

Examen de JavaScript

Índice

Examen de JavaScript.....	1
Contexto común.....	1
A) Variables, tipos, ámbito y conversiones (10 min).....	1
B) Decisiones y bucles (10 min).....	2
C) Arrays y operaciones agregadas (15 min).....	2
D) Funciones y modularidad ligera (10 min).....	2
Rúbrica.....	3

Contexto común

- Trabaja en script.js con "use strict".
- Documenta con comentarios breves cada bloque.
- Valida en consola sin errores.
- No uses librerías externas.

A) Variables, tipos, ámbito y conversiones (10 min)

Enunciado

1. Declara como constante `centro = "CIFP X"`; y como variable modificable `a grupo = "2º DAW"`; y `turno`;
2. Inicializa `turno` con el valor "mañana".
3. Crea la variable modificable `plazas = "25"` y convierte a número de forma segura en `plazasNum`.
4. Escribe una función corta `tipoDe(x)` que devuelva un string con el tipo *real* de `x` (null, array, number, etc.).
5. Demuestra ámbito con `let` y `const` dentro de un bloque y fuera, sin errores. Muestra por consola los resultados.

Evidencias mínimas: `typeof`, comprobación `Array.isArray`, conversión `Number()` y `Number.isFinite`, ejemplo de *shadowing* o inaccesibilidad de bloque.

RA/CE: RA2: b, d, g, h RA3: a, h

B) Decisiones y bucles (10 min)

Enunciado

1. Implementa `clasificarNota(n)` que devuelva "SS" (de 1 a 4 puntos), "AP" (de 5 a 6 puntos), "NT" (de 7 a 9 puntos), "SB" (10 puntos). Usa `switch`.
2. Recorre la constante `notas = [3,5,6.5,8.2,9.7]` y construye un array de objetos `{ nota, tramo: clasificarNota(nota) }`.
3. Con un `for` clásico calcula cuántas están en "SS".
4. Con un `while` calcula cuántas están en "AP".
5. Con un `for..of` o `for..in` calcula cuántas están en "NT". Justifica tu elección.

RA/CE: RA2: e, f, g, h

C) Arrays y operaciones agregadas (15 min)

Enunciado

Dado los datos de productos servidos en por **profesor** en la url: <http://10.103.255.0:3000/coches>

1. Crea `soloModelos` con `map` → `["Corolla","Focus","Ceed","Golf",.....]`.
2. Filtra `year >= 2018` en recientes.
3. Calcula con `reduce`:
 - `sumaPrecios`
 - `maxPrecio` (objeto completo)
 - `mediaPrecio` redondeada a 2 decimales.
4. Ordena por precio ascendente sin mutar el original.

RA/CE: RA2: h RA4: c, d, e, k

D) Funciones y modularidad ligera (10 min)

Enunciado

1. Escribe función `aplicarIVA(productos, tipoIVA=0.15)` que devuelva nuevos objetos `{...p, precioConIVA}` sin mutar.
2. Escribe una función flecha llamada `toEuro` que reciba un número `n` y devuelva una cadena con el importe formateado en euros usando el locale "es-ES" y las opciones `{ style: "currency", currency: "EUR" }`.
3. Escribe el código que, **partiendo del array `coches` y de la función `aplicarIVA` ya implementada**, obtenga un nuevo array llamado `resumen` con el siguiente formato por elemento: `{ etiqueta: "Marca Modelo (Año)", total: precioConIVA }`
4. Documenta cada función con comentario de propósito, parámetros y retorno.

