

Nombre: Iskander Emilio Mercader Olivares

Instrucción. Aplica JavaScript para responder los siguientes ejercicios. Integra tu respuesta en este documento y comparte tu enlace GitHub con los códigos:

---Selección---

----centinelas en bucles----

-----Bucles while-----

-----Bucles for-----

-----Ejercicios de bucles/loops-----

-----Cadenas (Strings)----

-----Arreglos-----

-----Funciones-----

Una función es un grupo de enunciados diseñados para llevar a cabo una tarea específica. Las funciones son los componentes básicos de los programas. Son importantes porque permiten a los programadores almacenar funcionalidades útiles que se pueden invocar en una sola línea de código: la llamada a la función.

Sintaxis de funciones: definición y llamada de funciones

Una función solo necesita definirse una vez, pero se puede llamar varias veces. Cada vez que se llama a una función, se ejecutan las instrucciones que componen la función. Esto significa que las funciones pueden evitar que los programadores tengan que repetir las mismas líneas de código cada vez que necesitan realizar una tarea específica. Por lo tanto, las funciones se pueden utilizar para evitar la duplicación de código.

La sintaxis general para definir una nueva función es la siguiente:

```
function <function-name> ([parameters]) {  
    statement(s)  
}
```

NOTA: Una función es un fragmento corto de código reutilizable que lleva a cabo una tarea específica cuando se llama.

Estudie las dos definiciones de funciones que se muestran a continuación y responda las preguntas que siguen.

| | |
|---|--|
| <pre>function displayGreeting1(msg) { console.log(msg); }</pre> | <pre>function displayGreeting2(msg1, msg2) { console.log(msg1); console.log(msg2); }</pre> |
| <i>displayGreeting1</i> | <i>displayGreeting2</i> |

| Bloque de código | Resultado esperado |
|---|---------------------------------|
| a) displayGreeting1("Good evening, Jose"); | Good evening, Jose |
| b) let str = "Good evening, Jose"; displayGreeting1(str); | Good evening, Jose |
| c) let name = "Jose"; displayGreeting1("Good evening", name); | Good evening |
| d) displayGreeting2("Good evening", " Jose"); | Good evening Jose |
| e) let name = "Dave"; displayGreeting2("Good evening", name); | Good evening Dave |
| f) let str = "Good evening, Dave"; displayGreeting2(str); | Good evening, Dave Undefined |
| g) let sum = 2+3; displayGreeting1(sum); | 5 |
| h) let sum = "2+3"; displayGreeting1(sum); | 2+3 |
| i) let sum = "2+3"; displayGreeting2(sum, 2+3); | 2+3 5 |
| j) displayGreeting2("2+3", "equals", "5"); | 2+3 equals |

Describe lo que está mal (si es que hay algo) con cada uno de los siguientes fragmentos de código y, en cada caso, describa una solución (si corresponde).

| | |
|---|---|
| a) <pre>function displayMessage() { console.log(msg); } displayMessage("I am Sam");</pre> | <p>La variable msg no está declarada, se solucionaría agregándola a la función.</p> <pre>function displayMessage(msg) { console.log(msg); }</pre> |
| b) <pre>function displayMessage(msg) { console.log(msg); } displayMessage();</pre> | <p>El resultado es undefined. Hay que agregarle un mensaje al uso de la función.</p> <pre>displayMessage("Hola");</pre> |
| c) <pre>function displayMessage(msg) { console.log(msg); } displayMessage("I am", "Sam");</pre> | <p>El resultado es solo el primer argumento. Hay que agregarle otro a la función.</p> <pre>function displayMessage(msg, msg2) { console.log(msg, msg2); }</pre> |
| d) <pre>function displayMessage(msg) { console.log(message); } displayMessage("I am Sam");</pre> | <p>La variable message no está declarada, se solucionaría cambiando el nombre del argumento en la función.</p> <pre>function displayMessage(message) { console.log(message); }</pre> |

| | |
|---|--|
| <p>e)</p> <pre>function displayMessage(msg) { console.log(msg); } displayMessage(I am Sam);</pre> | <p>El argumento en el uso de la función debe entrecomillarse.</p> <pre>displayMessage("I am Sam");</pre> |
| <p>f)</p> <pre>function displayMessage(msg1, msg2) { console.log(msg1); } displayMessage("I am", "Sam");</pre> | <p>El resultado es "I am". La impresión en consola debe incluir ambas variables.</p> <pre>console.log(msg1, msg2);</pre> |

