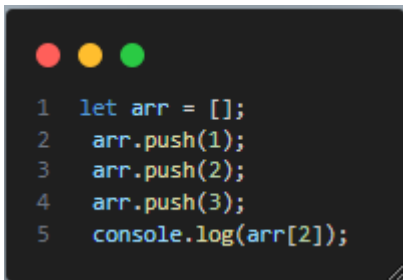


Curso: KrakeDev M1 Programando con Javascript
Formulario: Prueba Teorica
Username: Luis Miguel Llumiquinga Llumiquinga
Correo electrónico: luis_mk4@hotmail.com

puntuación: 91% ¡Aprobado!
Puntos: 41 / 45
Respuesta correcta.: 41 / 45
Tiempo empleado: 00:18:27
Fecha de envío: 26 October, 2025

1

A partir de ese código



```
1 let arr = [];  
2 arr.push(1);  
3 arr.push(2);  
4 arr.push(3);  
5 console.log(arr[2]);
```

¿Qué número se imprimiría?

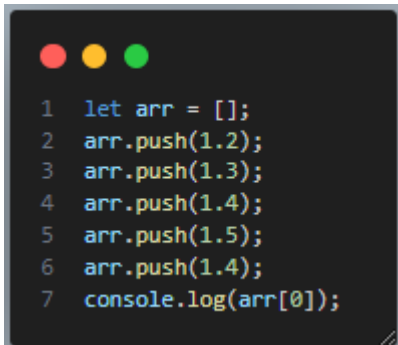
1 / 1

- ☐ 2
- ☒ 3
- ☐ 1
- ☐ -

Respuesta correcta

3

2 A partir de este código.

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It contains the following JavaScript code:

```
1 let arr = [];  
2 arr.push(1.2);  
3 arr.push(1.3);  
4 arr.push(1.4);  
5 arr.push(1.5);  
6 arr.push(1.4);  
7 console.log(arr[0]);
```

¿Qué se imprimiría?

1 / 1

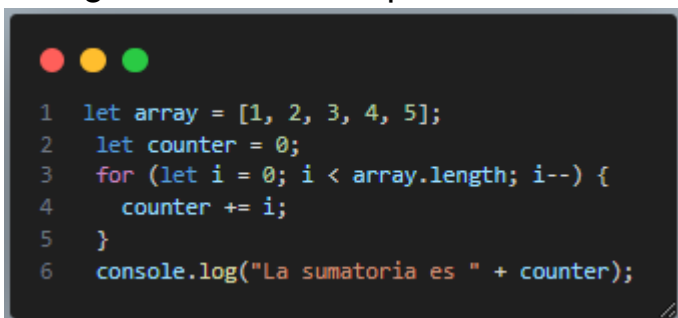
- ☒ 1.2
- ☐ 1.3
- ☐ 1.4
- ☐ 1.5

Respuesta correcta

1.2

3

La siguiente función imprime en consola "*La sumatoria es 15*"

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It contains the following JavaScript code:

```
1 let array = [1, 2, 3, 4, 5];  
2 let counter = 0;  
3 for (let i = 0; i < array.length; i++) {  
4   counter += i;  
5 }  
6 console.log("La sumatoria es " + counter);
```

- 1 / 1

- ☐ Verdadero
- ☒ Falso

Respuesta correcta

Falso

4 ¿Qué imprime la siguiente función?

```
1 let array = [1, 2, 3, 4, 5];
2 let ultimoIndex = array.length;
3 console.log("El último valor del array es " + array[ultimoIndex]);
4
5
```

1 / 1

- ☐ El último valor del array es 5
- ☐ El último valor del array es 1
- ☐ No imprime nada
- ☒ El último valor del array es undefined

Respuesta correcta

El último valor del array es undefined

5

Dado el siguiente código, que afirmación es correcta

```
1 let array = ["Manzana", "Pera", "Uva", "Durazno", "Aguacate"];
2
3 buscarIndex = function (valor) {
4   let index = -1;
5   for (let i = 0; i <= array.length; i++) {
6     if (array[i] == valor) {
7       index = i;
8       break;
9     }
10  }
11  return index;
12 };
```

1 / 1

- ☐ Al llamar a la función buscarIndex con el parámetro "Manzana", esta devuelve 1.
- ☐ El bucle for se repite 5 veces si valor es igual a "Uva"
- ☐ La máxima cantidad de veces que el bucle for se repite es 5.
- ☒ El bucle for se ejecuta 1 vez si valor es igual a "Manzana"

