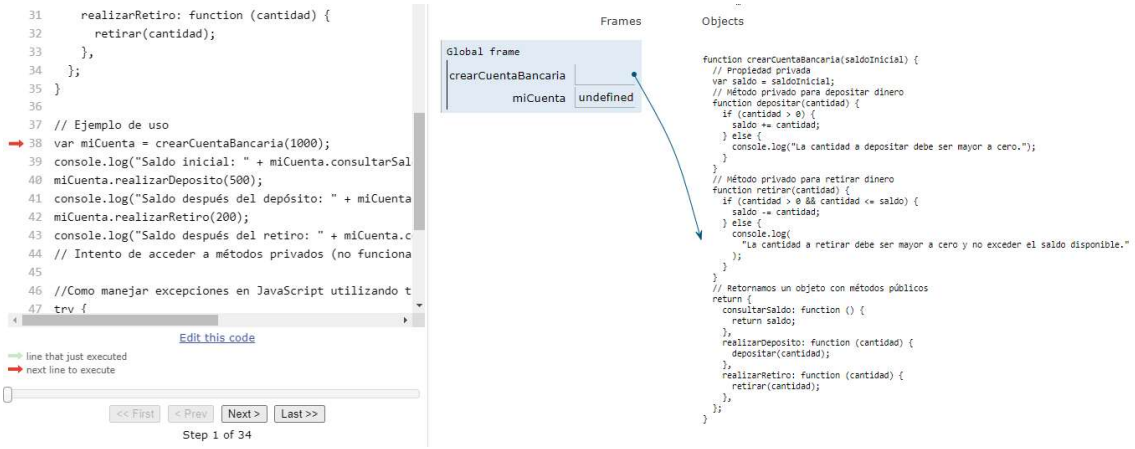
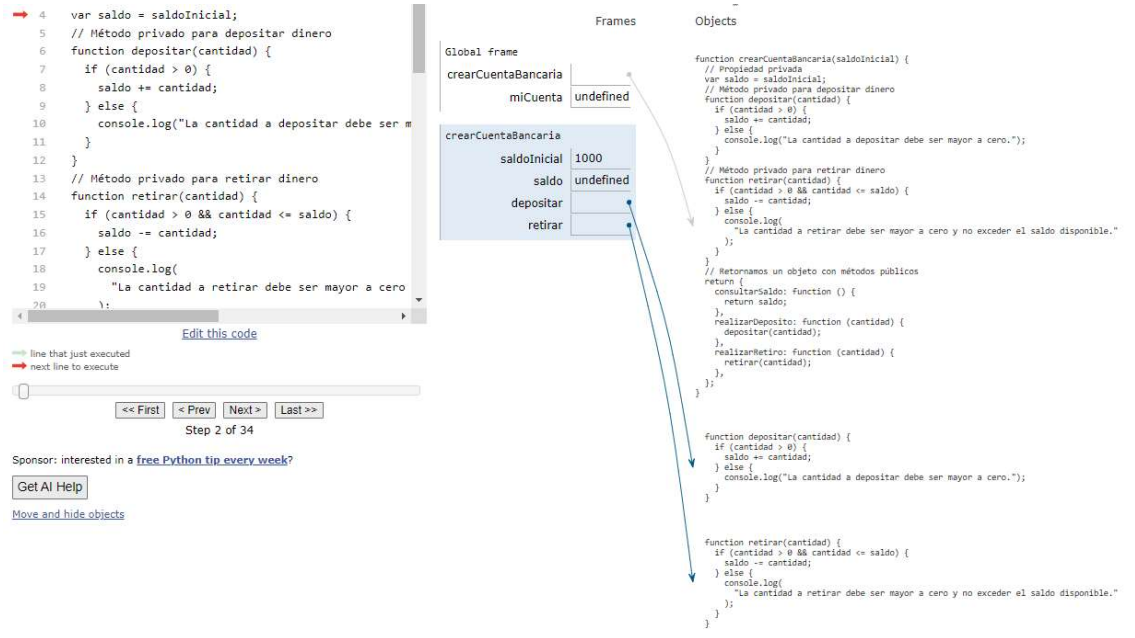
TAREA 8

Paso 1 de 34: Se define la función crearCuentaBancaria que toma saldoInicial como parámetro.

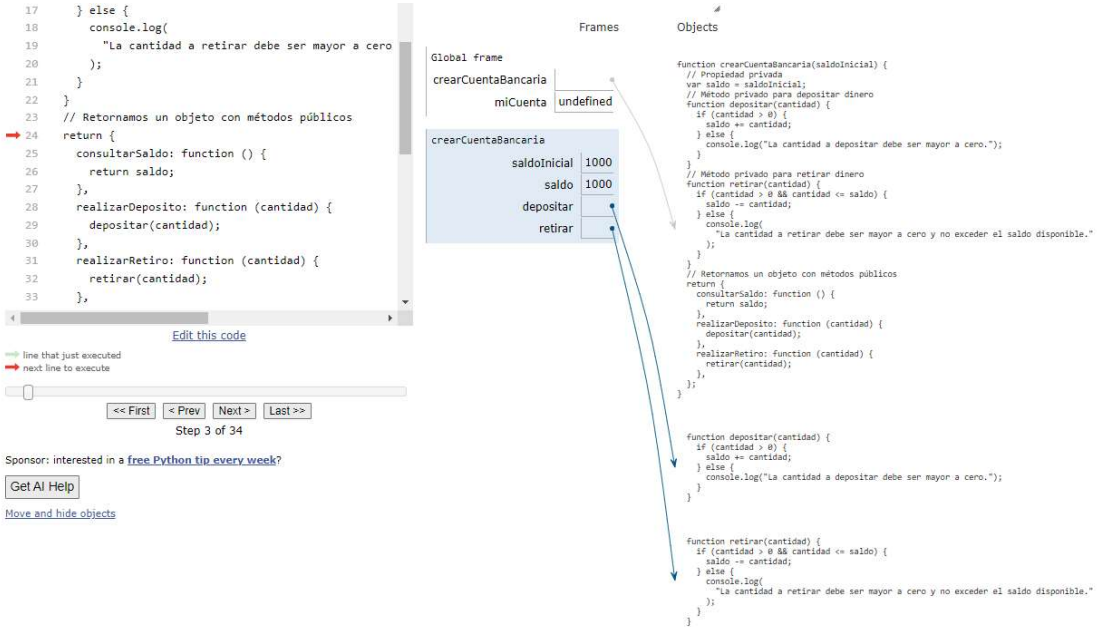


Paso 2 de 34: Se declara una variable saldo que almacenará el saldo inicial de la cuenta.