Valores devueltos (return)

PUNTO CLAVE: El valor devuelto de una función se puede guardar para su posterior procesamiento haciendo que la llamada a la función forme parte de una instrucción de asignación.

Analiza las siguientes dos definiciones de función para add y responde las preguntas que siguen.

| | |
|--|---|
| <pre>function add1(n1, n2) { return n1+n2; }</pre> | <pre>function add2(n1, n2) { let sum = n1+n2; console.log(n1+"+"+n2+"="+sum); }</pre> |
|--|---|

¿Qué función es mejor en tu opinión? ¿Por qué? Escriba una línea de código para llamar a add1: guarde su respuesta en una variable, por ejemplo, answer. Ahora intente hacer lo mismo para add2. **¿Con qué problema te encuentras?**

Es mejor la primera función debido a que sigue la buena convención de que el valor de las funciones utilice un return y no se imprima. El problema con el que nos encontramos a la hora de guardar los valores en variables es que en la función 2 depende del formato de la función, siendo que si esta imprime algún otro valor como una instrucción al usuario, esta generará problemas con la variable.

```

main.js
1 function add1(n1, n2) {
2   return n1+n2;
3 }
4
5 console.log(add1(1,2));
6
7 function add2(n1, n2) {
8   let sum = n1+n2;
9   console.log(n1+"+"+n2+"="+sum);
10 }
11
12 add2(1,2);
13
14 let resultadoUno = (add1(1,2));
15 console.log(resultadoUno)
16
17 let resultadoDos = (add2(1,2));
  
```

Output

```

3
1+2=3
3
1+2=3

=== Code Execution Successful ===
  
```

Describe lo que está mal (si es que hay algo) con cada uno de los siguientes fragmentos de código y, en cada caso, describe una solución (si corresponde).

| | |
|---|--|
| <p>a)</p> <pre>function add(n1, n2) { let sum = n1+n2; return sum; } let result = add(8, 3); console.log(sum);</pre> | <p>La variable sum no está declarada en la impresión final, es necesario cambiar la variable por la que guarda el resultado de la función.</p> <pre>console.log(result);</pre> |
| <p>b)</p> <pre>function add(n1, n2) { let sum = n1+n2; return; } let result = add(8, 3); console.log(result);</pre> | <p>El resultado es undefined. Al return se le debe agregar la variable con el resultado.</p> <pre>return sum;</pre> |
| <p>c)</p> <pre>function add(n1, n2) { let sum = n1+n2; return sum; } let result = add(8, 3);</pre> | <p>No se imprime el resultado. Se le debe agregar la función para imprimir en consola.</p> <pre>console.log(result);</pre> |
| <p>d)</p> <pre>function add(n1, n2) { return(n1+n2); } console.log(add(8, 3));</pre> | <p>Se imprime el resultado correctamente, no hay ningún error.</p> |

| | |
|---|---|
| e) | |
| <pre>function add(n1, n2) { let sum = n1+n2; return sum; } let result = add(8, add(2,1)); console.log(result);</pre> | Se imprime el resultado correctamente, no hay ningún error. |

Funciones booleanas

Una función booleana es una función que devuelve verdadero o falso. Por lo general, se utilizan como abstracción para algún tipo de prueba. Por ejemplo, podríamos escribir una función booleana para determinar si un número dado es primo o no.

¡Reto!

