



EJERCICIOS DE JAVASCRIPT

DESESTRUCTURACIÓN

Tomás cabello Fernández

Desarrollo Web en Entorno Servidor

Desarrollo de Aplicaciones Web



6. ¿Qué valores tienen a, b y c tras la ejecución del siguiente código?

```
let a,b,c;  
[a, b, c] = ['Juan', 'Pedro', 'Maria', 'Luis'];
```

```
> let a,b,c;  
undefined  
> [a, b, c] = ['Juan', 'Pedro', 'Maria', 'Luis'];  
[ 'Juan', 'Pedro', 'Maria', 'Luis' ]  
> console.log(a,b,c)  
Juan Pedro Maria
```

7. ¿Qué valores tienen a y b tras la ejecución del siguiente código?

```
let a,b;  
[a, , b] = ['Juan', 'Pedro', 'Maria', 'Luis'];
```

```
> let a,b;  
undefined  
> [a, , b] = ['Juan', 'Pedro', 'María', 'Luis'];  
[ 'Juan', 'Pedro', 'María', 'Luis' ]  
> console.log(a,b)  
Juan María  
undefined
```

8. ¿Qué valores tienen a y b tras la ejecución del siguiente código?

```
let a,b;  
[a, , , b] = ['Juan', 'Pedro', 'Maria', 'Luis'];
```

```
> let a,b;  
undefined  
> [a, , , b] = ['Juan', 'Pedro', 'Maria', 'Luis'];  
[ 'Juan', 'Pedro', 'María', 'Luis' ]  
> console.log(a,b);  
Juan Luis  
undefined
```



9. ¿Qué valores tienen a y b tras la ejecución del siguiente código?

```
let a,b;  
const amigos = ['Juan', 'Pedro', 'Maria', 'Luis']  
[a, , b] = amigos;
```

```
> let a,b  
undefined  
> const amigos = ['Juan','Pedro','Maria','Luis']  
undefined  
> [a,,,b] = amigos  
[ 'Juan', 'Pedro', 'Maria', 'Luis' ]  
> console.log(a,b)  
Juan Luis
```

10. Usa desestructuración para obtener los primeros tres nombres.

```
const nombres = ['Juan', 'Pedro', 'Maria', 'Luis'];
```

```
> const nombre = ['Juan', 'Pedro', 'Maria', 'Luis']  
undefined  
> [a,b,c] = nombre  
[ 'Juan', 'Pedro', 'Maria', 'Luis' ]  
> console.log(a,b,c)  
Juan Pedro Maria
```

11. Dado este array de frutas, realiza desestructuración para obtener las primeras tres.

```
const frutas = ['manzana', 'banana', 'cereza', 'durazno'];
```

```
> const frutas = ['manzana', 'banana', 'cereza', 'durazno'];  
undefined  
> [man,ban,cer] = frutas  
[ 'manzana', 'banana', 'cereza', 'durazno' ]  
> console.log(man,ban,cer)  
manzana banana cereza  
undefined
```

12. Desestructura este array de números y asigna el primer y último elemento a variables.

```
const numeros = [1, 2, 3, 4];
```

```
> const numeros = [1, 2, 3, 4];  
undefined  
> const [num1,,,num4] = numeros  
undefined  
> num1  
1  
> num4  
4
```



13. Usa desestructuración para intercambiar estas dos variables.

```
let a = 1, b = 2;
```

```
> let a = 1, b = 2;
undefined
>
> [a, b] = [b, a];
[ 2, 1 ]
> a
2
> b
1
```

14. ¿Qué valores tienen nombre y edad tras la ejecución del siguiente código?

```
const {nombre, edad} = {nombre: 'Carlos', edad: 20};
```

```
> const {nombre, edad} = {nombre: 'Carlos', edad: 20};
undefined
> nombre
'Carlos'
> edad
20
```

15. ¿Qué valores tienen name y age tras la ejecución del siguiente código?

```
const {nombre: name, edad: age} = {nombre: 'Ana', edad: 20};
> const {nombre: name, edad: age} = {nombre: 'Ana', edad: 20};
undefined
> name
'Ana'
> age
20
```