Respuesta correcta

El bucle for se ejecuta 1 vez si valor es igual a "Manzana"

6 Qué comando sirve para clonar el repositorio remoto de git y generar un repositorio local? 1 / 1

- ☐ git push
- ☐ git commit
- ☐ git add
- ☐ git copy
- ☒ git clone

Respuesta correcta

git clone

7Cuál de las siguientes afirmaciones es correcta? 1 / 1

- ☐ Cuando se ejecuta un git commit, se crea una versión de todos los archivos que se encuentren en el staging area en ese momento
- ☐ Con el comando git add se puede llevar un archivo en estado untracked hacia el staging area
- ☐ Con el comando git add se puede llevar un archivo en estado changed hacia el staging area
- ☐ Cuando un archivo no está bajo el control de git, se dice que su estado es untracked
- ☒ Todas las anteriores

Respuesta correcta

Todas las anteriores

8 Qué comando permite generar una versión con todos los archivos que se encuentren en ese momento en el Staging Area? 0 / 1

- ☐ git commit
- ☒ git add
- ☐ git push

- ☐ git clone

Respuesta correcta

git commit

9 Qué ejecución de git commit es correcta? 1 / 1

- ☐ git commit
- ☐ git commit archivo1.txt
- ☐ git commit -m se agrega la función calcularIVA
- ☒ git commit -m "se agrega la función calcularIVA"
- ☐ Todas las anteriores
- ☐ Ninguna de las anteriores

Respuesta correcta

git commit -m "se agrega la función calcularIVA"

10 La siguiente afirmación es verdadera o falsa? El comando **git log** nos permite ver el historial de commits del repositorio 1 / 1

- ☒ Verdadero
- ☐ Falso

Respuesta correcta

Verdadero

11 Verdadero o Falso El comando **git push** permite clonar el repositorio remoto en un repositorio local 1 / 1

- ☐ Verdadero
- ☒ Falso

Respuesta correcta

Falso

12 Verdadero o Falso El comando **git push** permite subir los commits del repositorio local hacia el repositorio remoto 1 / 1

☒ Verdadero

☐ Falso

Respuesta correcta

Verdadero

13

Dado el siguiente resultado de ejecutar git status, qué afirmación es correcta?

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   ejemplo/estilos.css
    new file:   ejemplo/funciones.js
    new file:   ejemplo/pagina1.html
```

0 / 1

- ☐ Existen 3 archivos untracked
- ☒ Existen 3 cambios pendientes de commit
- ☐ No existe ningún cambio pendiente de commit
- ☒ No existe ningún archivo en el staging area

Respuesta correcta

Existen 3 cambios pendientes de commit

14 Dado el siguiente resultado de ejecutar git status, qué afirmación es correcta?

```
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

0 / 1

- ⦿ No existen commits pendientes
- Existen archivos untracked
- Existen commits pendientes

Respuesta correcta

Existen commits pendientes

15

Dado el siguiente resultado de ejecutar git status, que comando utilizaría para que los archivos estilos.css y pagina2.html pasen al staging area?

```
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   ejemplo/funciones.js

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   ejemplo/estilos.css

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ejemplo/pagina2.html
```

1 / 1

- git add

- ☒ `git add .`
- ☐ `git commit`
- ☐ `git commit -m "se agregan estilos"`

Respuesta correcta

`git add .`

16 Dado el siguiente fragmento de código

```
if(vidas == 0){  
    console.log("GAME OVER");  
}else{  
    console.log("vidas: "+vidas);  
}
```

Si al ingresar al if, la variable vidas tiene un valor de 10. Qué imprime en consola?

1 / 1

- ☐ No imprime nada
- ☒ Imprime> vidas: 10
- ☐ Imprime> GAME OVER

Respuesta correcta

Imprime> vidas: 10

17 Dado el siguiente código

```
if(vidas = 0){  
    console.log("Pierde el juego");  
}
```

La siguiente afirmación es verdadera o falsa

El código compara si *vidas* es igual a 0, si cumple la condición imprime Pierde el juego 1 / 1

- ☐ Verdadero
- ☒ Falso

Respuesta correcta

Falso

18 Dado el siguiente código

```
if(estatura >= 1.20){  
    console.log("Puede ingresar al juego");  
}else{  
    console.log("No puede ingresar");  
}
```

En qué casos imprimiría Puede ingresar al juego

1 / 1

- ☐ Si estatura es 1.0
- ☒ Si estatura es 1.30
- ☒ Si estatura es 1.20

Respuesta correcta.