Implemente las siguientes funciones booleanas. Pruebe su código usando los datos ingresados a través del mensaje.

| | |
|----------------------------------|--|
| <code>isLessThan(a, b)</code> | Una función que devuelve <code>true</code> si <code>a</code> es menor que <code>b</code> , <code>false</code> en caso contrario. |
| <code>isGtREqual2(a, b)</code> | Función que devuelve <code>true</code> si <code>a</code> es mayor o igual que <code>b</code> , <code>false</code> en caso contrario. |
| <code>isSpeeding(speed)</code> | Función que devuelve <code>true</code> si el valor de <code>speed</code> es mayor que 120, <code>false</code> en caso contrario. |
| <code>isTeenager(age)</code> | Una función que devuelve <code>true</code> si el valor de <code>age</code> está entre 13 y 18 inclusive, <code>false</code> en caso contrario. |
| <code>isGoodMusic(artist)</code> | Una función que devuelve <code>true</code> si el valor de <code>artist</code> es miembro de una matriz que contiene los nombres de todos tus artistas musicales favoritos, <code>false</code> en caso contrario. |

```
function isLessThan(a, b) {
  let resultado = false;

  if (a < b) {
    resultado = true;
  } else {
```

```
    resultado = false;

    }

    return resultado;
}

console.log(isLessThan(2, 4));
console.log(isLessThan(5, 4));
```

```
function isGtREqual2(a, b) {
    if (a >= b) {
        resultado = true;
    } else {
        resultado = false;
    }

    return resultado;
}
```

```
console.log(isGtREqual2(5, 4));
console.log(isGtREqual2(2, 4));
```

```
function isSpeeding(speed) {
    let resultado = false;

    if (speed > 120) {
        resultado = true;
    } else {
        resultado = false;
    }
}
```

```
    return resultado;
```

```
}
```

```
console.log(isSpeeding(180));
```

```
console.log(isSpeeding(115));
```

```
function isTeenager(age) {
```

```
    let resultado = false;
```

```
    if (age >= 13 && age <= 18) {
```

```
        resultado = true;
```

```
    } else {
```

```
        resultado = false;
```

```
    }
```

```
    return resultado;
```

```
}
```

```
console.log(isTeenager(14));
```

```
console.log(isTeenager(19));
```

```
function isGoodMusic(artist) {
```

```
    const artistasFavoritos = ["Muse", "Opeth", "Blur", "Leprous", "Magma"];
```

```
    if (artistasFavoritos.includes(artist)) {
```

```
        resultado = true;
```

```
    } else {
```

```
        resultado = false;
```

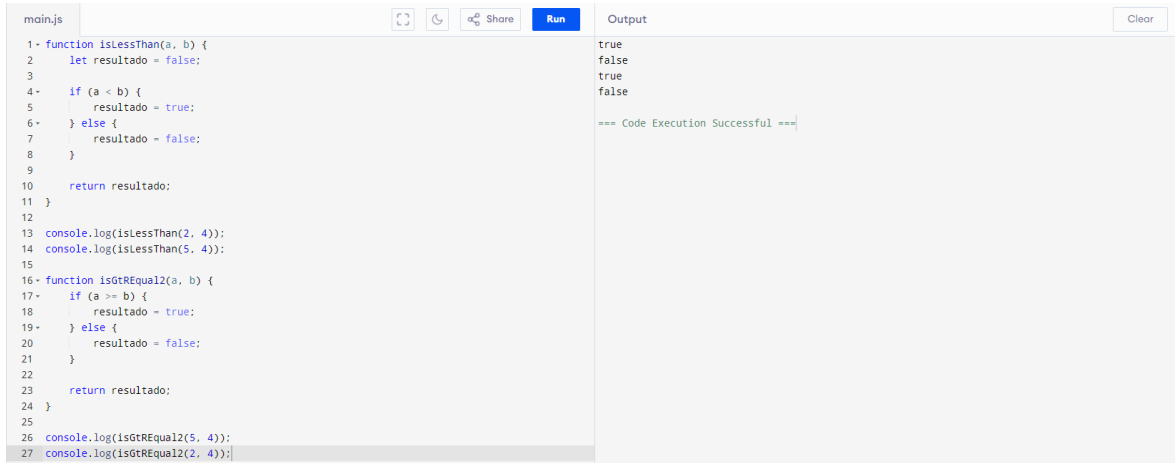
```
    }
```

```
    return resultado;
```

