

EXERCISES QUE TRABAJAREMOS EN EL CUE:

- EXERCISE 1: REFACTORIZANDO CÓDIGO.

EXERCISE 1: REFACTORIZANDO CÓDIGO.

Consideremos que tenemos que realizar una refactorización de código de un programa antiguo, escrito sin las características más nuevas de ES6. En este contexto, nuestra labor será tomar todas las características y funciones aprendidas en el Text Class, y aplicarlas aquí, para así poder transformar el código de este programa en uno más moderno, optimizado y fácil de elegir.

```
JS ejercicio5.js x
JS ejercicio5.js > ...
1  var autoDeportivo = {
2    marca: 'audi',
3    modelo: 'r8',
4    puertas: '3',
5    tipoTransmision: 'mecanico',
6    cantidadCambios: '6',
7    velocidadMaximaKMh: 350,
8    asientos: '2',
9    tipo: 'deportivo'
10 }
11 var autoFamiliar = {
12   marca: 'ford',
13   modelo: 'explorer',
14   puertas: '5',
15   tipoTransmision: 'automatico',
16   cantidadCambios: '5',
17   velocidadMaximaKMh: 200,
18   asientos: '7',
19   tipo: 'familiar'
20 }
21 var autoActual;
22
23 var describeAuto = (auto) => {
24   autoActual = auto.marca + " " + auto.modelo;
25
26   var descripcion = "El " + auto.marca + " " + auto.modelo + " es un auto que consta de "
27   + auto.puertas + " puertas. Es de cambio " + auto.tipoTransmision + " con " + auto.cantidadCambios
28   + " velocidades. Alcanza una velocidad de " + auto.velocidadMaximaKMh + " KM/hr y tiene capacidad para "
29   + auto.asientos + " asientos, por lo que lo hace un perfecto auto de tipo " + auto.tipo + "."
30
31   console.log(autoActual);
32   console.log(descripcion);
33 }
34
35 function retornaTipo(auto){
36   return auto.tipo;
37 }
38
39 describeAuto(autoFamiliar);
40 console.log(retornaTipo(autoFamiliar));
41
42 function retornaCantidadAsientos(auto){
43
44   if(auto == undefined){
45     auto = autoFamiliar;
46   }
47
48   console.log("El auto proporcionado tiene " + auto.asientos + " asientos." )
49 }
50
51
52 retornaCantidadAsientos(autoDeportivo);
53
54 autoDeportivo.velocidadMaximaKMh = 400;
55 autoDeportivo.asientos = 3;
56
57 arrayPropiedadesDeportivo = Object.values(autoDeportivo) // array de propiedades
58
59 for(var i = 0; i < arrayPropiedadesDeportivo.length; i++){
60   console.log(arrayPropiedadesDeportivo[i]);
61 }
62
63
64
65
```

Este código tiene partes que son bastante difíciles de leer, e incluso, puedes escribirlo en tu editor de texto y comprobar lo complicado que es, principalmente en donde se genera el texto para la variable "descripción".

En primer lugar, veamos el uso de variables.

Las variables **"autoDeportivo"** y **"autoFamiliar"** son usadas dentro del programa, pero en ningún momento son redefinidas.

Por lo tanto, el cambio que debemos hacer es sustituir la palabra clave **"var"** por **"const"**, ya que los valores de éstas se mantienen constantes durante todo el programa.

```
JS ejercicio5.js X
JS ejercicio5.js > [0] autoDeportivo
1  const autoDeportivo = {
2      marca: 'audi',
3      modelo: 'r8',
4      puertas: '3',
5      tipoTransmision: 'mecanico',
6      cantidadCambios: '6',
7      velocidadMaximaKMH: 350,
8      asientos: '2',
9      tipo: 'deportivo'
10 }
11 const autoFamiliar = {
12     marca: 'ford',
13     modelo: 'explorer',
14     puertas: '5',
15     tipoTransmision: 'automatico',
16     cantidadCambios: '5',
17     velocidadMaximaKMH: 200,
18     asientos: '7',
19     tipo: 'familiar'
20 }
```