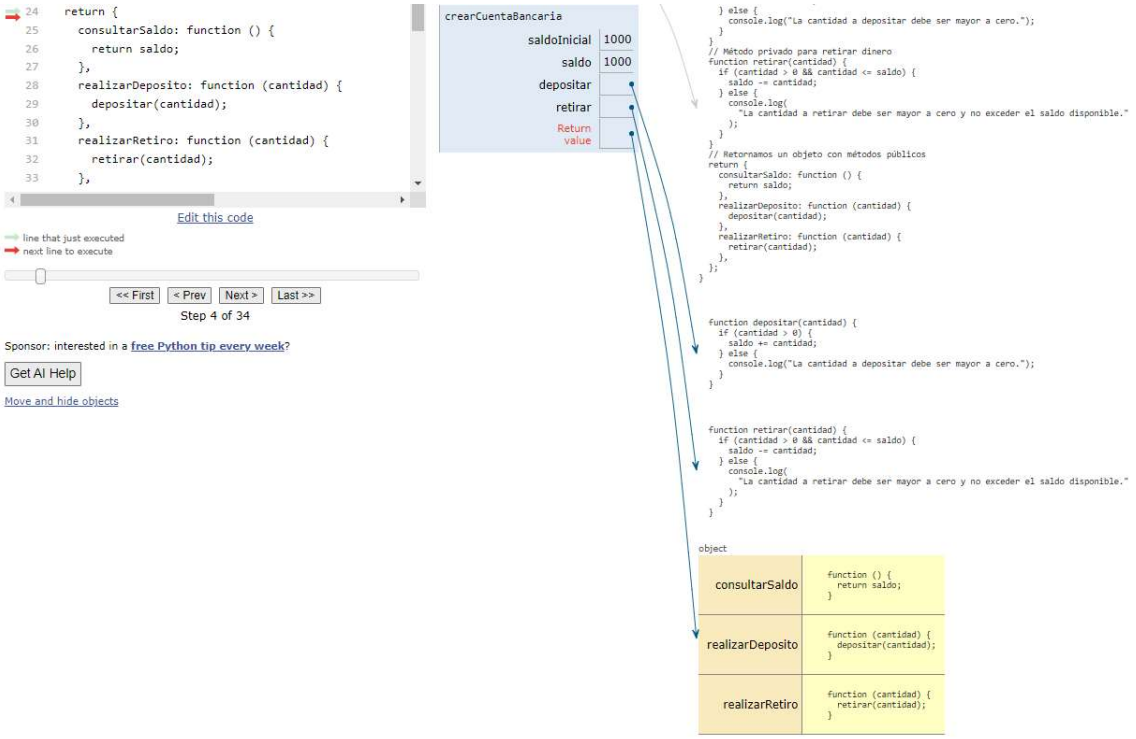


.

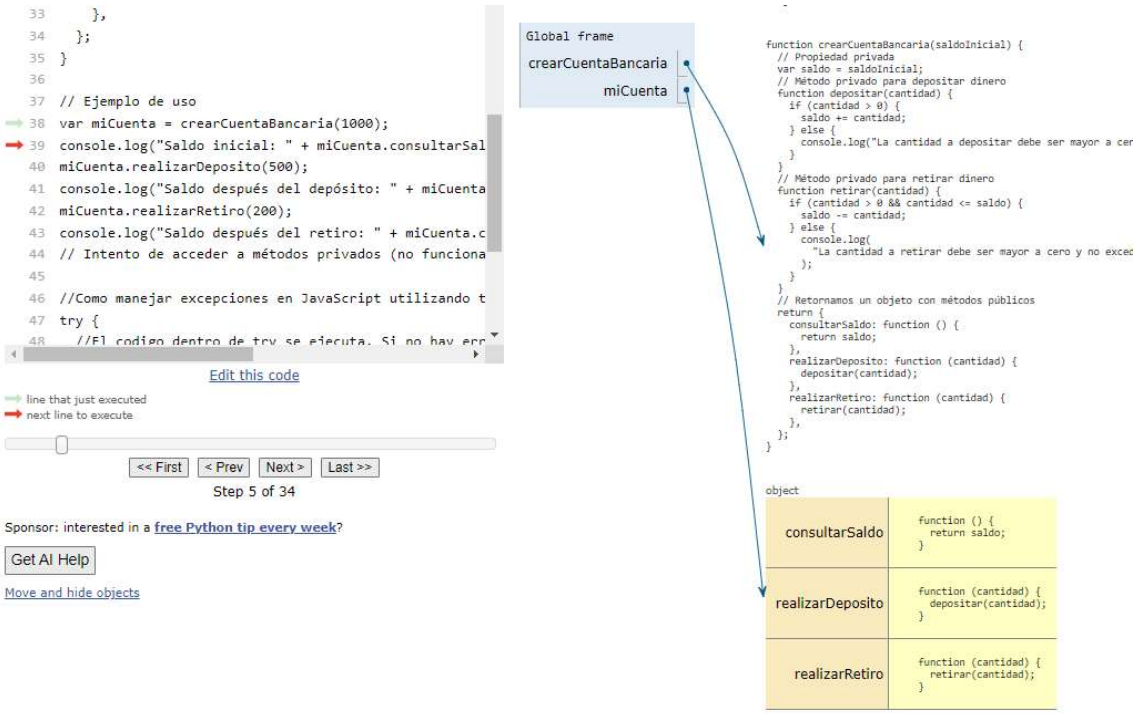
Paso 3 de 34: Se retorna un objeto que contiene la funcion consultar saldo



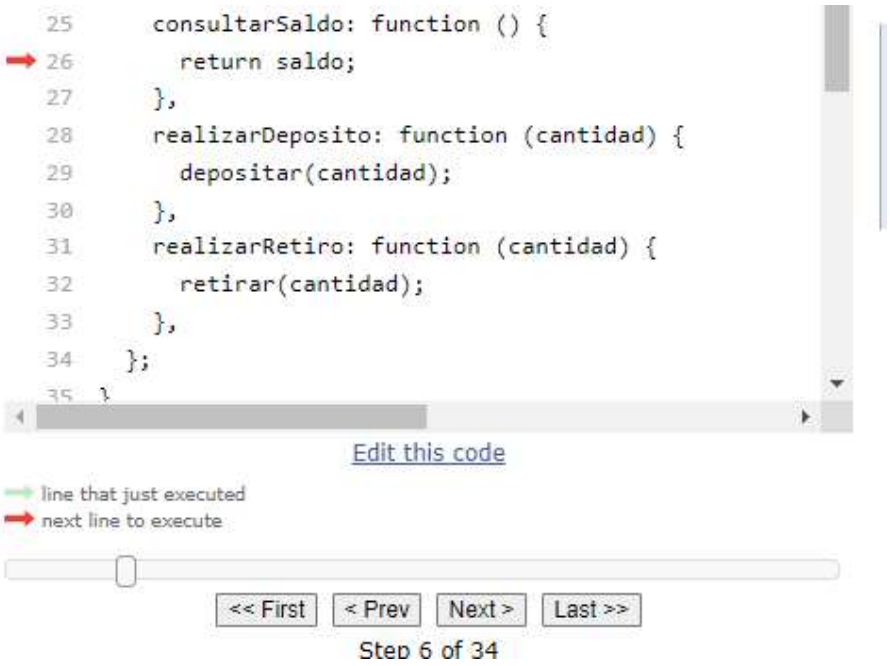
Paso 4 de 34: Ejecuta la funcion consultar saldo



