mostrar el último contacto (último)
- Mostrar todos los contactos (todos)
- Añadir un nuevo contacto (nuevo)
- Salir del programa (salir)

Después de ejecutar la acción seleccionada, el programa le dará la oportunidad de elegir nuevamente. El programa debe finalizar las acciones solo después de que el usuario dé un comando específico (salir).

Para ello , modifico el ejercicio anterior , cambiando sus opciones .

```
do {
  let elegir = Number(prompt("Selecciona que quieres hacer ,\n (1) • Mostrar el primer contacto (primero), \n "+
    " (2) • Mostrar el último contacto (último), \n (3) • Mostrar todos los contactos (todos) \n (4) • Añadir un nuevo contacto (nuevo) , "+
    "\n (5) Salir")); ;
}
```

Y de la misma manera que antes , realizo las funciones directamente en el propio caso .

```
switch(elegir) {
  case 1:
    console.log(listaContactos[0]);
    window.alert(listaContactos[0].nombre);
    break;
  case 2:
    console.log(listaContactos[listaContactos.length-1]);
    window.alert(listaContactos[listaContactos.length-1].nombre);
    break;
  case 3:
    var i = 0 ;
    for (const iterator of listaContactos) {
      console.log(iterator)
      i = i + 1 ;
    }
    break;
  case 4:
    let nombreIntroducido = prompt("Introduce el nombre") ;
    let telefonoIntroducido = prompt("Introduce el telefono");
    let correoTelefono = prompt("Introduce correo");

    if(nombreIntroducido.length == 0 || telefonoIntroducido.length == 0 || correoTelefono.length == 0 )
    {
      window.alert("ERROR ");
      break;
    }
    listaContactos.push({nombre : nombreIntroducido , telefono : telefonoIntroducido , correo : correoTelefono} );
    console.log(listaContactos[listaContactos.length-1]);
    window.alert(listaContactos[listaContactos.length-1].nombre);

    break;
  case 5 :
    validar=true ;
    break ;
  default:
    console.log("ERROR");
    window.alert("ERROR");
    break;
}
}while(validar!= true);
```

6.- Trata de organizar el código de tu programa (¿quitando switch?) usando funciones. Define y llama a tres funciones en los lugares apropiados:

- showContact: la función debe tomar dos argumentos; el primero es la lista de contactos y el segundo es el número de índice del contacto a mostrar; dentro de la función, verifica si se pasan los argumentos correctos, es decir, si los contactos son un array (utiliza la construcción instanceof Array para esto).
- showAllContacts: la función debe tomar un argumento, la lista de contactos; dentro de la función, verifica si el argumento dado es un array.
- addNewContact: la función debe tomar cuatro argumentos, una lista de contactos y los datos del nuevo contacto, es decir: nombre, teléfono y número; antes de agregar un nuevo contacto, verifica si el argumento pasado es un array y si los datos del nuevo contacto tienen algún valor.

En este caso , yo he decidido no quitar el switch , ya que la utilidad que da me parece buena , y he simplificado el código de manera que he creado funciones que directamente se ejecutan cuando llamamos al caso y a la función .

Quedando el código mas simplificado :

```
switch(elegir) {  
  case 1:  
    let longitud = listaContactos.length-1 ;  
    let indice = Number(prompt("¿ Que numero de elementos quieres mostrar con un máximo de "+longitud+"?"));  
    showContact(listaContactos , indice);  
    break;  
  case 2 :  
    console.log( showAllContacts(listaContactos));  
    break ;  
  case 3:  
    let nombreIntroducido = prompt("Introduce el nombre" ) ;  
    let correoTelefono = prompt("Introduce correo");  
    let telefonoIntroducido = prompt("Introduce el telefono");  
    addNewContact (listaContactos , nombreIntroducido , correoTelefono , telefonoIntroducido);  
    break;  
  case 5 :  
    validar=true ;  
    break ;  
  default:  
    console.log("ERROR");  
    window.alert("ERROR");  
    break;  
}  
}while(validar!= true);
```

Así en el caso del switch .

