



ESCUELA POLITÉCNICA NACIONAL

Facultad de Ingeniería Software

Nombre: Sánchez Doménica

Fecha: 01/12/2023

COMPILADORES

TAREA PARTE 1

Tarea 1

Juguemos a la floristería. Declara seis variables, recordando nombrarlas según su propósito:

- el precio de una sola rosa (8) y el número de rosas que tienes (70)
- el precio de un solo lirio (10) y el número de lirios que tienes (50)
- el precio de un solo tulipán (2) y la cantidad de tulipanes que tienes (120)

Ahora declara tres variables, una para cada una de las rosas, lirios y tulipanes que tienes, en las que colocas su precio total. Inserta los valores correspondientes en las variables utilizando las variables declaradas en el paso anterior. Finalmente, declara una variable en la que almacenes el precio de todas tus flores (nuevamente, usa las variables anteriores para la inicialización). Muestra toda la información del inventario en la consola de la siguiente forma:

```
1 let precioPorRosa = 8;
2 let numeroDeRosas = 70;
3
4 let precioPorLirio = 10;
5 let numeroDeLirios = 50;
6
7 let precioPorTulipan = 2;
8 let numeroDeTulipanes = 120;
9
10 let valoresRosas = precioPorRosa * numeroDeRosas;
11 let valoresLirios = precioPorLirio * numeroDeLirios;
12 let valoresTulipanes = precioPorTulipan * numeroDeTulipanes;
13
14 let total = valoresRosas + valoresLirios + valoresTulipanes;
15 console.log("Rosa: precio unitario:", precioPorRosa, ", cantidad:", numeroDeRosas, ", valor:", valoresRosas);
16 console.log("Lirio: precio unitario:", precioPorLirio, ", cantidad:", numeroDeLirios, ", valor:", valoresLirios);
17 console.log("Tulipán: precio unitario:", precioPorTulipan, ", cantidad:", numeroDeTulipanes, ", valor:", valoresTulipanes);
18 console.log("Total: ", total);
```

Console

```
Rosa: precio unitario: 8, cantidad: 70, valor: 560
Lirio: precio unitario: 10, cantidad: 50, valor: 500
Tulipán: precio unitario: 2, cantidad: 120, valor: 240
Total: 1300
```

Tarea 2

Modifica el código del ejemplo anterior. Supón que los precios de las flores serán constantes (no cambiarán). Declara e inicializa las variables restantes de la misma manera que en el ejemplo anterior. Muestra toda la información recopilada en la consola. Ahora disminuye el número de rosas en 20 y el de lirios en 30. Vuelve a mostrar toda la información recopilada en la consola.

```

app.js x index.html
1 let precioPorRosa = 8;
2 let numeroDeRosas = 70;
3
4 let precioPorLirio = 10;
5 let numeroDeLirios = 50;
6
7 let precioPorTulipan = 2;
8 let numeroDeTulipanes = 120;
9
10 let valoresRosas = precioPorRosa * numeroDeRosas;
11 let valoresLirios = precioPorLirio * numeroDeLirios;
12 let valoresTulipanes = precioPorTulipan * numeroDeTulipanes;
13
14 let total = valoresRosas + valoresLirios + valoresTulipanes;
15 console.log("Rosa: precio unitario:", precioPorRosa, ", cantidad:", numeroDeRosas, ", valor:", valoresRosas);
16 console.log("Lirio: precio unitario:", precioPorLirio, ", cantidad:", numeroDeLirios, ", valor:", valoresLirios);
17 console.log("Tulipán: precio unitario:", precioPorTulipan, ", cantidad:", numeroDeTulipanes, ", valor:", valoresTulipanes);
18 console.log("Total: ", total);
19
20 numeroDeRosas = numeroDeRosas - 20;
21 numeroDeLirios = numeroDeLirios - 30;
22
23 valoresRosas = precioPorRosa * numeroDeRosas;
24 valoresLirios = precioPorLirio * numeroDeLirios;
25 valoresTulipanes = precioPorTulipan * numeroDeTulipanes;
26
27 total = valoresRosas + valoresLirios + valoresTulipanes;
28 console.log("Rosa: precio unitario:", precioPorRosa, ", cantidad:", numeroDeRosas, ", valor:", valoresRosas);
29 console.log("Lirio: precio unitario:", precioPorLirio, ", cantidad:", numeroDeLirios, ", valor:", valoresLirios);
30 console.log("Tulipán: precio unitario:", precioPorTulipan, ", cantidad:", numeroDeTulipanes, ", valor:", valoresTulipanes);
31 console.log("Total: ", total);