16. ¿Qué valores tienen nombre y edad tras la ejecución del siguiente código?

```
const estudiante = {nombre: 'Eva', edad: 21};
const {nombre, edad} = estudiante;
```

```
> const estudiante = {nombre: 'Eva', edad: 21};
undefined
> const {nombre, edad} = estudiante;
undefined
> nombre
'Eva'
> edad
21
```



17. Desestructura este objeto en variables nombre y edad.

```
const estudiante = {nombre: 'Carlos', edad: 20};
```

```
> const estudiante = {nombre: 'Carlos', edad: 20}
undefined
> estudiante
{ nombre: 'Carlos', edad: 20 }
> const { nombre, edad } = estudiante
undefined
> nombre
'Carlos'
> edad
20
```

18. Desestructura este objeto en variables llamadas alias y annos.

```
const estudiante = {nombre: 'Carlos', edad: 20};
```

```
> const estudiante = {nombre: 'Carlos', edad: 20};
undefined
> const { edad: anno, nombre: alias} = estudiante
undefined
> anno
20
> alias
'Carlos'
```

19. Desestructura este array y extrae el nombre y edad de Juan y la edad de Ana.

```
const personas = [
  {nombre: 'Juan', edad: 25},
  {nombre: 'Ana', edad: 30}
];
```



```
> const personas = [  
... {nombre: 'Juan', edad: 25},  
... {nombre: 'Ana', edad: 30}  
... ];  
undefined  
> const [{nombre,edad},{ edad : edadAna}] = personas  
undefined  
> nombre  
'Juan'  
> edad  
25  
> edadAna  
30
```

20. ¿Qué valores tienen a, b, c y d tras la ejecución del siguiente código?

```
let a,b,c,d;  
[[a, b], [c,d]] = [[1, 2], [3, 4]];
```

```
> let a,b,c,d;  
undefined  
> [[a, b], [c,d]] = [[1, 2], [3, 4]];  
[ [ 1, 2 ], [ 3, 4 ] ]  
> a  
1  
> b  
2  
> c  
3  
> d  
4
```

21. ¿Qué valores tienen a y b tras la ejecución del siguiente código?

```
let a,b;  
[[a, ], [ ,b]] = [[1, 2], [3, 4]];
```



```
> let a,b;  
undefined  
> [[a, ], [ ,b]] = [[1, 2], [3, 4]];  
[ [ 1, 2 ], [ 3, 4 ] ]  
> a  
1  
> b  
4
```

22. Utiliza desestructuración para extraer todos los elementos de este array anidado.

```
const anidado = [[1, 2], [3, 4]];
```

```
> const anidado = [[1, 2], [3, 4]];  
undefined  
> const [[a,b],[c,d]] = anidado  
undefined  
> a  
1  
> b  
2  
> c  
3  
> d  
4
```

23. ¿Qué valor tienen las variables titulo y autor tras ejecutar este código?

```
const {autor, titulo} = {titulo: '1984', autor: 'Orwell'};
```

```
> const {autor, titulo} = {titulo: '1984', autor: 'Orwell'};  
undefined  
> titulo  
'1984'  
> autor  
'Orwell'
```

24. ¿Qué valor tienen las variables title y author tras ejecutar este código?

```
const {autor: author, titulo: title} = {titulo: '1984', autor: 'Orwell'};
```



```
> const {autor: author, titulo: title} = {titulo: '1984', autor: 'Orwell'};
undefined
> author
'Orwell'
> title
'1984'
```

25. Desestructura este objeto con información de la revista (nombre, editorial).

```
const revista = {nombre: 'PC computer', editorial: 'Prisma'};
> const revista = {nombre: 'PC computer', editorial: 'Prisma'};
undefined
> const {editorial, nombre} = revista
undefined
> nombre
'PC computer'
> editorial
'Prisma'
```

26. Desestructura este objeto con información de la revista (titulo, editor).

```
const revista = {nombre: 'PC computer', editorial: 'Prisma'};
> const revista = {nombre: 'PC computer', editorial: 'Prisma'};
undefined
> const {nombre: titulo, editorial: editor} = revista
undefined
> titulo
'PC computer'
> editor
'Prisma'
```