- Si estatura es 1.30
- Si estatura es 1.20

19

Dado el siguiente código

```

6      let nombre;
7      let componente = document.getElementById("txtNombre");
8      if(componente != null){
9          |       nombre=componente.value;
10     }

```

Qué afirmación es **INCORRECTA**?

1 / 1

- ☐ Si no existe en la página un componente con el id txtNombre, la función document.getElementById retorna null
- ☐ Con la validación de la línea 8, nos aseguramos que se ejecute la línea 9 solamente si se encuentra el componente con id txtNombre en la página
- ☐ La validación de la línea 8, permite que entre al if solamente si encontró el componente
- ☒ En este código siempre ejecuta la línea 9, sin importar si encuentra o no encuentra el componente en la línea 7

Respuesta correcta

En este código siempre ejecuta la línea 9, sin importar si encuentra o no encuentra el componente en la línea 7

20 Si en el momento de ingresar al if, la variable **valor** tiene 8. Cuál de los siguientes fragmentos de código imprimiría TOY DENTRO 1 / 1

- ☒
- ☐
- ☐
- ☐
- ☐
- ☐

Respuesta correcta.

•

21 Dado el siguiente fragmento

```
if(componente !== null){  
    console.log("COMPONENTE ENCONTRADO!");  
}
```

En qué casos imprimiría en consola COMPONENTE ENCONTRADO!

1 / 1

- ☒ Cuando la variable componente es distinta de null
- ☒ Si la variable componente tiene un valor de 100
- ☐ Si la variable componente tiene un valor de null
- ☒ Si la variable componente tiene un valor "Hola Mundo"

Respuesta correcta.

- Cuando la variable componente es distinta de null
- Si la variable componente tiene un valor de 100
- Si la variable componente tiene un valor "Hola Mundo"

22 Que fragmento de código representa correctamente el requerimiento: si el valor obtenido es diferente de -1, significa que encontró el elemento buscado. 1 / 1

- ☒
- ☐
- ☐

Respuesta correcta

23 Qué fragmento de código representa correctamente el requerimiento: No se permite ingresar valores negativos 1 / 1

- ☒
- ☐

Respuesta correcta

24 Qué fragmento de código representa correctamente el requerimiento: se aplica un descuento si compra 3 productos o más 1 / 1

- ☐
- ☒
- ☐
- ☐

Respuesta correcta

25

Dado el siguiente código

```
if(a>=b){  
    console.log("PUEDE PASAR");  
}
```

Qué valores de a y b, hacen que imprima PUEDE PASAR

1 / 1

- ☒ a vale 100 y b vale 100
- ☒ a vale 500 y b vale 300
- ☐ a vale 500 y b vale 800

Respuesta correcta.

- a vale 100 y b vale 100
- a vale 500 y b vale 300

26 Dado el siguiente código

```
if(ingresos>=1000 && ingresos<2000 ){  
    console.log("OK");  
}
```

En qué casos imprimiría OK

1 / 1

- ☐ Cuando ingresos tenga un valor de 2000
- ☒ Cuando ingresos tenga un valor de 1000
- ☐ Cuando ingresos tenga un valor de 500
- ☒ Cuando ingresos tenga un valor de 1500
- ☐ Cuando ingresos tenga un valor de 2500

Respuesta correcta.

- Cuando ingresos tenga un valor de 1000
- Cuando ingresos tenga un valor de 1500

27 Dado el siguiente código

```
if(ingresos>=1000 && ingresos<2000 ){  
    console.log("OK");  
}
```

En qué casos NO imprimiría OK

0 / 1

- ☐ Cuando ingresos tenga un valor de 2000
- ☒ Cuando ingresos tenga un valor de 1000
- ☐ Cuando ingresos tenga un valor de 500
- ☒ Cuando ingresos tenga un valor de 1500
- ☐ Cuando ingresos tenga un valor de 2500

Respuesta correcta.

