

¿Enviar la respuesta?

Solo puedes responder una vez este formulario. ¿Quieres continuar?

[CAMBIAR DE CUENTA](#)

ENVIAR

## Prueba de Admisión

**\*Obligatorio**

Prueba Técnica

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
let athleteFinalPosition = 'first placea';  
  
switch(athleteFinalPosition){  
  case 'first place': console.log('You get the  
gold medal!'); break;  
  case 'second place': console.log('You get the  
silver medal!');break;  
  case 'third place': console.log('You get the  
bronze medal!'); break;  
  default: console.log('No medal awarded.');
```

- ☐ You get the bronze medal!
- ☐ error
- ☐ You get the gold medal!
- ☒ No medal awarded.

¿Cómo verificamos que cierta function en Javascript sea compatible con los navegadores? \*

- ☐ Ejecutando la función en los navegadores más usados
- ☒ Buscando la function en <https://caniuse.com/>
- ☐ Todas las funciones son compatibles con los navegadores

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
▼ const playerFactory = (life, power, type)=>{
▼   return {
      life,
      power,
      type,
▼   attack(player){
▼     if(this.type === 'warrior'){
        player.life -= this.power;
      }
    }
  }
}

const hero = playerFactory(100,10, 'hero');
const enemy = playerFactory(30, 5, 'warrior');

enemy.attack(hero)

console.log(hero.life)
```

- ☐ 90
- ☒ 100
- ☐ 95
- ☐ 40

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
let retreatMessage = 'Retreat';  
  
▼ const alienShip = {  
  ▼ retreat:()=>{  
    console.log(retreatMessage)  
  },  
  ▼ takeOff: ()=>{  
    console.log('Spim... Borp... Glix...  
    Blastoff!')  
  }  
}  
  
alienShip.retreat();  
alienShip.takeOff();
```

- ☐ Retreat
- ☐ Spim... Borp... Glix... Blastoff!
- ☒ Retreat Spim... Borp... Glix... Blastoff!

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
const cupsOfSugarNeeded =2;  
  
let cupsAdded= 0;  
  
▼ do {  
    cupsAdded++;  
} while(cupsAdded < cupsOfSugarNeeded)  
  
console.log(cupsAdded)
```

- ☒ 1
- ☐ 0
- ☐ 2

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
const satellite = 'The Moon'
const galaxy = 'The Milky Way'
const stars = 'North Star'

▼ const callMyNightSky = ()=>{
  return 'Night Sky: ' + satellite + ', ' +
  stars + ', and ' + galaxy;
}

console.log(callMyNightSky())
```

- ☐ Night Sky: , and
- ☒ Night Sky: The Moon, North Star, and The Milky Way
- ☐ Night Sky: satellite, starts, and galaxy

¿Qué herramienta podemos usar para tener compatibilidad de Javascript con todos los navegadores? \*

- ☐ Babel
- ☐ Rollup
- ☒ Webpack
- ☐ No se necesita, Javascript es compatible con todos los navegadores

¿Qué tipo de variables se pueden declarar en Javascript? \*

- ☐ val
- ☒ let
- ☒ var
- ☒ const

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
const rapperArray = ["Lil' Kim", "Jay-Z",  
"Notorious B.I.G.", "Tupac"];  
  
// Write your code below  
▼ for(let i=0;i<rapperArray.length;i++){  
    console.log(rapperArray[i])  
    if(rapperArray[i] === 'Notorious B.I.G.')  
    break;  
}
```

- ☐ Tupac
- ☒ Lil' Kim, Jay-Z, Notorious B.I.G.
- ☐ Notorious B.I.G.
- ☐ Lil' Kim, Jay-Z, Tupac

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
const entree = 'Enchiladas'  
console.log(entree)  
entree = 'Tacos'
```

- ☒ Enchiladas
- ☐ Error, una constante no puede ser re asignada
- ☐ undefined
- ☐ Tacos



¿Cuál es el resultado de ejecutar el siguiente código? \*

```
▼ const robot = {  
  _model: '1E78V2',  
  _energyLevel: 100,  
  ▼ get energyLevel(){  
    ▼ if(typeof this._energyLevel === 'number'){  
      return `My current energy level is ${this._energyLevel}`  
    }else{  
      return 'System malfunction: cannot retrieve energy level'  
    }  
  }  
};  
  
console.log(robot.energyLevel)
```

- ☒ 'My current energy level is 100'
- ☐ ''
- ☐ 'My current energy level is 1E78V2'
- ☐ 'System malfunction: cannot retrieve energy level'

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
const newNumbers = [1, 3, 5, 7];  
  
const newSum = newNumbers.reduce((accumulator,  
▼ currentValue)=>{  
    return accumulator + currentValue  
}, 10)  
  
console.log(newSum)
```

- ☐ 26
- ☒ 16
- ☐ 21

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
▼ function sayThanks(name) {  
    console.log('Thank you for your purchase '  
+name+'! We appreciate your business.');
```