La variable **"autoActual"** es inicializada primero, y luego, su valor es definido dependiendo del argumento que recibe la función. Por lo tanto, debemos cambiar el uso de la palabra clave **"var"** por **"let"**.

```
JS ejercicio5.js x
JS ejercicio5.js > ...
1  const autoDeportivo = {
2    marca: 'audi',
3    modelo: 'r8',
4    puertas: '3',
5    tipoTransmision: 'mecanico',
6    cantidadCambios: '6',
7    velocidadMaximaKMH: 350,
8    asientos: '2',
9    tipo: 'deportivo'
10 }
11 const autoFamiliar = {
12   marca: 'ford',
13   modelo: 'explorer',
14   puertas: '5',
15   tipoTransmision: 'automatico',
16   cantidadCambios: '5',
17   velocidadMaximaKMH: 200,
18   asientos: '7',
19   tipo: 'familiar'
20 }
21 let autoActual;
22
23 function describeAuto(auto){
24   autoActual = auto.marca + " " + auto.modelo;
```

Para el caso de la variable descripción, puede que sea un poco más difícil determinar si debemos utilizar **"const"** o **"let"**. Por un lado, el resultado de esta variable dependerá del argumento que se le pase a la función (de éste extraeremos la información del objeto). El punto clave es que el valor de la variable descripción nunca cambia, sino que lo hacen las variables contenidas en ella (las características del auto, obtenidas desde el objeto). En ninguna parte del programa existe una línea con la expresión "descripción = 'equivalente a otra cosa'"; por lo tanto, si usamos la palabra clave **"const"**, no tendremos un error de ejecución, pues no estamos redefiniendo esta variable.

```
function describeAuto(auto){
  autoActual = auto.marca + " " + auto.modelo;

  const descripcion = "El " + auto.marca + " " + auto.modelo + " es un auto que consta de "
+ auto.puertas + " puertas. Es de cambio " + auto.tipoTransmision + " con " + auto.cantidadCambios
+ " velocidades. Alcanza una velocidad de " + auto.velocidadMaximaKMH + " KM/Hr y tiene capacidad para "
+ auto.asientos + " asientos, por lo que lo hace un perfecto auto de tipo " + auto.tipo + ".";

  console.log(autoActual);
  console.log(descripcion);
}
```

Ahora revisemos el uso de las funciones. **"describeAuto"** puede ser transformada usando la sintaxis de flecha, y guardándola en una variable de tipo **"const"** con el mismo nombre.

```
22
23 const describeAuto = auto => {
24   autoActual = auto.marca + " " + auto.modelo;
25
26   const descripcion = "El " + auto.marca + " " + auto.modelo + " es un auto que consta de "
27 + auto.puertas + " puertas. Es de cambio " + auto.tipoTransmision + " con " + auto.cantidadCambios
28 + " velocidades. Alcanza una velocidad de " + auto.velocidadMaximaKMH + " KM/Hr y tiene capacidad para "
29 + auto.asientos + " asientos, por lo que lo hace un perfecto auto de tipo " + auto.tipo + ".";
30
31   console.log(autoActual);
32   console.log(descripcion);
33 }
34
```

Y **"retornaTipo"** puede ser convertida también a la sintaxis de función de flecha, y hacer uso de su **"return"** implícito de la siguiente forma. Estas funciones son invocadas de la misma manera, y no requieren ningún cambio.

```
const describeAuto = auto => {
  autoActual = auto.marca + " " + auto.modelo;

  const descripcion = "El " + auto.marca + " " + auto.modelo + " es un auto que consta de "
+ auto.puertas + " puertas. Es de cambio " + auto.tipoTransmision + " con " + auto.cantidadCambios
+ " velocidades. Alcanza una velocidad de " + auto.velocidadMaximaKMH + " KM/Hr y tiene capacidad para "
+ auto.asientos + " asientos, por lo que lo hace un perfecto auto de tipo " + auto.tipo + ".";

  console.log(autoActual);
  console.log(descripcion);
}

const retornaTipo = auto => auto.tipo;

describeAuto(autoFamiliar);
console.log(retornaTipo(autoFamiliar));
```

Ahora revisaremos la parte de la concatenación de cadenas de texto con variables, que es la más complicada de leer en este programa.

Para optimizar esta porción de código, utilizaremos **"template literals"** y **"destructuring"**.

Partiremos con **"template literals"**, ya que este cambio representará un gran aporte a la claridad del código.

```
const describeAuto = auto => {  
  autoActual = auto.marca + " " + auto.modelo;  
  
  const descripcion = `El ${auto.marca} ${auto.modelo} es un auto que consta de ${auto.puertas} puertas. Es de cambio ${auto.tipoTransmision}  
  con ${auto.cantidadCambios} velocidades. Alcanza una velocidad de ${auto.velocidadMaximaKMH} KM/Hr y tiene capacidad para ${auto.asientos}  
  asientos, por lo que lo hace un perfecto auto de tipo ${auto.tipo}.`  
  
  console.log(autoActual);  
  console.log(descripcion);  
}  
  
const retornaTipo = auto => auto.tipo;
```

¿Notas que está más claro este código? Es mucho más "humano" de leer, y además, fácil de escribir.

