

PRUEBA ANTONIO MORO BULLÓN

TEST 1.

1. El operador assign, asigna las claves y los valores de los parámetros al primero y también devuelve ese valor. Por lo tanto los valores quedarían así:
 - rgb = {red: "#FF0000", green: "#00FF00", blue: "#0000FF", white: "#FFFFFF", black: "#000000"}
 - wb = {white: "#FFFFFF", black: "#000000"}
 - colors = {red: "#FF0000", green: "#00FF00", blue: "#0000FF", white: "#FFFFFF", black: "#000000"}
2. Una posible solución es modificar el operador assign para que los valores no se vuelquen en el primer parámetro.
 - var colors = Object.assign({}, rgb, wb);De esta manera se asignan los valores a colors y a un objeto vacío.
3. Hay muchos problemas con IE, el principal es que ya no hay soporte para él, pero en este caso nos encontramos que específicamente este operador no tiene compatibilidad con este navegador.

Compatibilidad con navegadores

[Update compatibility data on GitHub](#)

assign	
Chrome	45
Edge	12
Firefox	34
IE	No
Opera	32
Safari	9
WebView Android	45
Chrome Android	45
Firefox Android	34
Opera Android	32
Safari iOS	9
Samsung Internet Android	5.0
nodejs	4.0.0

Principalmente se me ocurren dos soluciones, la primera es adaptar el código y no usar assign, y hacer algún reduce. Otra solución puede ser adaptar los polyfills para que este script sea compatible con los navegadores que queramos.

TEST 2.

1. El log que nos muestra sería 5, el valor final de i cuando se ejecuta dentro del setTimeout. El bucle itera y el valor de i va cambiando, cuando pasa el segundo del retardo ese valor ya es 5 y es lo que nos muestra.
2. Para que muestre eso, simplemente borramos el setTimeout y dejamos el console.log():

```
for (var i = 0; i < 5; i++) {  
    console.log(i);  
}
```

TEST 3.

En este script se observan varias funcionalidades de ES6, tales como la declaración de variables con let o const o crear callbacks con expresiones lambda. El motor de IE solo soporta hasta ES5, por lo tanto se debería adaptar nuestro código a sus exigencias. Debemos declarar variables con var y en vez de => usar function. Otra solución, que casi siempre la tenemos, es usar alguna librería que compila nuestro código y lo adapte a versiones anteriores de javascript y así tenga compatibilidad.