- Cuando ingresos tenga un valor de 2000
- Cuando ingresos tenga un valor de 500
- Cuando ingresos tenga un valor de 2500

28 Dado el siguiente código

```
if(ingresos>1000 && aniosTrabajo >=4){  
    console.log("ES SUJETO DE CREDITO");  
}
```

En qué casos imprime ES SUJETO DE CREDITO

1 / 1

- ☐ Si ingresos es 2000, sin importar el valor de aniosTrabajo
- ☐ Si aniosTrabajo es 5, sin importar el valor de ingresos
- ☐ Si ingresos es 1000 y aniosTrabajo es 4
- ☒ Si ingresos es 1001 y aniosTrabajo es 4
- ☒ Si ingresos es 1500 y aniosTrabajo es 10
- ☐ Si ingresos es 1500 y aniosTrabajo es 2

Respuesta correcta.

- Si ingresos es 1001 y aniosTrabajo es 4
- Si ingresos es 1500 y aniosTrabajo es 10

29 Dado el siguiente código

```
if(edad<=14 || estatura <1.50){  
    console.log("NO PUEDE SUBIR AL JUEGO");  
}
```

En qué casos imprime NO PUEDE SUBIR AL JUEGO

1 / 1

- ☐ Si edad es 15 y estatura es 1.70
- ☒ Si edad es 14, sin importar la estatura
- ☐ Si estatura es 1.50 y edad es 18
- ☒ Si edad es 18 y estatura es 1.48

Respuesta correcta.

- Si edad es 14, sin importar la estatura
- Si edad es 18 y estatura es 1.48

30 Dado el siguiente código

```
if(edad<=14 || estatura <1.50){  
    console.log("NO PUEDE SUBIR AL JUEGO");  
}else{  
    console.log("SI PUEDE SUBIR AL JUEGO");  
}
```

En qué casos imprime SI PUEDE SUBIR AL JUEGO

1 / 1

- ☒ Si edad es 15 y estatura es 1.70
- ☐ Si edad es 14, sin importar la estatura
- ☒ Si estatura es 1.50 y edad es 18
- ☐ Si edad es 18 y estatura es 1.48

Respuesta correcta.

- Si edad es 15 y estatura es 1.70
- Si estatura es 1.50 y edad es 18

31 Dado el siguiente código

```
if(edad>=0 && edad <=12){  
    console.log("es un niño");  
}else if(edad > 12 && edad <18){  
    console.log("es un adolescente");  
}else if(edad>=18 && edad <=120){  
    console.log("es un adulto");  
}else{  
    console.log("el valor de edad no es corre  
}
```

En qué casos imprime es un adolescente?

1 / 1

- ☐ Cuando edad vale 10
- ☐ Cuando edad vale 12
- ☐ Cuando edad vale 18
- ☒ Cuando edad vale 15

Respuesta correcta

Cuando edad vale 15

32 Que fragmento de código representa mejor este requerimiento:

Se otorgará el crédito solamente a las personas que cumplan con todos los requisitos:

- Sus ingresos mensuales debe ser mayores a 1000
- Al menos tienen 3 años de trabajo

- Sus deudas totales tienen un valor inferior a 300

1 / 1

- ☐
- ☒
- ☐
- ☐

Respuesta correcta

33 Qué fragmento de código implementa correctamente el siguiente requerimiento:
Se permiten valores entre 10(incluido) y 100(excluido) 1 / 1

- ☐
- ☐
- ☒
- ☐

Respuesta correcta

34 Qué fragmentos de código implementa correctamente el siguiente requerimiento:
Se permiten valores entre 10(incluido) y 100(excluido). Elija 2 (Ya sea que verifique si se cumple e imprima dicho mensaje, o verifique que no se cumpla e imprima un mensaje diciendo que no se cumple lo requerido). 1 / 1

- ☐
- ☐
- ☒
- ☒

Respuesta correcta.