}

```
console.log(isGoodMusic("Blur"));
```

```
console.log(isGoodMusic("The Beatles"));
```

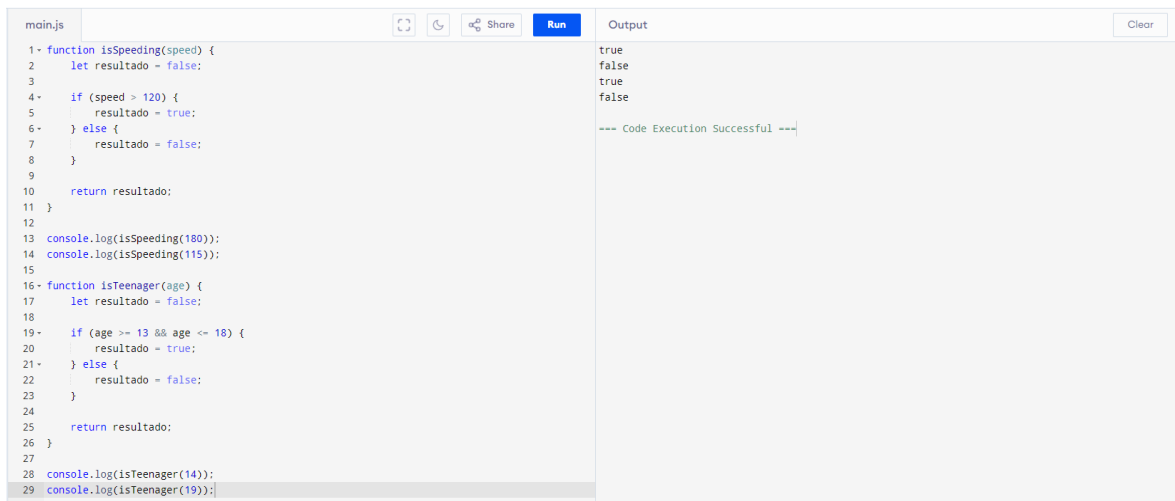


```
main.js
1 • function isLessThan(a, b) {
2   let resultado = false;
3
4   if (a < b) {
5     resultado = true;
6   } else {
7     resultado = false;
8   }
9
10  return resultado;
11 }
12
13 console.log(isLessThan(2, 4));
14 console.log(isLessThan(5, 4));
15
16 • function isGreaterEqual2(a, b) {
17   if (a >= b) {
18     resultado = true;
19   } else {
20     resultado = false;
21   }
22
23   return resultado;
24 }
25
26 console.log(isGreaterEqual2(5, 4));
27 console.log(isGreaterEqual2(2, 4));
```

Output

```
true
false
true
false

=== Code Execution Successful ===
```

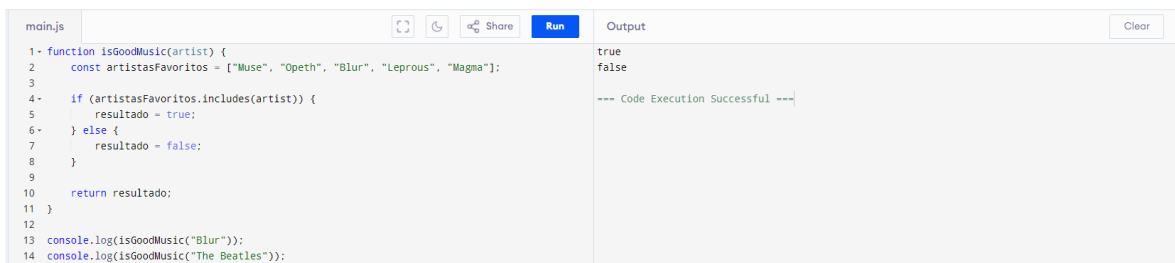


```
main.js
1- function isSpeeding(speed) {
2-   let resultado = false;
3-
4-   if (speed > 120) {
5-     resultado = true;
6-   } else {
7-     resultado = false;
8-   }
9-
10-  return resultado;
11- }
12-
13- console.log(isSpeeding(180));
14- console.log(isSpeeding(115));
15-
16- function isTeenager(age) {
17-   let resultado = false;
18-
19-   if (age >= 13 && age <= 18) {
20-     resultado = true;
21-   } else {
22-     resultado = false;
23-   }
24-
25-   return resultado;
26- }
27-
28- console.log(isTeenager(14));
29- console.log(isTeenager(19));
```