Gracias a la desestructuración de objetos, podemos evitar repetir el nombre del objeto, haciendo incluso aún más claro nuestro código.

```
const describeAuto = auto => {  
  const { marca, modelo, puertas, tipoTransmision, cantidadCambios, velocidadMaximaKMH, asientos, tipo } = auto;  
  
  autoActual = auto.marca + " " + auto.modelo;  
  
  const descripcion = `El ${marca} ${modelo} es un auto que consta de ${puertas} puertas. Es de cambio ${tipoTransmision}  
  con ${cantidadCambios} velocidades. Alcanza una velocidad de ${velocidadMaximaKMH} KM/Hr y tiene capacidad para ${asientos}  
  asientos, por lo que lo hace un perfecto auto de tipo ${tipo}.`  
  
  console.log(autoActual);  
  console.log(descripcion);  
}
```

También podemos reutilizar el nombre de esta variable para definir el valor de **"autoActual"**.

```
const describeAuto = auto => {  
  const { marca, modelo, puertas, tipoTransmision, cantidadCambios, velocidadMaximaKMH, asientos, tipo } = auto;  
  
  autoActual = marca + " " + modelo;  
  
  const descripcion = `El ${marca} ${modelo} es un auto que consta de ${puertas} puertas. Es de cambio ${tipoTransmision}  
  con ${cantidadCambios} velocidades. Alcanza una velocidad de ${velocidadMaximaKMH} KM/Hr y tiene capacidad para ${asientos}  
  asientos, por lo que lo hace un perfecto auto de tipo ${tipo}.`  
  
  console.log(autoActual);  
  console.log(descripcion);  
}
```

Una característica más de la desestructuración de objetos, es que podemos utilizarla en los argumentos de nuestra función, de la siguiente manera:

```
const describeAuto = ({ marca, modelo, puertas, tipoTransmision, cantidadCambios, velocidadMaximaKMh, asientos, tipo }) => {  
  autoActual = marca + " " + modelo;  
  
  const descripcion = `El ${marca} ${modelo} es un auto que consta de ${puertas} puertas. Es de cambio ${tipoTransmision}  
  con ${cantidadCambios} velocidades. Alcanza una velocidad de ${velocidadMaximaKMh} KM/Hr y tiene capacidad para ${asientos}  
  asientos, por lo que lo hace un perfecto auto de tipo ${tipo}.`;  
  
  console.log(autoActual);  
  console.log(descripcion);  
}
```

Si bien no necesariamente cuenta como una optimización, nos permite que el cuerpo de la función tenga menos información, y esto puede ser útil tenerlo en cuenta para el futuro; ya depende de ti si usar esta característica, o no.

En la función **retornaCantidadAsientos()**, podemos observar que se busca otorgar un valor por defecto al argumento auto, en caso de que ésta evalúe el argumento como indefinido.

Haciendo uso de valores por defecto, podemos modificar nuestra función de la siguiente manera, y además, transformarla a una de tipo flecha:

```
const retornaCantidadAsientos = (auto = autoFamiliar) => {  
  console.log("El auto proporcionado tiene " + auto.asientos + " asientos.")  
}
```

En la línea 54 y 55 de nuestro código, se están redefiniendo las propiedades de un objeto.

Si quisiéramos evitar la modificación del objeto original, se puede usar el **spread operator** para crear uno nuevo con sus propiedades modificadas, sin la necesidad de redefinir todas las demás.

```
const autoDeportivoActualizado = { ...autoDeportivo, velocidadMaximaKMh: 400, asientos: 3}
```

Y finalmente, podemos refactorizar el ciclo for para que sea más legible, utilizando un ciclo for of.

```
for(propiedades of arrayPropiedadesDeportivo){
    console.log(propiedades);
}
```

Si comparamos nuestro código ya refactorizado, con el que empezamos a trabajar, notaremos claramente las ventajas de sus características, y la sintaxis ES6 en nuestro programa.

Es tan significativa la diferencia, que incluso en las ofertas de trabajo, se solicita tener conocimiento de la gran mayoría de estas características. Por lo que conocer, e implementar este estilo de código, siempre será beneficioso para tu carrera.