27. Dado este objeto de usuario, desestructúralo para obtener su nombre y edad.

```
const usuario = {nombre: 'Ana', edad: 28};
> const usuario = {nombre: 'Ana', edad: 28};
undefined
> const {nombre, edad} = usuario
undefined
> nombre
'Ana'
> edad
28
```

28. Desestructura este objeto de contacto para extraer solo el email y el teléfono.

```
const contacto = {
  nombre: 'Juan',
```




```
    email: 'juan@ejemplo.com',  
    telefono: '123456789'  
};  
  
> const {email,telefono} = contacto  
undefined  
> email  
'juan@ejemplo.com'  
> telefono  
'123456789'
```

29. Usa desestructuración para asignar el valor de x a una variable con nombre a y el de y a una variable con nombre b.

```
const objeto = {x: 10, y: 20};
```

```
> const objeto = {x: 10, y: 20};  
undefined  
> const {x:a,y:b} = objeto  
undefined  
> a  
10  
> b  
20
```

30. ¿Qué valor tiene la variable domicilio tras ejecutar este código?

```
const contacto = {  
  nombre: 'Eva',  
  domicilio: {  
    direccion: 'C/ Nueva, 1',  
    localidad: 'Montilla'  
  }  
};  
const {domicilio} = contacto;
```



```
> const contacto = {
... nombre: 'Eva',
... domicilio: {
... direccion: 'C/ Nueva, 1',
... localidad: 'Montilla'
... }
... };
undefined
> const {domicilio} = contacto;
undefined
> domicilio
{ direccion: 'C/ Nueva, 1', localidad: 'Montilla' }
```

31. ¿Qué valor tiene la variable localidad tras ejecutar este código?

```
const contacto = {
  nombre: 'Eva',
  domicilio: {
    direccion: 'C/ Nueva, 1',
    localidad: 'Montilla'
  }
};
const {domicilio: {localidad}} = contacto;
```

```
> const contacto = {
... nombre: 'Eva',
... domicilio: {
... direccion: 'C/ Nueva, 1',
...     localidad: 'Montilla'
... }
... };
undefined
> const {domicilio: {localidad}} = contacto;
undefined
> localidad
'Montilla'
```

32. Desestructura este objeto anidado para obtener la propiedad b.

```
const anidado = {a: {b: 1}};
const {a: {b}} = anidado;
```



```
> const anidado = {a: {b: 1}};
undefined
> const {a: {b}} = anidado;
undefined
> b
1
```

33. ¿Qué valor tiene la variable edad tras ejecutar este código?

```
const persona = {nombre: 'Eva', edad: 10};
const {nombre, edad = 12} = persona;
```

```
> const persona = {nombre: 'Eva', edad: 10};
undefined
> const {nombre, edad = 12} = persona;
undefined
> edad
10
```

34. Usa desestructuración para extraer los valores predeterminados, pero asignando a timeout el valor 5000.

```
const config = {url: 'http://ejemplo.com', timeout: 1000};
> const config = {url: 'http://ejemplo.com', timeout: 1000};
undefined
> const { timeout = 5000 } = config
undefined
> timeout
1000
```

35. Desestructura este objeto que contiene detalles de un producto (nombre, precio) y calcula el IVA.

```
const producto = {nombre: 'Laptop', precio: 1000};
> const producto = {nombre: 'Laptop', precio: 1000};
undefined
> const {nombre, precio, iva = precio*0.22} = producto
undefined
> iva
220
```

36. Usa desestructuración para extraer propiedades email1 y email2.

```
const contactos = [
  {email: 'ejemplo@dominio.com'},
  {email: 'otro@dominio.com'}
];
```



```
> const contactos = [
... {email: 'ejemplo@dominio.com'},
... {email: 'otro@dominio.com'}
... ];
undefined
> const [{email:email1},{email:email2}] = contactos
undefined
> email1
'ejemplo@dominio.com'
> email2
'otro@dominio.com'
```