Output

```
true
false
true
false

--- Code Execution Successful ---
```



```
main.js
1- function isGoodMusic(artist) {
2-   const artistasFavoritos = ["Muse", "Opeth", "Blur", "Leprous", "Magma"];
3-
4-   if (artistasFavoritos.includes(artist)) {
5-     resultado = true;
6-   } else {
7-     resultado = false;
8-   }
9-
10-  return resultado;
11- }
12-
13- console.log(isGoodMusic("Blur"));
14- console.log(isGoodMusic("The Beatles"));
```

Output

```
true
false

--- Code Execution Successful ---
```

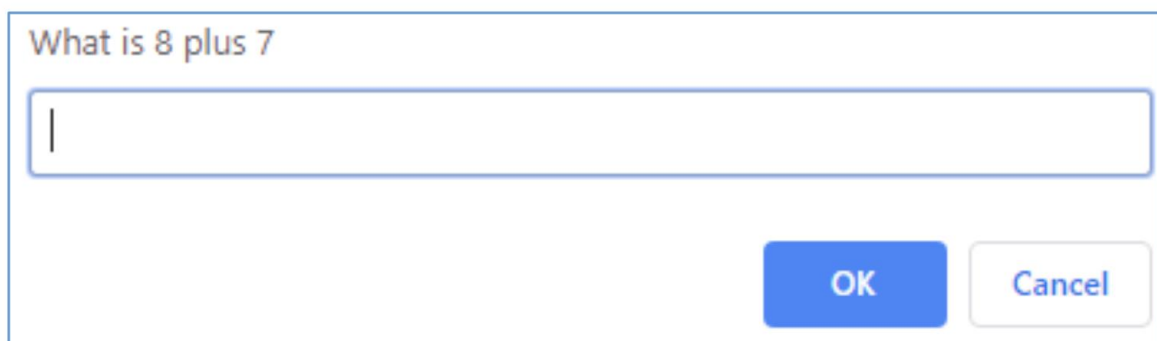
PRÁCTICA 1

Aprendizaje Asistido por Computadora

La Instrucción Asistida por Computadora (CAI) está desempeñando un papel cada vez más importante en la educación. Para esta actividad, debe **escribir una aplicación JavaScript que ayude a los alumnos de primaria a aprender aritmética**. Llamamos a la aplicación Computer Aided Learning o CAL para abreviar.

Considere el siguiente código en el que deberá realizar cambios para desarrollar una solución mejorada.

Quando se ejecuta, el programa muestra una ventana emergente que pide al usuario que ingrese la respuesta a un problema de suma simple que involucra dos números enteros de un solo dígito seleccionados al azar entre 0 y 9 inclusive. Por ejemplo

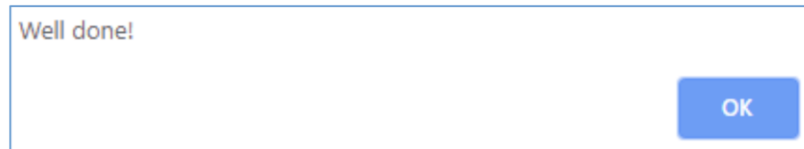


What is 8 plus 7

OK Cancel

El usuario escribe su respuesta y clic en **OK**. Si la respuesta es correcta, el programa muestra un mensaje de motivación en una nueva ventana emergente como la que se muestra a continuación. El mensaje que se muestra se selecciona aleatoriamente de la siguiente lista (matriz) de cadenas:

- Well done!
- Very good!
- Correct!
- Keep it up!
- Nice work!