```

Console

```

Rosa: precio unitario: 8, cantidad: 70, valor: 560
Lirio: precio unitario: 10, cantidad: 50, valor: 500
Tulipán: precio unitario: 2, cantidad: 120, valor: 240
Total: 1300
Rosa: precio unitario: 8, cantidad: 50, valor: 400
Lirio: precio unitario: 10, cantidad: 20, valor: 200
Tulipán: precio unitario: 2, cantidad: 120, valor: 240
Total: 840

```

TAREAS DATOS PRIMITIVOS PARTE 2

Tarea 1

Escribe un código que cree variables y las inicialice con valores `Boolean`, `Number`, `BigInt`, `String` y tipos `undefined` usando (cuando sea posible) literales y funciones constructoras.

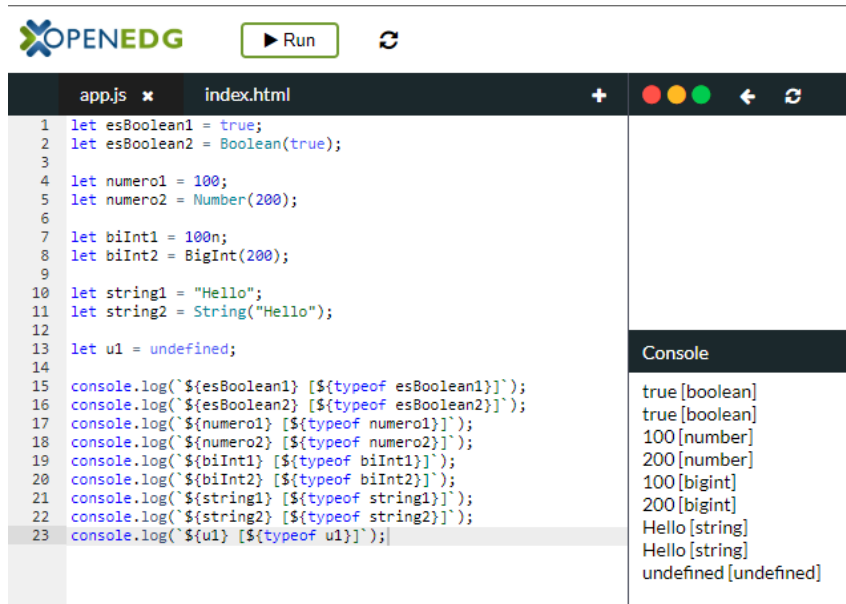
```

1 let esBoolean1 = true;
2 let esBoolean2 = Boolean(true);
3
4 let numero1 = 100;
5 let numero2 = Number(200);
6
7 let biInt1 = 100n;
8 let biInt2 = BigInt(200);
9
10 let string1 = "Hello";
11 let strind2 = String("Hello");
12
13 let u1 = undefined;
14
app.js

```

Tarea 2

Imprime todos los valores y todos los tipos de esos valores usando `console.log`. Intenta usar la interpolación de cadenas para mostrar el valor y el tipo al mismo tiempo con una sola llamada a `console.log`.



```

1 let esBoolean1 = true;
2 let esBoolean2 = Boolean(true);
3
4 let numero1 = 100;
5 let numero2 = Number(200);
6
7 let biInt1 = 100n;
8 let biInt2 = BigInt(200);
9
10 let string1 = "Hello";
11 let string2 = String("Hello");
12
13 let u1 = undefined;
14
15 console.log(`${esBoolean1} [${typeof esBoolean1}]`);
16 console.log(`${esBoolean2} [${typeof esBoolean2}]`);
17 console.log(`${numero1} [${typeof numero1}]`);
18 console.log(`${numero2} [${typeof numero2}]`);
19 console.log(`${biInt1} [${typeof biInt1}]`);
20 console.log(`${biInt2} [${typeof biInt2}]`);
21 console.log(`${string1} [${typeof string1}]`);
22 console.log(`${string2} [${typeof string2}]`);
23 console.log(`${u1} [${typeof u1}]`);