RA/CE: RA2: g, h RA4: a, b, k

Rúbrica

Criterio (RA)	Excelente (10)	Bien (7)	Suficiente (5)	Insuficiente (0)
A1 — RA2-b (0,370 pts) Tipos y operadores en A).	Usa correctamente tipos, typeof, operadores y Array.isArray. Convierte sin errores. (0,370)	1 uso incorrecto o falta un control. (0,259)	Conversión insegura o incoherente. (0,185)	Mezcla tipos sin control. (0,000)
A2 — RA2-d (0,370 pts) Conversiones en A).	Convierte "25" a número de forma segura con Number() y Number.isFinite. Explica peculiaridades. (0,370)	Convierte bien pero sin validar. (0,259)	Convierte con truco no seguro. (0,185)	No convierte o falla. (0,000)
A3 — RA2-g (0,370 pts) Comentarios en A).	Comentarios breves y claros por bloque. (0,370)	Comentarios escuetos o parciales. (0,259)	Comentarios mínimos. (0,185)	Sin comentarios. (0,000)
A4 — RA2-h (0,370 pts) Prueba/entorno en A).	Código sin errores en consola; organización limpia. (0,370)	Avisos menores. (0,259)	Algún error corregible. (0,185)	Errores que impiden ejecutar. (0,000)
A5 — RA3-a (0,370 pts) Objetos nativos en A).	Identifica con acierto tipos/objetos nativos relevantes y su detección real (null, array...). (0,370)	Omite 1 caso. (0,259)	Listado sin verificación real. (0,185)	Confunde los objetos nativos. (0,000)
A6 — RA3-h (0,370 pts) Depuración/documentación A).	Depura y documenta lo realizado. (0,370)	Depura o documenta, falta el otro. (0,259)	Evidencias escasas. (0,185)	Sin evidencias. (0,000)
B1 — RA2-e (0,556 pts) Decisiones en B).	switch correcto con todos los tramos. (0,556)	Falta un tramo o límites imprecisos. (0,389)	Lógica parcial. (0,278)	Incorrecto o ausente. (0,000)
B2 — RA2-f (0,556 pts) Bucles en B).	Usa for, while, y for..of/in con justificación. (0,556)	Un bucle con fallo menor. (0,389)	Usa solo dos bucles o sin justificación. (0,278)	Bucles erróneos. (0,000)
B3 — RA2-g (0,556 pts) Comentarios en B).	Comentarios claros por bloque. (0,556)	Parciales. (0,389)	Mínimos. (0,278)	Ninguno. (0,000)
B4 — RA2-h (0,556 pts) Prueba/entorno en B).	Resultados correctos y trazas útiles. (0,556)	Alguna traza confusa. (0,389)	Pocas evidencias. (0,278)	Sin verificación. (0,000)
C1 — RA2-h (0,666 pts) Prueba/entorno en C).	Verifica fetch/datos y resultados de agregados. (0,666)	Verificación parcial. (0,466)	Verificación mínima. (0,333)	Sin verificación. (0,000)
C2 — RA4-c (0,666 pts) Características de arrays.	No muta original; usa copias en ordenación. (0,666)	Un descuido sin afectar salida. (0,466)	Duda sobre mutación. (0,333)	Mutación del origen. (0,000)
C3 — RA4-d (0,666 pts) Creación/uso de arrays.	map, filter, reduce correctos. (0,666)	Falta un agregado o detalle. (0,466)	Implementación básica con errores leves. (0,333)	Agregados mal empleados. (0,000)
C4 — RA4-e (0,666 pts) Operaciones agregadas.	Suma, máximo objeto y media redondeada correctos. (0,666)	1 cálculo impreciso. (0,466)	1 cálculo ausente. (0,333)	No calcula. (0,000)
C5 — RA4-k (0,666 pts) Depuración/documentación C).	Código comentado y probado. (0,666)	Uno de los dos. (0,466)	Escaso. (0,333)	Nulo. (0,000)
D1 — RA2-g (0,444 pts) Comentarios en	Propósito, params y retorno en cada función. (0,444)	Falta en una función. (0,311)	Comentarios genéricos. (0,222)	Sin comentarios. (0,000)

Criterio (RA)	Excelente (10)	Bien (7)	Suficiente (5)	Insuficiente (0)
D).				
D2 — RA2-h (0,444 pts) Prueba/entorno en D).	Evidencias de prueba sin errores. (0,444)	Avisos menores. (0,311)	Pocas evidencias. (0,222)	Sin pruebas. (0,000)
D3 — RA4-a (0,444 pts) Funciones predefinidas.	Usa toLocaleString correctamente. (0,444)	Uso con opciones incompletas. (0,311)	Uso básico. (0,222)	No usa o mal. (0,000)
D4 — RA4-b (0,444 pts) Funciones de usuario.	aplicarIVA inmutable y toEuro flecha correctas. (0,444)	Un detalle de estilo/retorno mejorable. (0,311)	Cambia datos pero compensa. (0,222)	Mutación o fallo. (0,000)
D5 — RA4-k (0,450 pts) Depuración/documentación D).	Depura y documenta. (0,450)	Uno de los dos. (0,315)	Mínimo. (0,225)	Nulo. (0,000)