Si la respuesta es incorrecta, el programa muestra el mensaje `Respuesta incorrecta`. Vuelva a intentarlo repetidamente hasta que el usuario finalmente ingrese la respuesta correcta.

A continuación se muestra el programa que se le proporciona como base de esta práctica 1:

```
// PASO 1. Generar la pregunta
// Genera 2 números aleatorios
let n1 = Math.floor(Math.random() * 10); // 0 <= n1 < 10
let n2 = Math.floor(Math.random() * 10); // 0 <= n2 < 10

// PASO 2. Haga la pregunta al usuario y obtenga una respuesta
// n1 y n2 se convierten en cadenas para que puedan mostrarse como
// parte de la cadena de solicitud (prompt)
let problemInWords = "What is " + String(n1) + " + " + String(n2); // Q1
console.log(problemInWords);
let userResponse = Number(prompt(problemInWords));
console.log("User entered %d", userResponse); // Esto es para fines de
depuración

// PASO 3. Procesar la respuesta
// Calcula la respuesta correcta. Entonces...
// ... siempre y cuando la respuesta del usuario sea diferente a la
// respuesta de la computadora ... le dice al usuario que está equivocado
// y ...solicita al usuario otra respuesta
let correctAnswer = n1 + n2; // Calcular la respuesta correcta
while (userResponse != correctAnswer) {
    console.log("Wrong answer! Try again.");
    userResponse = Number(prompt(problemInWords));
    console.log("User entered %d", userResponse); // Esto es para fines
// de depuración
} // end while

// PASO 4. Mostrar un mensaje de motivación seleccionado al azar
const messages = ["Well done!", "Very good!", "Correct!", "Keep it up!",
"Nice work!"];
const r = Math.floor(Math.random() * messages.length);
```

```
console.log(messages[r]);  
alert(messages[r]);
```

Listado de código inicial de CAL v1.0

Nota. Implemente funciones en la práctica 1

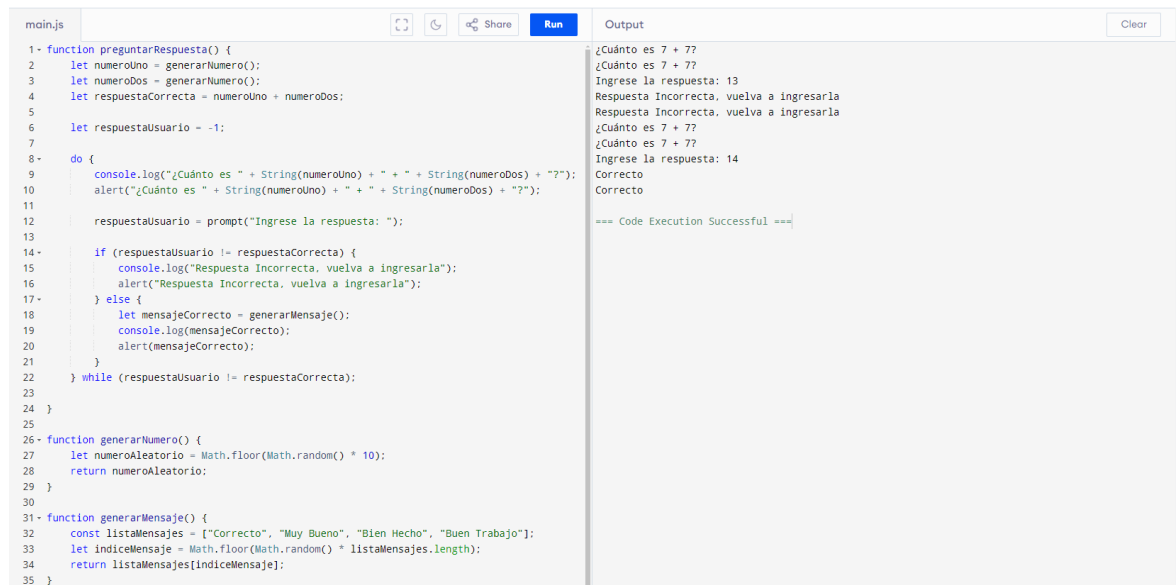
```
function preguntarRespuesta() {  
    let numeroUno = generarNumero();  
    let numeroDos = generarNumero();  
    let respuestaCorrecta = numeroUno + numeroDos;  
  
    let respuestaUsuario = -1;  
  
    do {  
        console.log("¿Cuánto es " + String(numeroUno) + " + " + String(numeroDos) + "?");  
        alert("¿Cuánto es " + String(numeroUno) + " + " + String(numeroDos) + "?");  
  
        respuestaUsuario = prompt("Ingrese la respuesta: ");  
  
        if (respuestaUsuario != respuestaCorrecta) {  
            console.log("Respuesta Incorrecta, vuelva a ingresarla");  
            alert("Respuesta Incorrecta, vuelva a ingresarla");  
        } else {  
            let mensajeCorrecto = generarMensaje();  
            console.log(mensajeCorrecto);  
            alert(mensajeCorrecto);  
        }  
    } while (respuestaUsuario != respuestaCorrecta);  
}
```

```
function generarNumero() {  
    let numeroAleatorio = Math.floor(Math.random() * 10);
```

```
    return numeroAleatorio;
}
```

```
function generarMensaje() {
    const listaMensajes = ["Correcto", "Muy Bueno", "Bien Hecho", "Buen Trabajo"];
    let indiceMensaje = Math.floor(Math.random() * listaMensajes.length);
    return listaMensajes[indiceMensaje];
}
```

```
preguntarRespuesta();
```