Y en el caso de las llamadas de las funciones :

```
function showContact(listaContactos , indice)
{
    /* let i = 0 ;
    for (const iterator of listaContactos) {
        console.log(iterator)
        alert(iterator);
        i = i + 1 ;
    }
    */
    if(listaContactos instanceof Array){
        for(let i = 0 ; i < listaContactos.length ; i++)
        {
            if(i < indice )
            {
                console.log(listaContactos[i]);
            }
        }
    }
}
```

```
function showAllContacts(listaContactos)
{
    console.log(listaContactos instanceof Array);
}

function addNewContact (listaContactos , nombreIntroducido , correoTelefono , telefonoIntroducido)
{
    if(listaContactos instanceof Array)
    {
        if(nombreIntroducido.length == 0 || telefonoIntroducido.length == 0 || correoTelefono.length == 0 )
        {
            window.alert("ERROR ARGUMENTOS INVALIDOS . ");
        }
        else
        {
            listaContactos.push({nombre : nombreIntroducido , telefono : telefonoIntroducido , correo : correoTelefono} );
            console.log(listaContactos[listaContactos.length-1]);
            window.alert(listaContactos[listaContactos.length-1].nombre);
        }
    }
    else
    {
        console.log("Los argumentos del array no son validos .");
    }
}
```

7.- Da al usuario la opción de seleccionar una acción sort de la lista. Cuando se selecciona esta opción, el usuario debería poder especificar si desea ordenar los contactos por nombre, teléfono o correo electrónico.

Para este ultimo caso , lo que tenemos que realizar es una multiple opción de poder ordenar dependiendo de porque atributo del array .

Para ello hago llamada al .sort() del propio array , y sabiendo que tiene una función de comparación muy similar al compareTo() de Java , visto el año pasado , la he implementado , comparando el primero con el segundo , el segundo con el tercero y así sucesivamente para ir intercambiando posiciones .

```
function ordenarContactos(listaContactos , seleccion)
{
    if(seleccion == 1)
    {
        let num = 0 ;
        for (const iterator of listaContactos) {
            console.log("Posición : "+num+" : "+iterator.nombre);
            num ++ ;
        }

        listaContactos.sort( function(a, b) {

            if (a.nombre > b.nombre) {
                return 1;
            }
            if (a.nombre < b.nombre) {
                return -1;
            }
            return 0;
        })

        for (const iterator of listaContactos) {
            console.log(iterator);
        }
    }

    if(seleccion == 2)
    {
        let num = 0 ;
        for (const iterator of listaContactos) {
            console.log("Posición : "+num+" : "+iterator.nombre);
            num ++ ;
        }
        listaContactos.sort( function(a, b) {

            if (a.telefono > b.telefono) {
                return 1;
            }
            if (a.telefono < b.telefono) {
                return -1;
            }
            return 0;
        })

        for (const iterator of listaContactos) {
            console.log(iterator);
        }
    }

    if(seleccion == 3)
    {
        let num = 0 ;
        for (const iterator of listaContactos) {
            console.log("Posición : "+num+" : "+iterator.nombre);
            num ++ ;
        }
        listaContactos.sort( function(a, b) {

            if (a.correo > b.correo) {
                return 1;
            }
            if (a.correo < b.correo) {
                return -1;
            }
            return 0;
        })

        for (const iterator of listaContactos) {
            console.log(iterator);
        }
    }
}
```

Entrega un pdf y los ficheros html y/o texto (acceso a github)

<https://github.com/Laannds/PR-CTICA2-PLM>

Fecha límite de presentación

28 de octubre de 2.022

Conclusiones

Para finalizar esta práctica , termino con una conclusión de ella , viendo que de manera práctica me ha servido para sentar conceptos básicos de lo que son declaración de variables , uso de arrays , y de funciones .

Viendo como ser más critico a la hora de buscar la mejor eficiencia para hacer que dicho programa sea más rápido y mas claro .

Me ha servido sobretodo para asentar conceptos básicos para el tratamiento de los arrays y de los elementos contenidos en ellos .

Y por último , ampliando un poco lo que es el uso de GitHub , para el control de versiones .