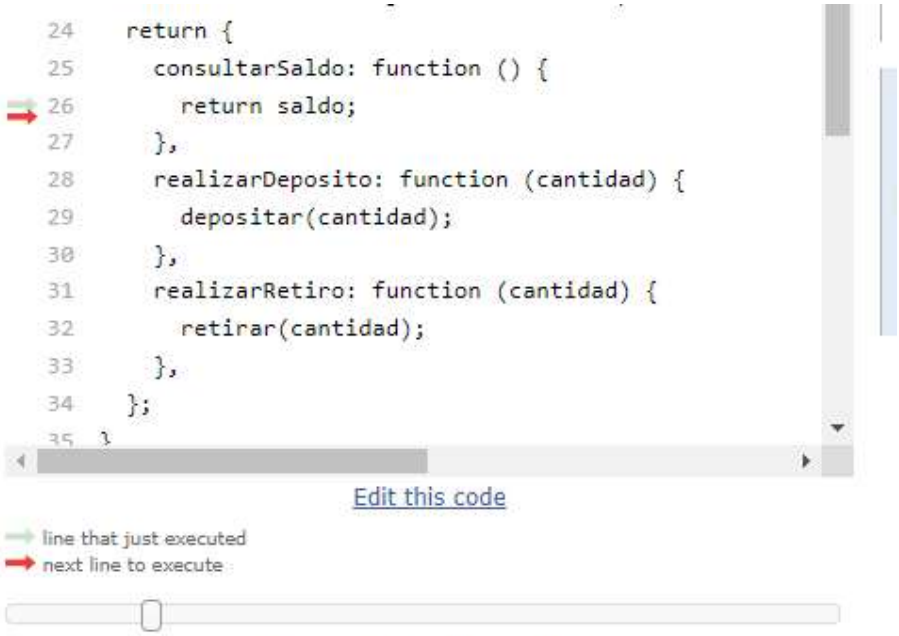
Paso 5 de 34: Se imprime el saldo inicial de la cuenta utilizando miCuenta.consultarSaldo().



Paso 6 de 34: regresa el saldo de la cuenta.



Paso 7 de 34: regresa el saldo de la cuenta.

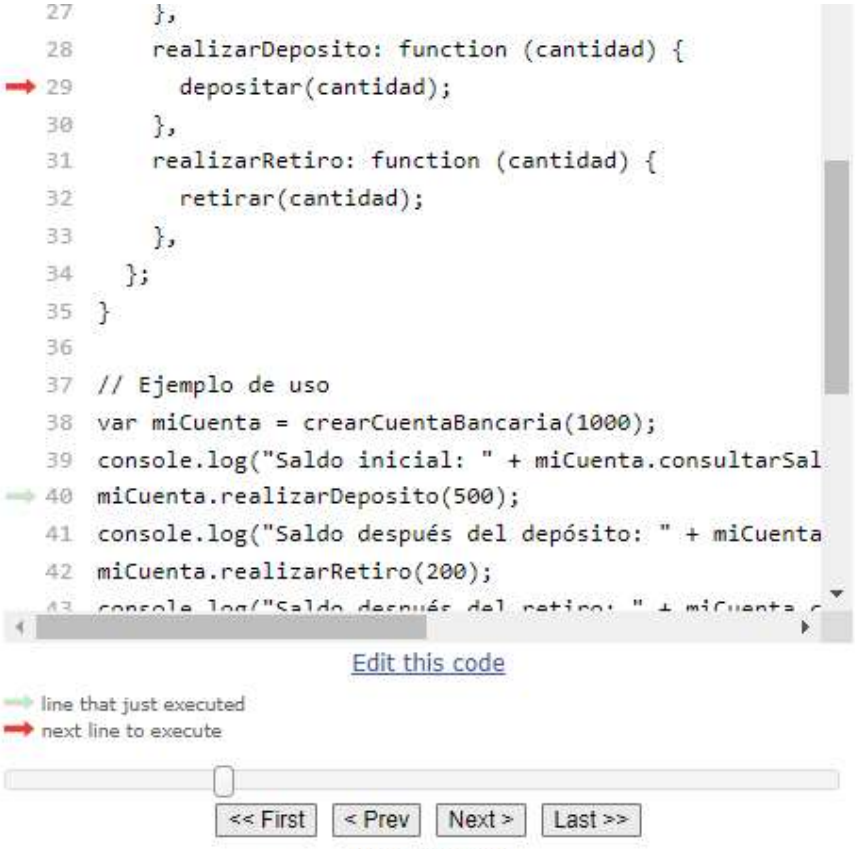


x

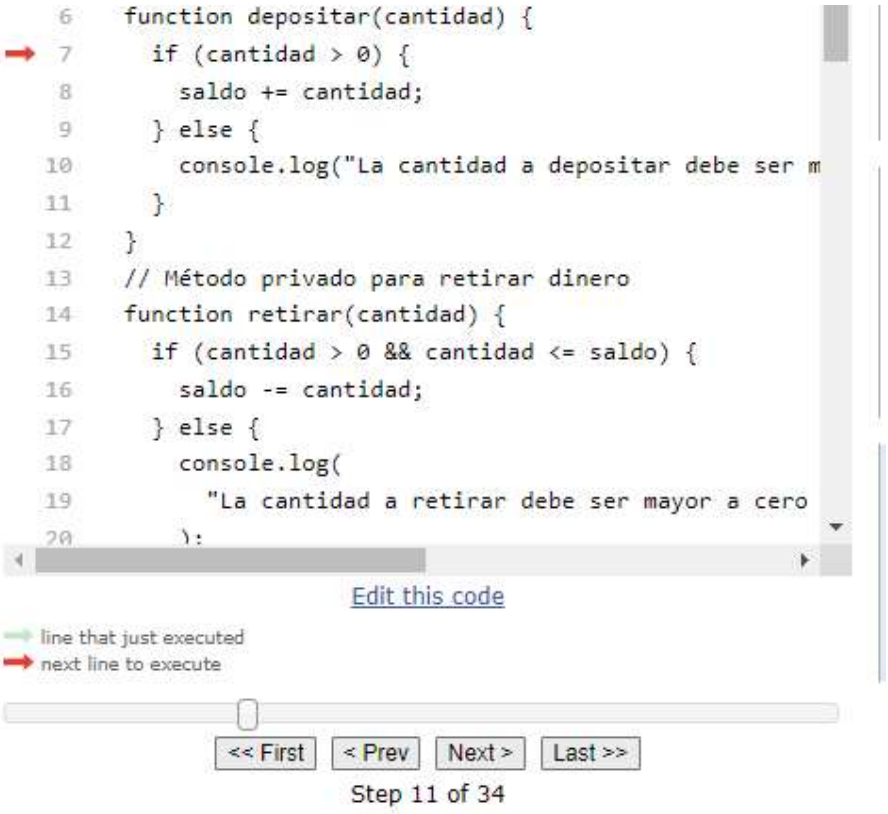
Paso 8 de 34: Se imprime el saldo inicial de la cuenta utilizando miCuenta.consultarSaldo().