```

Console

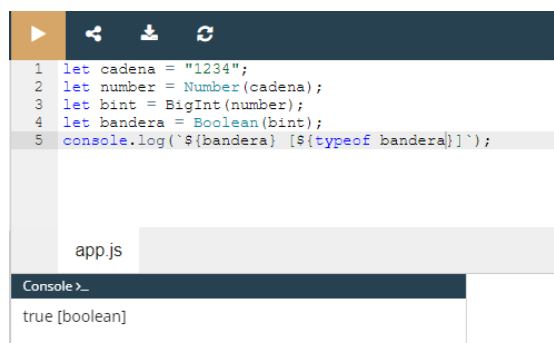
```

true [boolean]
true [boolean]
100 [number]
200 [number]
100 [bigint]
200 [bigint]
Hello [string]
Hello [string]
undefined [undefined]

```

Tarea 3

Realizar una cadena de conversiones: crear un `Boolean` a partir de un `BigInt` creado a partir de un `Number` que se creó a partir de un `String`. Comienza con el valor `"1234"`. ¿Es posible?



```

1 let cadena = "1234";
2 let number = Number(cadena);
3 let bint = BigInt(number);
4 let bandera = Boolean(bint);
5 console.log(`${bandera} [${typeof bandera}]`);

```

app.js

Console >_

```

true [boolean]

```

Tarea 4

Intenta agregar dos valores del mismo tipo y verifica el tipo de resultado. Pruébalo para todos los tipos primitivos.



```

1 let b = true + false;
2 let n = 10 + 25;
3 let bi = 12n + 38n;
4 let s = "Cam" + "ila";
5 let u = undefined + undefined;
6
7 console.log(`${b} [${typeof b}]`); // !!! numero
8 console.log(`${n} [${typeof n}]`);
9 console.log(`${bi} [${typeof bi}]`);
10 console.log(`${s} [${typeof s}]`);
11 console.log(`${u} [${typeof u}]`); // !!! numero

```

app.js

Console >_


```

1 [number]
35 [number]
50 [bigint]
Camila [string]
NaN [number]

```

Tarea 5

Prueba sumar dos valores de diferentes tipos y verifica los resultados.



▶ Run

↺

app.js	index.html	
<pre>1 let no1 = true + 100; 2 let st1 = false + "100"; 3 let no2 = 100 + true; 4 let st2 = 20 + "03"; 5 let st3 = Number("ABC") + Number("F231"); 6 let st4 = st2 + no1; 7 8 console.log(`\${no1} [\${typeof no1}]`); 9 console.log(`\${st1} [\${typeof st1}]`); 10 console.log(`\${no2} [\${typeof no2}]`); 11 console.log(`\${st2} [\${typeof st2}]`); 12 console.log(`\${st3} [\${typeof st3}]`); 13 console.log(`\${st4} [\${typeof st4}]`); 14</pre>		<div>Console</div> <div>101 [number] false100 [string] 101 [number] 2003 [string] NaN [number] 2003101 [string]</div>

Tarea 6

Intenta modificar la línea `const str1 = 42 + "1";` para obtener el resultado `43` (sin eliminar las comillas alrededor del `1`).

app.js

index.html

+

↺

↻

<pre>1 const str1 = 42 + "1"; 2 console.log(`\${str1} [\${typeof str1}]`);</pre>		<div>Console</div> <div>43 [number]</div>
--	--	---

DATOS COMPLEJOS PARTE 3

Tarea 1

Crea un objeto que describa un boleto de tren y guárdalo en la variable `ticket`. El objeto debe tener tres campos:

- estación inicial (el nombre clave es `from`, y como valor, proporciona el nombre de la estación más cercana en tu área)
- estación final (el nombre clave es `to`, y como valor, dar cualquier otra estación dentro de 100 km)
- el precio del boleto (el nombre clave es `price`, y como valor, proporciona la cantidad que te gustaría pagar por este boleto)

El objeto debe crearse usando llaves `{}`, en los que todos los campos se enumerarán inmediatamente. Luego muestra los valores de todos los campos del `ticket` en la consola.

```
1 let ticket = {
2   from: "Quito",
3   to: "Ambato",
4   price: 12
5 };
6 console.log(`Ticket from: ${ticket.from}`);
7 console.log(`Ticket to: ${ticket.to}`);
8 console.log(`Ticket price: ${ticket.price}`);
```

app.js

Console >_

Ticket from: Quito
Ticket to: Ambato
Ticket price: 12

Task 2

Declara un objeto vacío y guárdalo en la variable `person`. Usando la notación de punto, agrega los campos `name` y `surname` al objeto ingresando tus datos como valores. Intenta mostrar los campos individuales en la consola.

```
1 let person = {};
2 person.name = "Camila";
3 person.surname = "Sanchez";
4 console.log(`${person.name} ${person.surname}`);
```

app.js

Console >_

Camila Sanchez

Tarea 3

Estamos creando una pequeña biblioteca de libros sobre programación en JavaScript. Tenemos tres libros y queremos preparar una lista de ellos. Almacenaremos tres datos de cada libro: el título, el autor y el número de páginas:

- *Speaking JavaScript*, Axel Rauschmayer, 460;
- *Programming JavaScript Applications*, Eric Elliott, 254;
- *Understanding ECMAScript 6*, Nicholas C. Zakas, 352.