37. Usa desestructuración para extraer el email del tercer contacto.

```
const contactos = [
  {nombre: 'Isabel', email: 'isabel@dominio.com'},
  {nombre: 'Eva', email: 'eva@dominio.com'},
  {nombre: 'Rosa', email: 'rosa@dominio.com'}
];
```

```
> const contactos = [
... {nombre: 'Isabel', email: 'isabel@dominio.com'},
... {nombre: 'Eva', email: 'eva@dominio.com'},
... {nombre: 'Rosa', email: 'rosa@dominio.com'}
... ];
undefined
> const [{},{},{email:email3}] = contactos
undefined
> email3
'rosa@dominio.com'
```

38. Usa desestructuración para extraer el email del tercer contacto.

```
const contactos = [
  {nombre: 'Isabel', email: 'isabel@dominio.com'},
  {nombre: 'Eva', email: 'eva@dominio.com'},
  {nombre: 'Rosa', email: 'rosa@dominio.com'}
];
```



```
> const contactos = [
... {nombre: 'Isabel', email: 'isabel@dominio.com'},
... {nombre: 'Eva', email: 'eva@dominio.com'},
... {nombre: 'Rosa', email: 'rosa@dominio.com'}
... ];
undefined
> const [{},{},{email:email3}] = contactos
undefined
> email3
'rosa@dominio.com'
```

39. Completa el código que falta.

```
const estudiante = {nombre: 'Eva', edad: 21};
function obtenerInfo(persona) {
    const {nombre, edad} = persona;
    return `${nombre} tiene ${edad} años.`;
}
console.log(obtenerInfo(estudiante)); // "Eva tiene 21 años."
> const estudiante = {nombre: 'Eva', edad: 21};
undefined
> function obtenerInfo(persona) {
... const {nombre, edad} = persona;
... return `${nombre} tiene ${edad} años.`;
... }
undefined
> console.log(obtenerInfo(estudiante));
Eva tiene 21 años.
undefined
```

40. Completa el código que falta.

```
const estudiante = {nombre: 'Eva', edad: 21};
function obtenerInfo({nombre,edad}) {
    return `${nombre} tiene ${edad} años.`;
}
console.log(obtenerInfo(estudiante)); // "Eva tiene 21 años."
```



```
> const estudiante = {nombre: 'Eva', edad: 21};
undefined
> function obtenerInfo({nombre,edad}) {
... return `${nombre} tiene ${edad} años.`;
... }
undefined
> console.log(obtenerInfo(estudiante));
Eva tiene 21 años.
undefined
```

41. Completa el código que falta.

```
const nums = [1, 2]
function suma( numeros ) {
    return numeros[0] + numeros[1];
}
console.log(suma (nums)) // 3
```

```
> const nums = [1, 2]
undefined
> function suma( numeros ) {
... return numeros[0] + numeros[1];
... }
undefined
> console.log(suma (nums))
3
undefined
```

42. Completa el código que falta.

```
const nums = [1, 2];
function suma([a,b]) {
    return a + b;
}
console.log(suma (nums)) // 3
```

```
> const nums = [1, 2];
undefined
> function suma([a,b]) {
... return a + b;
... }
undefined
> console.log(suma (nums))
3
undefined
```



43. Completa el código que falta.

```
const amigo = { nombre: 'Juan', edad: 33 };  
function mostrarNombre( persona ) {  
    console.log(persona.nombre);  
}  
mostrarNombre(amigo); // Juan
```

```
> const amigo = { nombre: 'Juan', edad: 33 };  
undefined  
> function mostrarNombre( persona ) {  
... console.log(persona.nombre);  
... }  
undefined  
> mostrarNombre(amigo);  
Juan  
undefined
```

44. Completa el código que falta.

```
const amigo = { nombre: 'Juan', edad: 33 };  
function mostrarNombre({nombre,edad}) {  
    console.log(nombre);  
}  
  
mostrarNombre(amigo); // Juan
```

```
> const amigo = { nombre: 'Juan', edad: 33 };  
undefined  
> function mostrarNombre({nombre,edad}) {  
... console.log(nombre);  
... }  
undefined  
>  
> mostrarNombre(amigo);  
Juan  
undefined
```

TOMÁS CABELLO FERNÁNDEZ