Paso 9 de 34: Se realiza un depósito de 500 utilizando miCuenta.realizarDeposito(500).

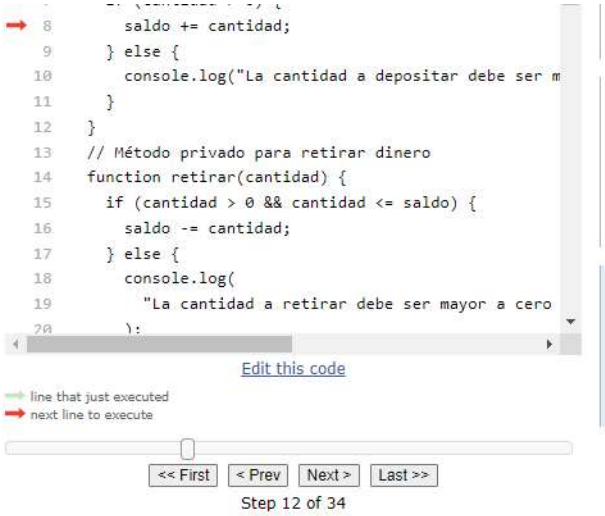
Paso 10 de 34: define la funcion de realizar el deposito



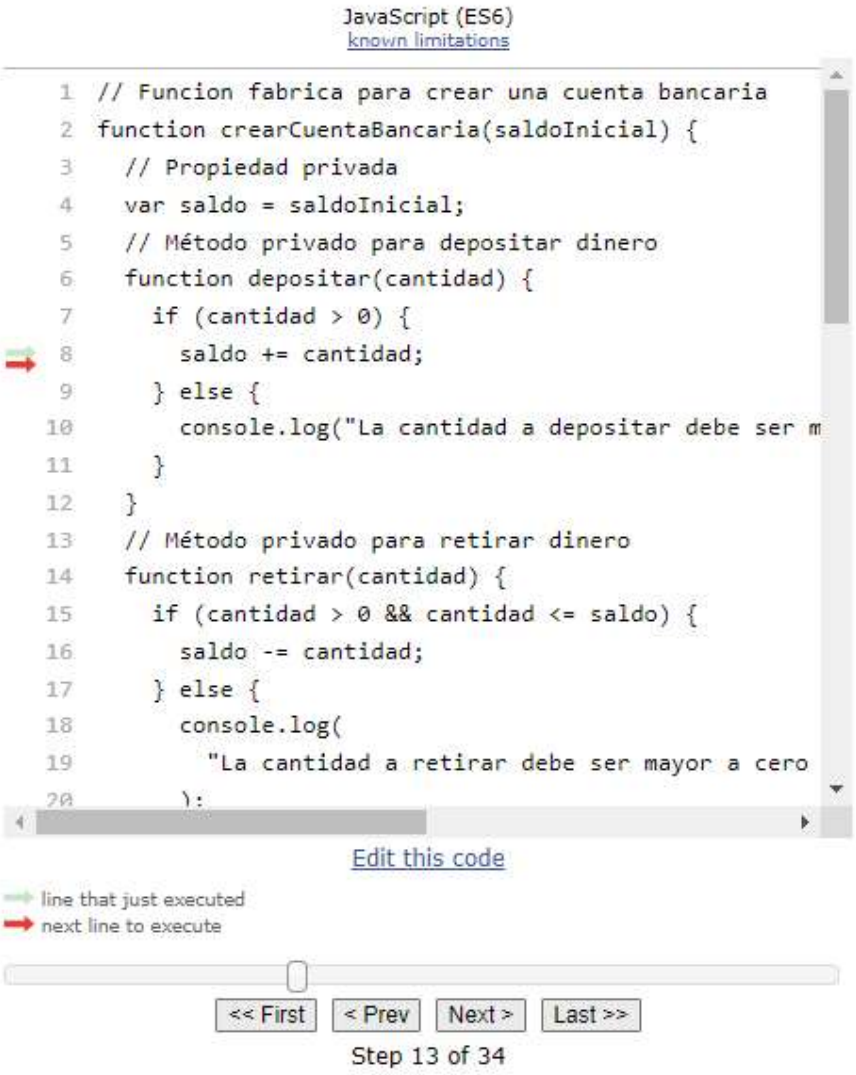
Paso 11 de 34: evalúa la condición



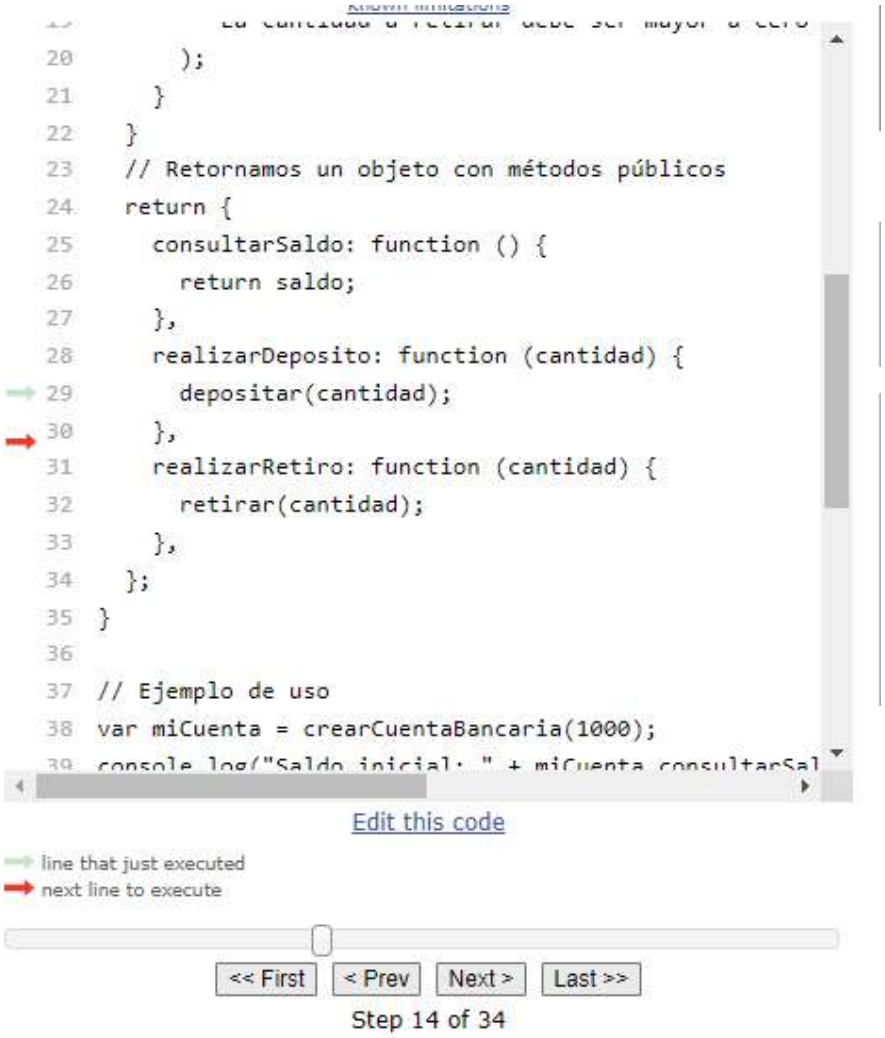