- Crea un arreglo de tres objetos que representen los libros. Cada objeto debe tener las siguientes propiedades: `title`, `author`, `pages`

```
1 let libros = [{
2     title: "Speaking JavaScript",
3     author: "Axel Rauschmayer",
4     pages: 460
5 },
6 {
7     title: "Programming JavaScript Applications",
8     author: "Eric Elliot",
9     pages: 254
10 },
11 {
12     title: "Understanding ECMAScript 6",
13     author: "Nicholas C. Zakas",
14     pages: 352
15 }
16 ];
```

app.js

Console >_

Tarea 4

Agregar un nuevo libro a la colección: *Learning JavaScript Design Patterns*, por Addy Osmani, 254 páginas. Usa el método apropiado para adjunta el libro al final del arreglo. Muestra la longitud del arreglo y, a su vez, todos los nombres de los libros en la colección.

```
app.js x index.html +
```

```
1 let libros = [{
2     title: "Speaking JavaScript",
3     author: "Axel Rauschmayer",
4     pages: 460
5 },
6 {
7     title: "Programming JavaScript Applications",
8     author: "Eric Elliot",
9     pages: 254
10 },
11 {
12     title: "Understanding ECMAScript 6",
13     author: "Nicholas C. Zakas",
14     pages: 352
15 }
16 ];
17
18 let libro3 = {
19     title: "Learning JavaScript Design Patterns",
20     author: "Addy Osmani",
21     pages: 254
22 };
23
24
25 libros.push(libro3);
26
27 console.log(libros.length);
28 console.log(libros[0].title);
29 console.log(libros[1].title);
30 console.log(libros[2].title);
31 console.log(libros[3].title);
```

Console

```
4
Speaking JavaScript
Programming JavaScript Applications
Understanding ECMAScript 6
Learning JavaScript Design Patterns
```

Task 5

Utiliza el comando slice para copiar los dos últimos libros al nuevo arreglo.

```
app.js x index.html +
1 let libros = [{
2   title: "Speaking JavaScript",
3   author: "Axel Rauschmayer",
4   pages: 460
5 },
6 {
7   title: "Programming JavaScript Applications",
8   author: "Eric Elliot",
9   pages: 254
10 },
11 {
12   title: "Understanding ECMAScript 6",
13   author: "Nicholas C. Zakas",
14   pages: 352
15 }
16 ];
17
18 let libro3 = {
19   title: "Learning JavaScript Design Patterns",
20   author: "Addy Osmani",
21   pages: 254
22 };
23
24
25 libros.push(libro3);
26
27 let selectedBooks = libros.slice(-2);
```

Tarea 6

El primer libro de la colección se pierde en circunstancias inexplicables. Ya has aceptado la pérdida, así que ahora elimínalo del arreglo. ¿Cuál método usarás para este propósito? Muestra la longitud del arreglo y todos los nombres de los libros de la colección a su vez.

```
app.js x index.html +
1 let libros = [{
2   title: "Speaking JavaScript",
3   author: "Axel Rauschmayer",
4   pages: 460
5 },
6 {
7   title: "Programming JavaScript Applications",
8   author: "Eric Elliot",
9   pages: 254
10 },
11 {
12   title: "Understanding ECMAScript 6",
13   author: "Nicholas C. Zakas",
14   pages: 352
15 }
16 ];
17
18 let libro3 = {
19   title: "Learning JavaScript Design Patterns",
20   author: "Addy Osmani",
21   pages: 254
22 };
23
24
25 libros.push(libro3);
26
27 console.log(libros.length);
28 console.log(libros[0].title);
29 console.log(libros[1].title);
30 console.log(libros[2].title);
31 console.log(libros[3].title);
32
33 libros.shift();
34 console.log(libros.length);
35 console.log(libros[0].title);
36 console.log(libros[1].title);
37 console.log(libros[2].title);
38
```

Console

```
4
Speaking JavaScript
Programming JavaScript Applications
Understanding ECMAScript 6
Learning JavaScript Design Patterns
3
Programming JavaScript Applications
Understanding ECMAScript 6
Learning JavaScript Design Patterns
```