-

35 Dado el siguiente código

```
if(opcion==1 && opcion==2 ){  
    console.log("CONTINUAR");  
}
```

En qué casos imprimiría CONTINUAR

1 / 1

- ☒ Nunca imprimirá CONTINUAR
- ☐ Cuando opcion valga 1 y opcion valga 2
- ☐ Cuando opcion valga 1
- ☐ Cuando opcion valga 2
- ☐ Cuando opcion valga 5

Respuesta correcta

Nunca imprimirá CONTINUAR

36 Dado el siguiente código

```
if(a==3 || a!=3){  
    console.log("BIENVENIDO");  
}
```

La sentencia a continuación es verdadera o falsa

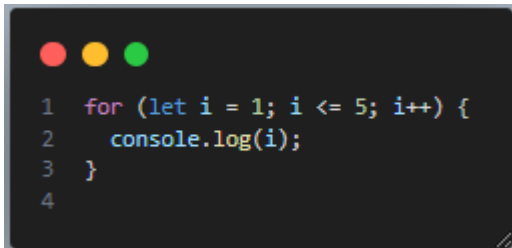
Siempre imprime BIENVENIDO porque siempre ingresa al if 1 / 1

- ☒ Verdadero
- ☐ Falso

Respuesta correcta

Verdadero

37 A partir del siguiente código.



```
1 for (let i = 1; i <= 5; i++) {  
2   console.log(i);  
3 }  
4
```

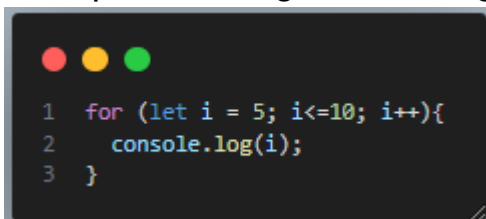
¿Cuál es el resultado esperado?

1 / 1

- ☒
- ☐
- ☐
- ☐ Se crea un loop infinito

Respuesta correcta

38 A partir del siguiente código.



```
1 for (let i = 5; i<=10; i++){  
2   console.log(i);  
3 }
```

¿Cuál es el resultado esperado??

1 / 1

- ☐
- ☐
- ☒
- ☐ Se crea un loop infinito.

Respuesta correcta

39

A partir del siguiente código

```
1 for (let i = 0; i <= 10; i++) {  
2   let imprimir = i + 2;  
3   console.log(imprimir);  
4 }
```

¿Cuál es el resultado esperado?

1 / 1

- ☐
- ☐
- ☒
- ☐

Respuesta correcta

40 A partir del siguiente código

```
1 for (let i = 9; i > 0; i--) {  
2   imprimir=i-1;  
3   console.log(imprimir);  
4 }
```

¿Cuál es el resultado esperado?

1 / 1

- ☒
- ☐
- ☐
- ☐

Respuesta correcta

41 ¿El siguiente código produce un loop infinito?

```
1 for (let i = 5; i>0; i --) {  
2   i = i + 2;  
3   console.log(i);  
4 }
```

- 1 / 1

- ☒ Verdadero
- ☐ Falso

Respuesta correcta

Verdadero

42 De las siguientes secciones de código cual nos serviría para conocer la sumatoria de los pares del 2 al 10, y dicho resultado se muestre en un solo console.log al terminar el For. 1 / 1

- ☒
- ☐
- ☐
- ☐ Se crea un loop infinito

Respuesta correcta

43 A partir del siguiente resultado

5
10
15
20
25
30
35
40

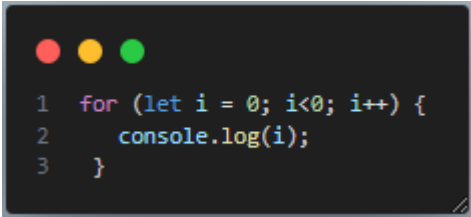
Cuál sería el código que lo produce.

1 / 1

- ☐
- ☐
- ☐
- ☒

Respuesta correcta

44 ¿El siguiente for si imprime algún *console.log()*?



```
1 for (let i = 0; i<0; i++) {  
2   console.log(i);  
3 }
```

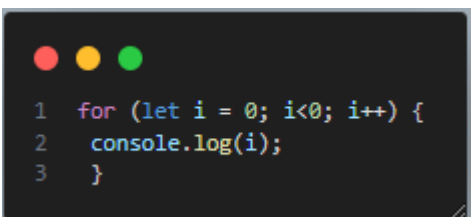
- 1 / 1

- ☐ Verdadero
- ☒ Falso

Respuesta correcta

Falso

45 ¿Qué cambio se le debe aplicar al siguiente código para que imprima solo una vez 0?



```
1 for (let i = 0; i<0; i++) {  
2   console.log(i);  
3 }
```

1 / 1

- ☐
- ☐
- ☒
- ☐

Respuesta correcta