Paso 12 de 34: evalúa la condición +=



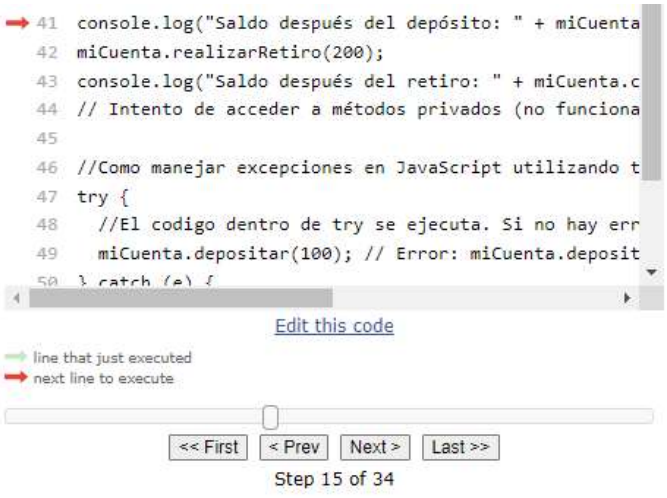
Paso 13 de 34: evalúa la condición else



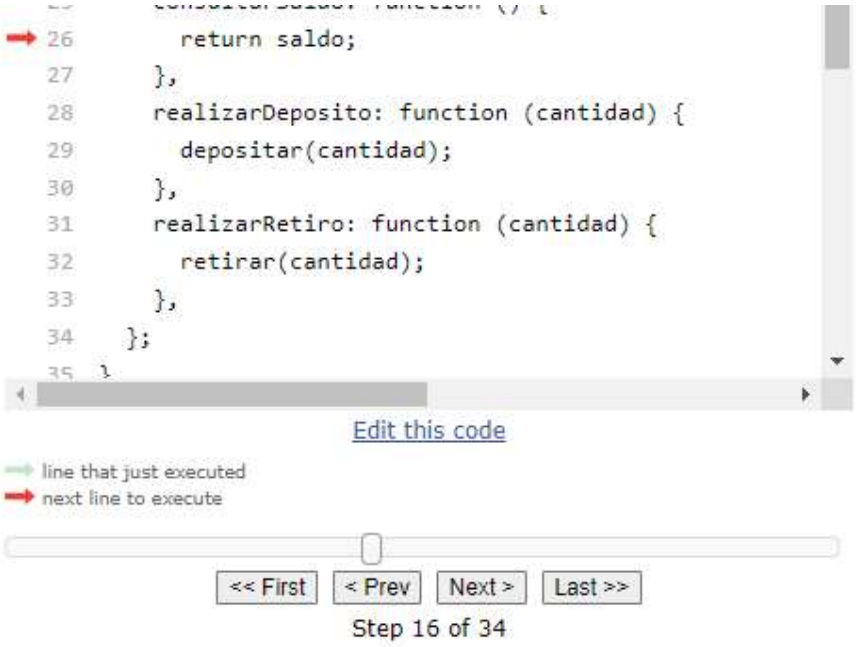
Paso 14 de 34: se ejecuta la funcion realizarretiro



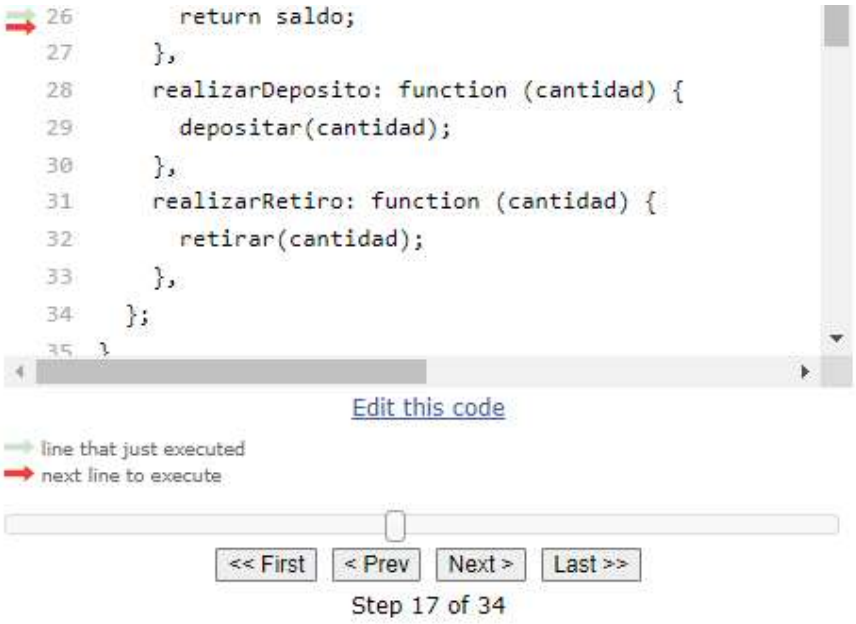
Paso 15 de 34: se muestra el mensaje de la sumatoria