Tarea 7

Muestra la suma de las páginas de todos los libros de la colección.



```
1 let libros = [{
2   title: "Speaking JavaScript",
3   author: "Axel Rauschmayer",
4   pages: 460
5 },
6 {
7   title: "Programming JavaScript Applications",
8   author: "Eric Elliot",
9   pages: 254
10 },
11 {
12   title: "Understanding ECMAScript 6",
13   author: "Nicholas C. Zakas",
14   pages: 352
15 }
16 ];
17
18 let libro3 = {
19   title: "Learning JavaScript Design Patterns",
20   author: "Addy Osmani",
21   pages: 254
22 };
23
24 libros.push(libro3);
25
26
27
28 libros.shift();
29
30 let totalPg = libros[0].pages + libros[1].pages + libros[2].pages;
31 console.log("El total de paginas de los libros es: " + totalPg)
```

Console

El total de paginas de los libros es: 860

COMENTARIOS CAP 4

Existe un código que actualmente no funciona. Intenta arreglarlo usando solo comentarios. Intenta, si es posible, usar los atajos de teclado en tu editor para este propósito.

```
"use strict";

const prefix = "username ";

let userName = "Jack";

const userName = "Adam";

let prefixedUserName;

const prefixedUserName;

userName = "John";

prefixedUserName = prefix + userName;

console.log(prefixedUserName + prefixedUserName2);

console.log(prefixedUserName2);
```



```

1  "use strict";
2
3  const prefix = "username_";
4
5  let userName = "Jack";
6  // const userName = "Adam";
7
8  let prefixedUserName;
9  // const prefixedUserName;
10
11  userName = "John";
12  prefixedUserName = prefix + userName;
13
14  console.log(prefixedUserName /*+ prefixedUserName2*/);
15  // console.log(prefixedUserName2);

```

app.js

Console >_

username_John

ARBOL NAVIDEÑO

```

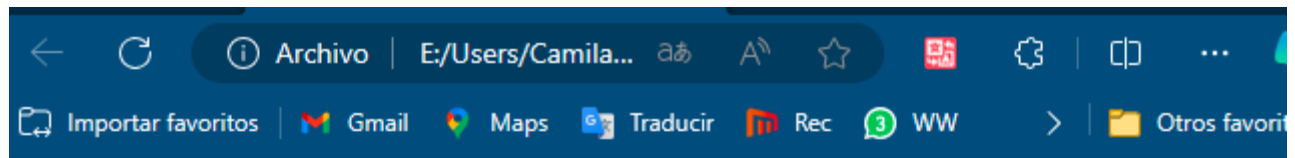
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <title>Christmas Tree</title>
8
9      <style>
10        img {
11          position: absolute;
12        }
13
14        .hide {
15          opacity: 0;
16        }
17      </style>
18    </head>
19    <body>
20      <input
21        onchange="toggleImage('lightsImage')"
22        checked
23        type="checkbox"
24        name="lights"
25      />
26      <label>Lights</label>
27
28      <input
29        onchange="toggleImage('starsImage')"
30        checked
31        type="checkbox"
32        name="stars"

```

```

33     />
34     <label>Stars</label>
35
36     <input
37         onchange="toggleImage('topperImage')"
38         checked
39         type="checkbox"
40         name="topper"
41     />
42     <label>Topper</label>
43
44     <input
45         onchange="toggleImage('snowImage')"
46         checked
47         type="checkbox"
48         name="snow"
49     />
50     <label>Snow</label>
51
52     <input
53         onchange="toggleImage('treeImage')"
54         checked
55         type="checkbox"
56         name="tree"
57     />
58     <label>Tree</label>
59
60     <input
61         onchange="toggleMusic('musicChristmas')"
62         type="checkbox"
63         name="music"
64     />
65
66     <label>Music</label>
67
68     <br />
69     
70     
71     
72     
73     
74     <audio id="musicChristmas" src="music/christmas.mp3" autoplay loop></audio>
75
76     <script>
77         function toggleImage(imageId) {
78             document.getElementById(imageId).classList.toggle("hide");
79         }
80     </script>
81     <script>
82         function toggleMusic(audioId) {
83             var audioElement = document.getElementById(audioId);
84             if (audioElement.paused) {
85                 audioElement.play();
86             } else {
87                 audioElement.pause();
88             }
89         }
90     </script>
91 </body>
92 </html>

```



☒ Lights ☒ Stars ☒ Topper ☒ Snow ☒ Tree ☐ Music