Antes:

```
ejercicio5.js X
ejercicio5.js ...
1 var autoDeportivo = {
2   marca: 'audi',
3   modelo: 'rs',
4   puertas: '3',
5   tipoTransmision: 'mecanico',
6   cantidadCambios: '6',
7   velocidadMaximaKMh: 350,
8   asientos: '2',
9   tipo: 'deportivo'
10 }
11 var autoFamiliar = {
12   marca: 'ford',
13   modelo: 'explorer',
14   puertas: '5',
15   tipoTransmision: 'automatico',
16   cantidadCambios: '5',
17   velocidadMaximaKMh: 200,
18   asientos: '7',
19   tipo: 'familiar'
20 }
21 var autoActual;
22
23 var describeAuto = (auto) => {
24   autoActual = auto.marca + " " + auto.modelo;
25
26   var descripcion = "El " + auto.marca + " " + auto.modelo + " es un auto que consta de "
27   + auto.puertas + " puertas. Es de cambio " + auto.tipoTransmision + " con " + auto.cantidadCambios
28   + " velocidades. Alcanza una velocidad de " + auto.velocidadMaximaKMh + " KM/h y tiene capacidad para "
29   + auto.asientos + " asientos, por lo que lo hace un perfecto auto de tipo " + auto.tipo + "."
30
31   console.log(autoActual);
32   console.log(descripcion);
33 }
34
35 function retornaTipo(auto){
36   return auto.tipo;
37 }
38
39 describeAuto(autoFamiliar);
40 console.log(retornaTipo(autoFamiliar));
41
42 function retornaCantidadAsientos(auto){
43
44   if(auto == undefined){
45     auto = autoFamiliar;
46   }
47
48   console.log("El auto proporcionado tiene " + auto.asientos + " asientos." )
49 }
50
51
52 retornaCantidadAsientos(autoDeportivo);
53
54 autoDeportivo.velocidadMaximaKMh = 400;
55 autoDeportivo.asientos = 3;
56
57 arrayPropiedadesDeportivo = Object.values(autoDeportivo) // array de propiedades
58
59 for(var i = 0; i < arrayPropiedadesDeportivo.length; i++){
60   console.log(arrayPropiedadesDeportivo[i]);
61 }
62
63
64
65
```

Después:

```
JS ejercicio5.js • JS respaldo.js
JS ejercicio5.js > ...
1  const autoDeportivo = {
2    marca: 'audi',
3    modelo: 'r8',
4    puertas: '3',
5    tipoTransmision: 'mecanico',
6    cantidadCambios: '6',
7    velocidadMaximaKMH: 350,
8    asientos: '2',
9    tipo: 'deportivo'
10 }
11 const autoFamiliar = {
12   marca: 'ford',
13   modelo: 'explorer',
14   puertas: '5',
15   tipoTransmision: 'automatico',
16   cantidadCambios: '5',
17   velocidadMaximaKMH: 200,
18   asientos: '7',
19   tipo: 'familiar'
20 }
21 let autoActual;
22
23 const describeAuto = ({ marca,modelo,puertas,tipoTransmision,cantidadCambios,velocidadMaximaKMH,asientos,tipo }) => {
24
25   autoActual = marca + " " + modelo;
26
27   const descripcion = `El ${marca} ${modelo} es un auto que consta de ${puertas} puertas. Es de cambio ${tipoTransmision}
28   con ${cantidadCambios} velocidades. Alcanza una velocidad de ${velocidadMaximaKMH} KM/Hr y tiene capacidad para
29   ${asientos} asientos, por lo que lo hace un perfecto auto de tipo ${tipo}.`
30
31   console.log(autoActual);
32   console.log(descripcion);
33 }
34
35 const retornaTipo = auto => auto.tipo;
36
37 describeAuto(autoFamiliar);
38 console.log(retornaTipo(autoFamiliar));
39
40 const retornaCantidadAsientos = (auto = autoFamiliar) =>{
41   console.log("El auto proporcionado tiene " + auto.asientos + " asientos." )
42 }
43
44 retornaCantidadAsientos();
45
46 const autoDeportivoActualizado = { ...autoDeportivo, velocidadMaximaKMH: 400, asientos: 3}
47
48 arrayPropiedadesDeportivo = Object.values(autoDeportivo) // array de propiedades
49
50
51 for(propiedades of arrayPropiedadesDeportivo){
52   console.log(propiedades);
53 }
54
55
56
57
58
59
60
61
62
63
64
```