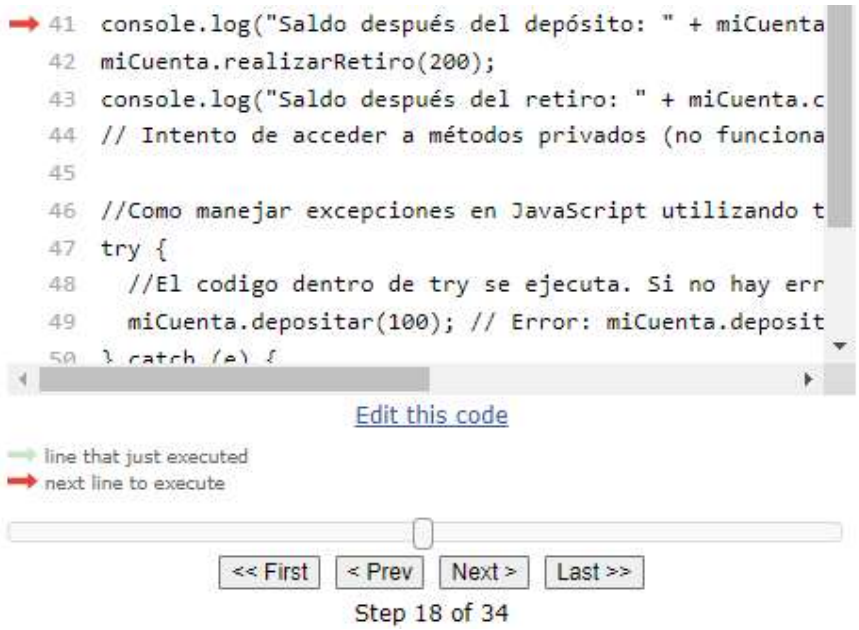
Paso 16 de 34: se retorna el saldo



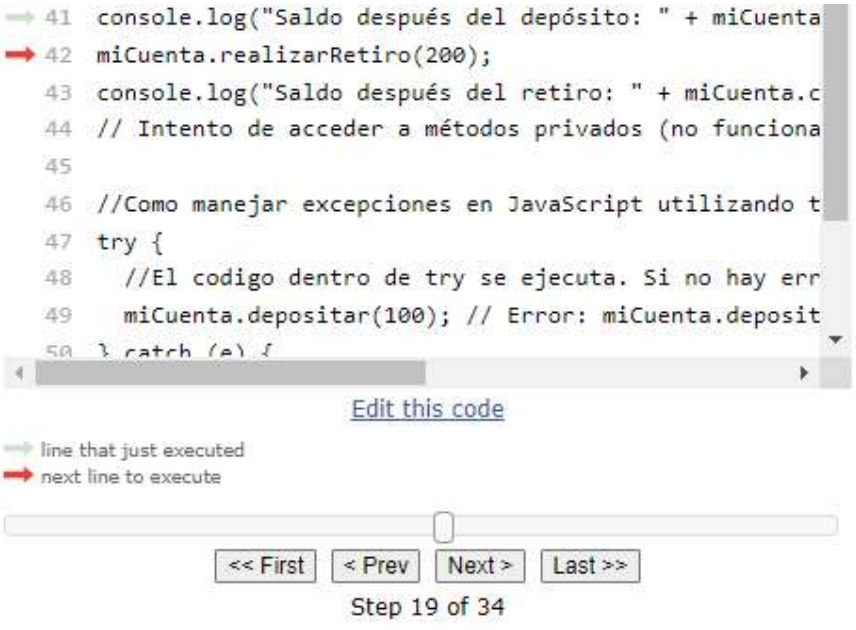
Paso 17 de 34: se retorna el saldo



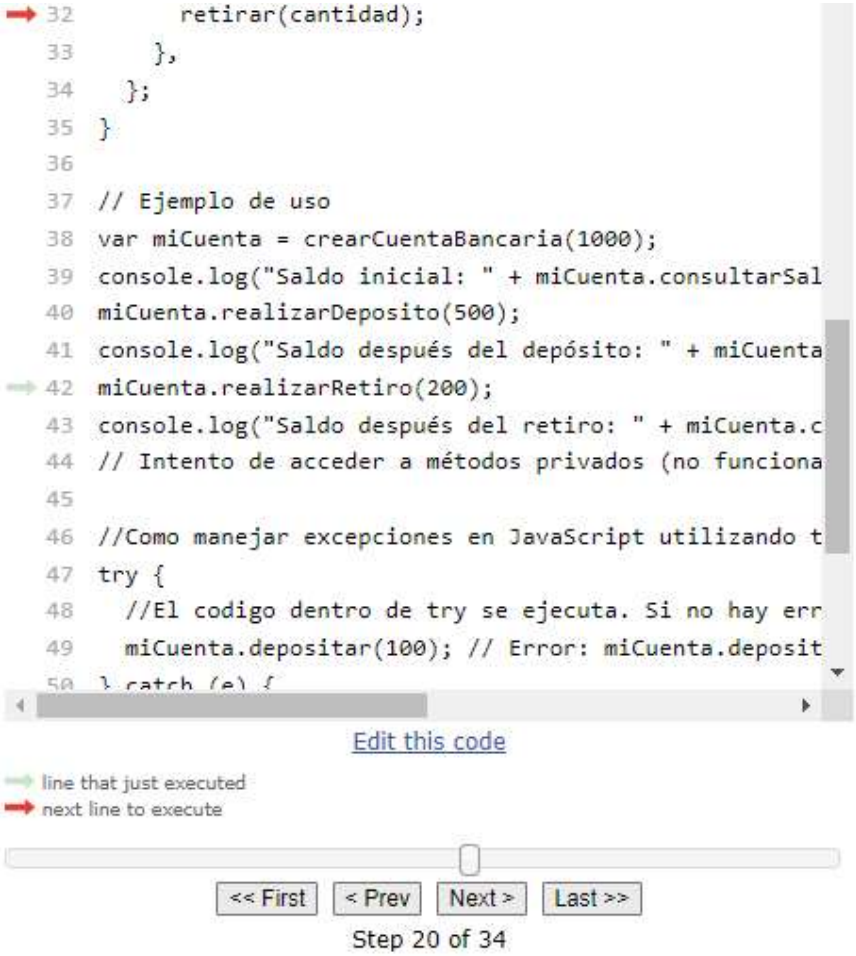
Paso 18 de 34: se muestra el mensaje con la sumatoria tomando en cuenta el retiro