```
}  
  
sayThanks('Cole')
```

- ☐ Thank you for your purchase name! We appreciate your business.
- ☐ Cole
- ☒ Thank you for your purchase Cole! We appreciate your business.

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
▼ const filter = (array, check) => {  
  let result = [];  
▼ for(let i=0;i<array.length;i++){  
▼   if(check(array[i])){  
     result.push(array[i]);  
   }  
}  
  
  console.log(result);  
}  
  
const numbers = [1,2,3,4,5,6,7];  
  
filter(numbers, p => p % 2 === 0);  
filter(numbers, q => q % 2 === 1);
```

- ☐ [ 1, 3, 5, 7 ] [ 2, 4, 6 ]
- ☐ [ 0 ] [ 1 ]
- ☒ [ 2, 4, 6 ] [ 1, 3, 5, 7 ]
- ☐ [ 1, 2, 3, 4, 5, 6, 7 ]

¿Cómo se llama el tipo de función que se ve a continuación? \*

```
const plantNeedsWater = day => day ===  
'Wednesday' ? true : false;
```

- ☒ Short Function
- ☐ Regular Function
- ☐ Arrow Function
- ☐ Concise Function

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
▼ class Surgeon {  
  ▼ constructor(name, department) {  
    this._name = name;  
    this._department = department;  
    this._remainingVacationDays = 20  
  }  
  
  ▼ get name(){  
    return this._name  
  }  
  
  ▼ get department(){  
    return this._department  
  }  
  
  ▼ get remainingVacationDays(){  
    return this._remainingVacationDays  
  }  
  
  ▼ takeVacationDays(daysOff){  
    this._remainingVacationDays -= daysOff;  
  }  
}  
  
const surgeonCurry = new Surgeon('Curry', 'Cardiovascular');  
const surgeonDurant = new Surgeon('Durant', 'Orthopedics');  
  
console.log(surgeonCurry.name)  
surgeonCurry.takeVacationDays(3)  
console.log(surgeonCurry.remainingVacationDays)
```

- ☐ Cardiovascular 20
- ☐ Durant 17
- ☒ Curry 17
- ☐ Durant 23

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
const animals = ['Hen', 'elephant', 'llama',  
'leopard', 'ostrich', 'Whale', 'octopus',  
'rabbit', 'lion', 'dog'];  
  
// Create the secretMessage array below  
const secretMessage = animals.map(animal =>  
  animal[0]);  
  
console.log(secretMessage.join(''));
```

- ☐ dlroWolleH
- ☐ HenHenHenHenHenHenHenHenHenHen
- ☒ HelloWorld
- ☐ Hen

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
const bobsFollowers = ['a','b','c','d']

const tinasFollowers = ['a','b','e']

const mutualFollowers = []

▼ for(let b=0;b< bobsFollowers.length;b++){
▼   for(let t=0;t< tinasFollowers.length;t++){
▼     if(bobsFollowers[b] === tinasFollowers[t]){
       mutualFollowers.push(bobsFollowers[b]);
     }
   }
}

console.log(mutualFollowers)
```

- ☐ []
- ☐ ['a', 'b', 'c', 'd', 'e']
- ☐ ['c', 'd', 'e']
- ☒ ['a', 'b']

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
let isLocked = false;

isLocked?
  console.log('You will need a key to open the
door.')
:
  console.log('You will not need a key to open
the door.');
```

- ☐ error
- ☒ You will not need a key to open the door.
- ☐ You will need a key to open the door.



¿Cuál es el resultado de ejecutar el siguiente código? \*

```
const urlParams = ['type', 'color', 'size',  
  'shape']  
  
▼ const request = urlParams  
  .filter(text => text.length < 5)  
  .map(name => `param=${name}`)  
  .join('&')  
  
console.log(request)
```

- ☐ param=type&param=size
- ☒ param=type&param=color&param=size&param=shape
- ☐ param=color&param=shape

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
let score = 3;  
  
const hitTarget= (damage,score) => score +  
damage;  
  
hitTarget(10);  
hitTarget(5);  
hitTarget(9);  
  
console.log(score);
```

- ☒ 3
- ☐ 9
- ☐ 27
- ☐ 12
- ☐ 24

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
for(let i =5;i<=10;i++){  
  console.log(i);  
}
```

- ☐ 0 1 2 3 4 5 6 7 8 9 10
- ☐ 10
- ☒ 5 6 7 8 9 10
- ☐ 1 2 3 4 5

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
▼ let spaceship = {  
  'Fuel Type' : 'Turbo Fuel',  
  'Active Mission' : true,  
  homePlanet : 'Earth',  
  numCrew: 5  
};  
  
let propName = 'Active Mission';  
  
const isActive = spaceship['Active Mission']  
  
console.log(spaceship[propName])
```

- ☒ 'Active Mission'
- ☐ Earth
- ☐ true

¿Cuál es el resultado de ejecutar el siguiente código? \*

```
const concept = ['arrays', 'can', 'be', 'mutated'];  
  
▼ function changeArr(arr){  
  arr[3] = 'MUTATED';  
}  
  
changeArr(concept);  
  
console.log(concept)
```

- ☐ [ 'arrays', 'can', 'be', 'mutated' ]
- ☒ [ 'arrays', 'can', 'be', 'MUTATED' ]
- ☐ 'MUTATED'

¿Qué contiene una variable no inicializada? \*

- ☐ 0
- ☐ null
- ☒ undefined
- ☐ "

Google Formularios