The screenshot shows a code editor with a file named 'main.js'. The code defines three functions: `preguntarRespuesta()`, `generarNumero()`, and `generarMensaje()`. `preguntarRespuesta()` is a loop that asks the user a math problem (7 + 7), checks the answer, and provides feedback. `generarNumero()` generates a random number between 0 and 10. `generarMensaje()` returns a random message from an array. The 'Output' panel shows the execution of `preguntarRespuesta()`, displaying the prompt, the user's input (13 and 14), and the feedback messages ('Respuesta Incorrecta, vuelva a ingresarla' and 'Correcto').

```
1- function preguntarRespuesta() {
2   let numeroUno = generarNumero();
3   let numeroDos = generarNumero();
4   let respuestaCorrecta = numeroUno + numeroDos;
5
6   let respuestaUsuario = -1;
7
8   do {
9       console.log("¿Cuánto es " + String(numeroUno) + " + " + String(numeroDos) + "?");
10      alert("¿Cuánto es " + String(numeroUno) + " + " + String(numeroDos) + "?");
11
12      respuestaUsuario = prompt("Ingrese la respuesta: ");
13
14      if (respuestaUsuario != respuestaCorrecta) {
15          console.log("Respuesta Incorrecta, vuelva a ingresarla");
16          alert("Respuesta Incorrecta, vuelva a ingresarla");
17      } else {
18          let mensajeCorrecto = generarMensaje();
19          console.log(mensajeCorrecto);
20          alert(mensajeCorrecto);
21      }
22  } while (respuestaUsuario != respuestaCorrecta);
23  }
24
25
26- function generarNumero() {
27   let numeroAleatorio = Math.floor(Math.random() * 10);
28   return numeroAleatorio;
29  }
30
31- function generarMensaje() {
32   const listaMensajes = ["Correcto", "Muy Bueno", "Bien Hecho", "Buen Trabajo"];
33   let indiceMensaje = Math.floor(Math.random() * listaMensajes.length);
34   return listaMensajes[indiceMensaje];
35 }
```

Output

```
¿Cuánto es 7 + 7?
¿Cuánto es 7 + 7?
Ingrese la respuesta: 13
Respuesta Incorrecta, vuelva a ingresarla
Respuesta Incorrecta, vuelva a ingresarla
¿Cuánto es 7 + 7?
¿Cuánto es 7 + 7?
Ingrese la respuesta: 14
Correcto
Correcto
=== Code Execution Successful ===
```