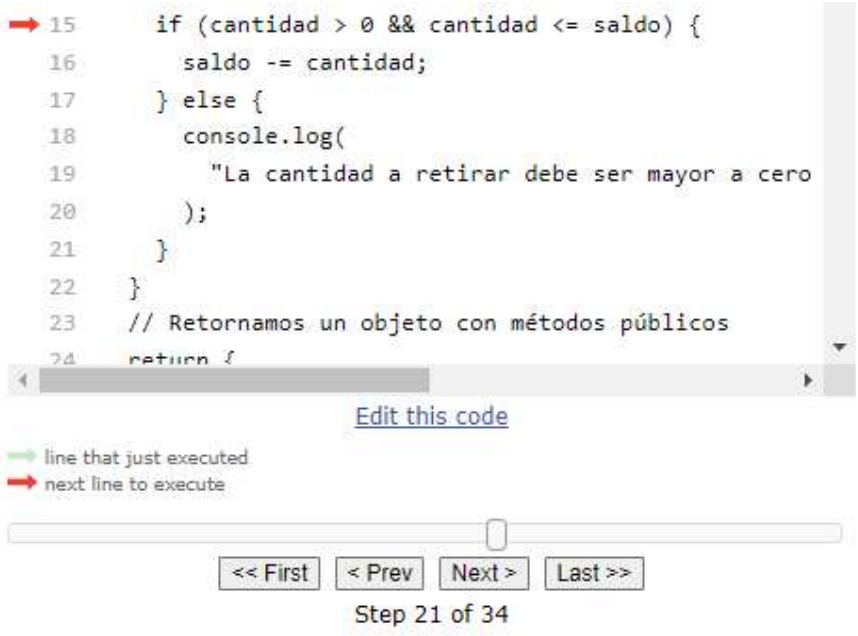
Paso 19 de 34: se ejecuta el retiro



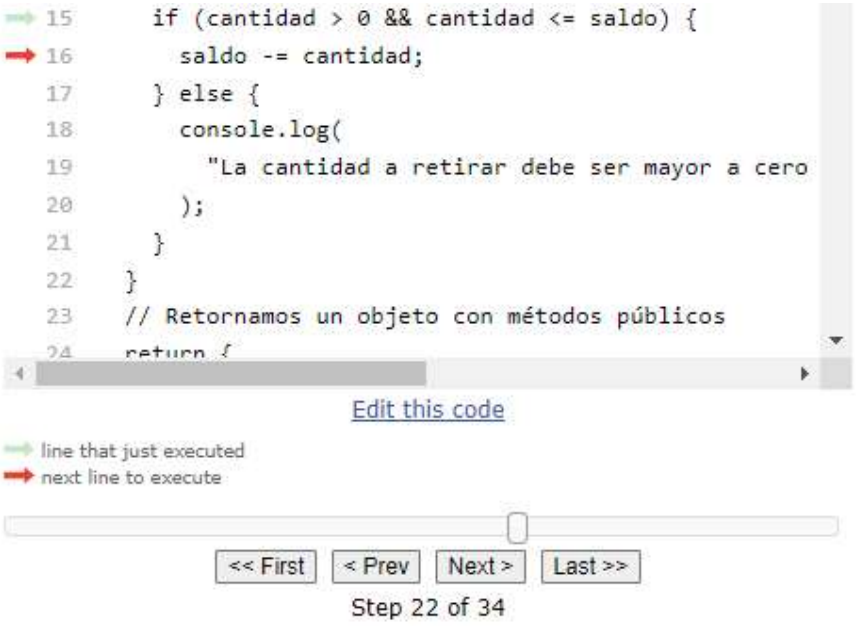
Paso 20 de 34: se llama la funcion de retirar



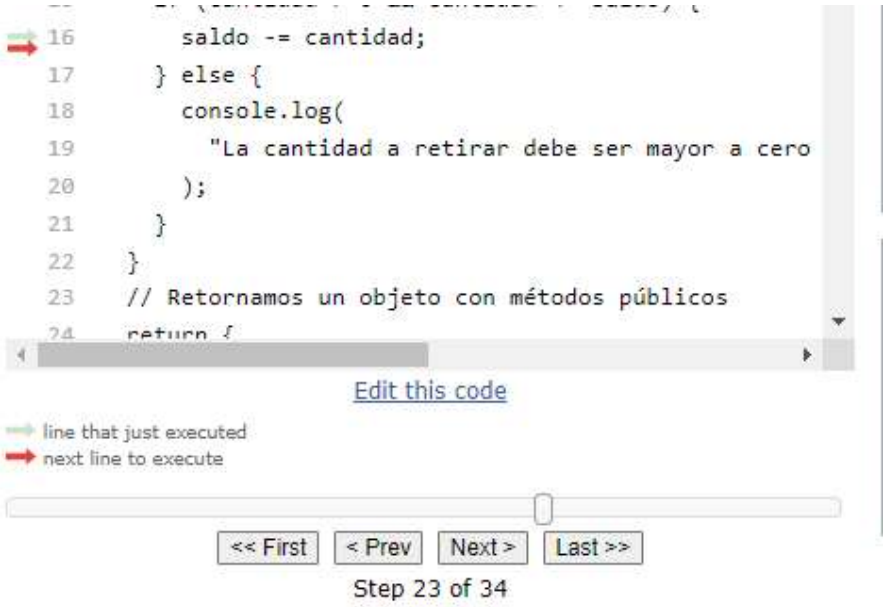
Paso 21 de 34: se evalua la condicion



Paso 22 de 34: se evalua la condición



Paso 23 de 34: se evalua la condición



Paso 24 de 34:

```
31      realizarRetiro: function (cantidad) {
32        retirar(cantidad);
33      },
34    };
35  }
36
37  // Ejemplo de uso
38  var miCuenta = crearCuentaBancaria(1000);
39  console.log("Saldo inicial: " + miCuenta.consultarSal
40  miCuenta.realizarDeposito(500);
41  console.log("Saldo después del depósito: " + miCuenta
42  miCuenta.realizarRetiro(200);
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 24 of 34

Paso 25 de 34 Se ejecuta el mensaje en consola