El código antes (para copiar):

```
1 var autoDeportivo = {
2   marca: 'audi',
3   modelo: 'r8',
4   puertas: '3',
5   tipoTransmision: 'mecanico',
6   cantidadCambios: '6',
7   velocidadMaximaKMH: 350,
8   asientos: '2',
9   tipo: 'deportivo'
10 }
11 var autoFamiliar = {
12   marca: 'ford',
13   modelo: 'explorer',
14   puertas: '5',
15   tipoTransmision: 'automatico',
16   cantidadCambios: '5',
17   velocidadMaximaKMH: 200,
18   asientos: '7',
19   tipo: 'familiar'
20 }
21 var autoActual;
22
23 var describeAuto = (auto) => {
24   autoActual = auto.marca + " " + auto.modelo;
25
26   var descripcion = "El " + auto.marca + " " + auto.modelo + " es
27 un auto que consta de "
28 + auto.puertas + " puertas. Es de cambio " + auto.tipoTransmision +
29 " con " + auto.cantidadCambios
30 + " velocidades. Alcanza una velocidad de " +
31 auto.velocidadMaximaKMH + " KM/Hr y tiene capacidad para "
32 + auto.asientos + " asientos, por lo que lo hace un perfecto auto
33 de tipo " + auto.tipo + "."
34
35   console.log(autoActual);
36   console.log(descripcion);
37 }
38
39 function retornaTipo(auto) {
40   return auto.tipo;
41 }
42
43 describeAuto(autoFamiliar);
44 console.log(retornaTipo(autoFamiliar));
45
46 function retornaCantidadAsientos(auto) {
47
48   if(auto == undefined){
49     auto = autoFamiliar;
50   }
```

```
51     console.log("El auto proporcionado tiene " + auto.asientos + "  
52 asientos." )  
53 }  
54  
55 retornaCantidadAsientos(autoDeportivo);  
56  
57 autoDeportivo.velocidadMaximaKMH = 400;  
58 autoDeportivo.asientos = 3;  
59  
60 arrayPropiedadesDeportivo = Object.values(autoDeportivo) // array  
61 de propiedades  
62  
63 for(var i = 0; i < arrayPropiedadesDeportivo.length; i++){  
64     console.log(arrayPropiedadesDeportivo[i]);  
65 }
```

El código después (para copiar):

```
1  const autoDeportivo = {  
2      marca: 'audi',  
3      modelo: 'r8',  
4      puertas: '3',  
5      tipoTransmision: 'mecanico',  
6      cantidadCambios: '6',  
7      velocidadMaximaKMH: 350,  
8      asientos: '2',  
9      tipo: 'deportivo'  
10 }  
11 const autoFamiliar = {  
12     marca: 'ford',  
13     modelo: 'explorer',  
14     puertas: '5',  
15     tipoTransmision: 'automatico',  
16     cantidadCambios: '5',  
17     velocidadMaximaKMH: 200,  
18     asientos: '7',  
19     tipo: 'familiar'  
20 }  
21 let autoActual;  
22  
23 const describeAuto = ({  
24     marca, modelo, puertas, tipoTransmision, cantidadCambios, velocidadMaxi  
25     maKMH, asientos, tipo }) => {  
26  
27     autoActual = marca + " " + modelo;  
28  
29     const descripcion = `El ${marca} ${modelo} es un auto que  
30     consta de ${puertas} puertas. Es de cambio ${tipoTransmision}
```



```
31     con ${cantidadCambios} velocidades. Alcanza una velocidad de
32     ${velocidadMaximaKMH} KM/Hr y tiene capacidad para
33     ${asientos} asientos, por lo que lo hace un perfecto auto de
34     tipo ${tipo}.
35
36     console.log(autoActual);
37     console.log(descripcion);
38 }
39
40 const retornaTipo = auto => auto.tipo;
41
42 describeAuto(autoFamiliar);
43 console.log(retornaTipo(autoFamiliar));
44
45 const retornaCantidadAsientos = (auto = autoFamiliar) =>{
46     console.log("El auto proporcionado tiene " + auto.asientos + "
47 asientos." )
48 }
49
50 retornaCantidadAsientos();
51
52 const autoDeportivoActualizado = { ...autoDeportivo,
53 velocidadMaximaKMH: 400, asientos: 3}
54
55 arrayPropiedadesDeportivo = Object.values(autoDeportivo) // array
56 de propiedades
57
58 for(propiedades of arrayPropiedadesDeportivo){
59     console.log(propiedades);
60 }
```