```
43  console.log("Saldo después del retiro: " + miCuenta.c
44  // Intento de acceder a métodos privados (no funciona
45
46  //Como manejar excepciones en JavaScript utilizando t
47  try {
48    //El codigo dentro de try se ejecuta. Si no hay err
49    miCuenta.depositar(100); // Error: miCuenta.deposit
50  } catch (e) {
51    //el parámetro e es una referencia al objeto de exc
52    console.log(e.message); //message es la propiedad d
```

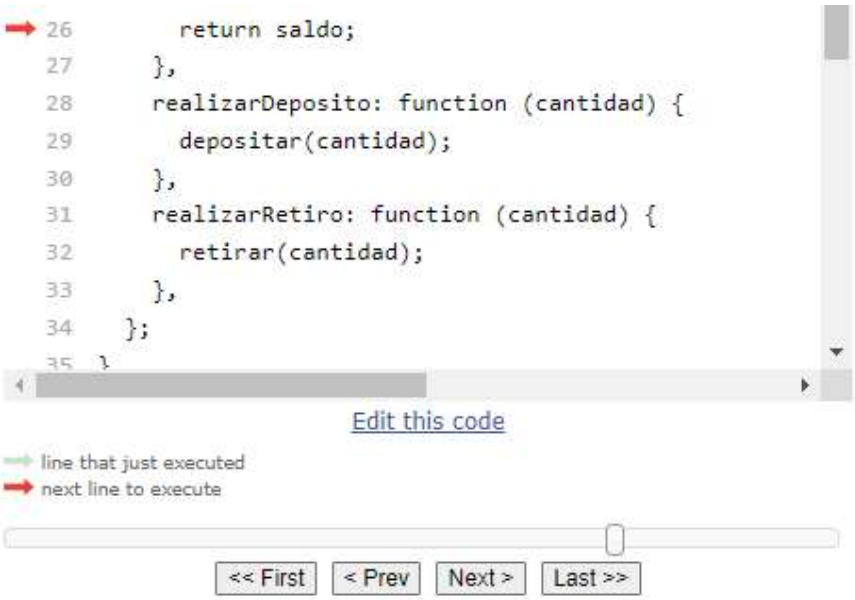
Edit this code

line that just executed
next line to execute

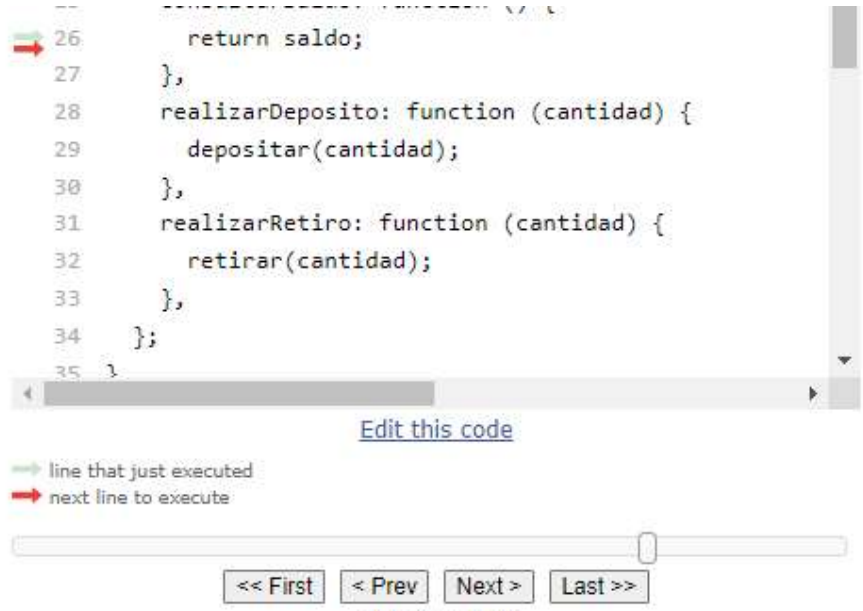
<< First | < Prev | Next > | Last >>

Step 25 of 34

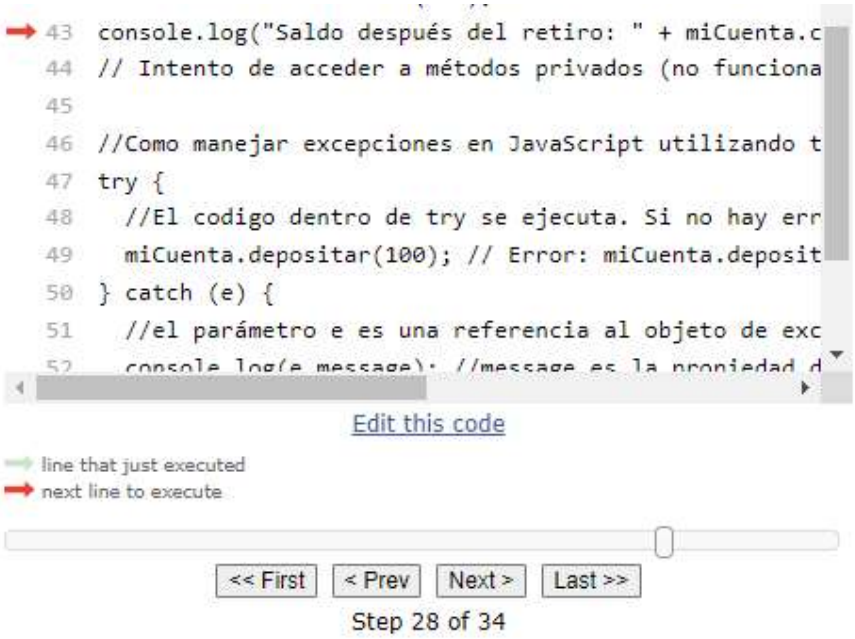
Paso 26 de 34 Se retorana saldo



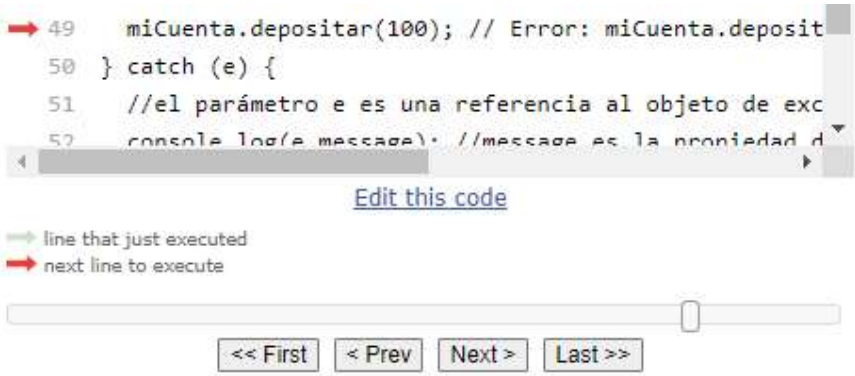
Paso 27 de 34 Se retorana saldo



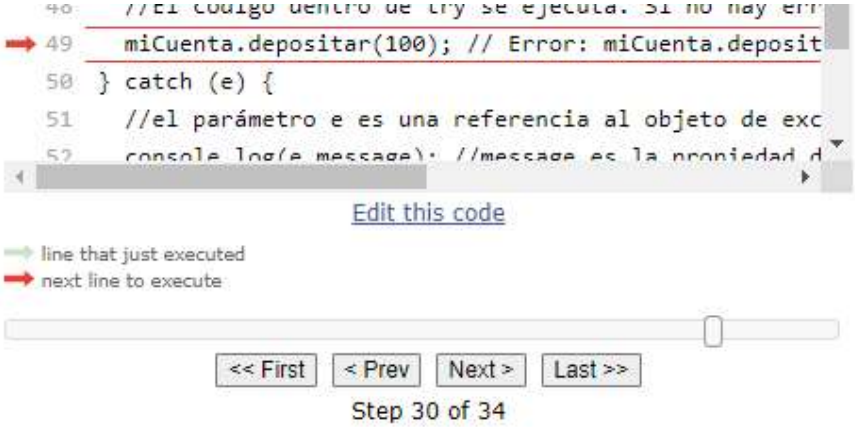
Paso 28 de 34 Se imprime el saldo después del retiro.



Paso 29 de 34 Se ejecuta la funcion depositar



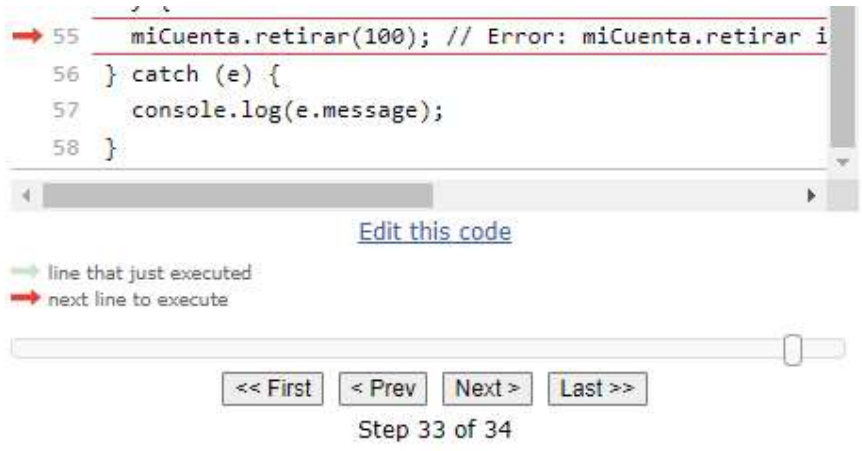
Paso 30 de 34 Se ejecuta la funcion depositar



Paso 31 de 34 Se muestra el mensaje

Paso 32 de 34 Se muestra el mensaje

Paso 33 de 34 Se muestra el mensaje



Paso 34 de 34 codigo en consola