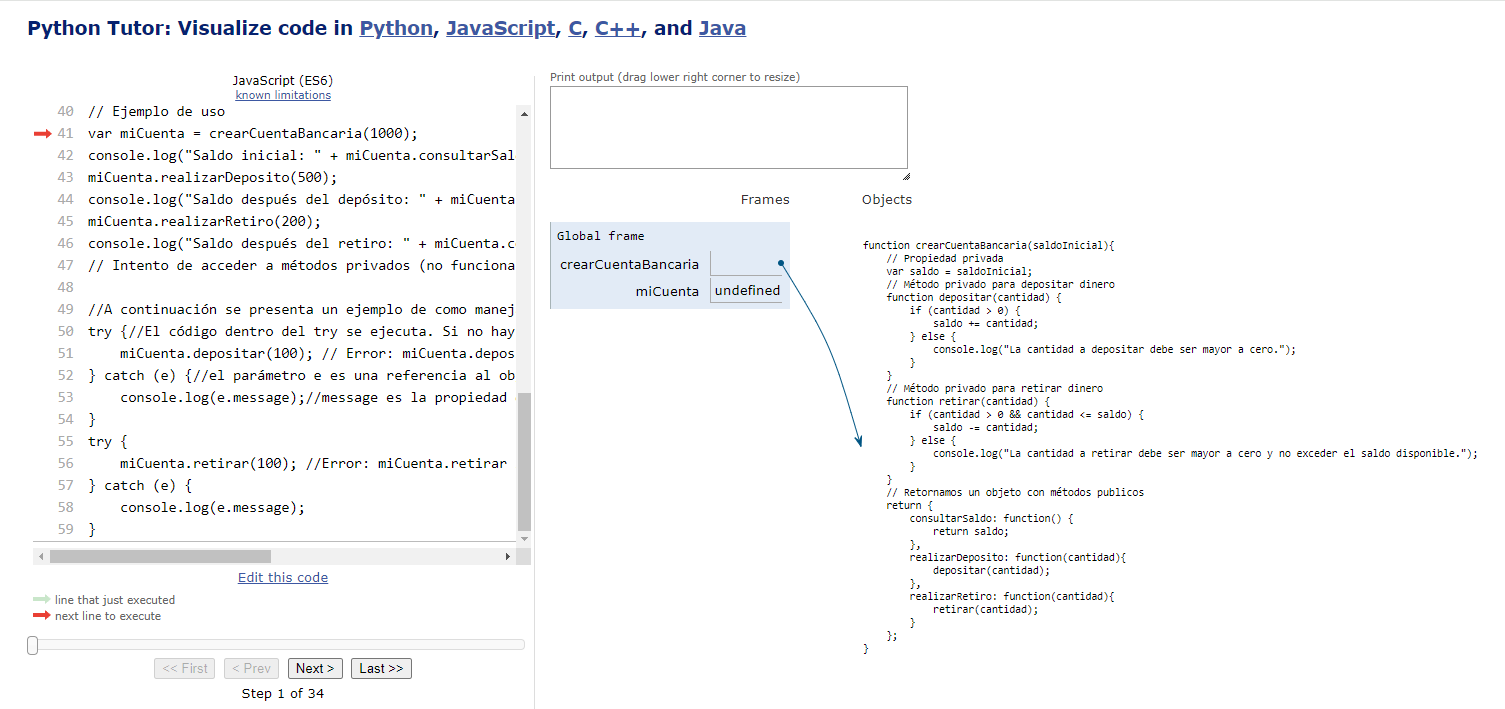
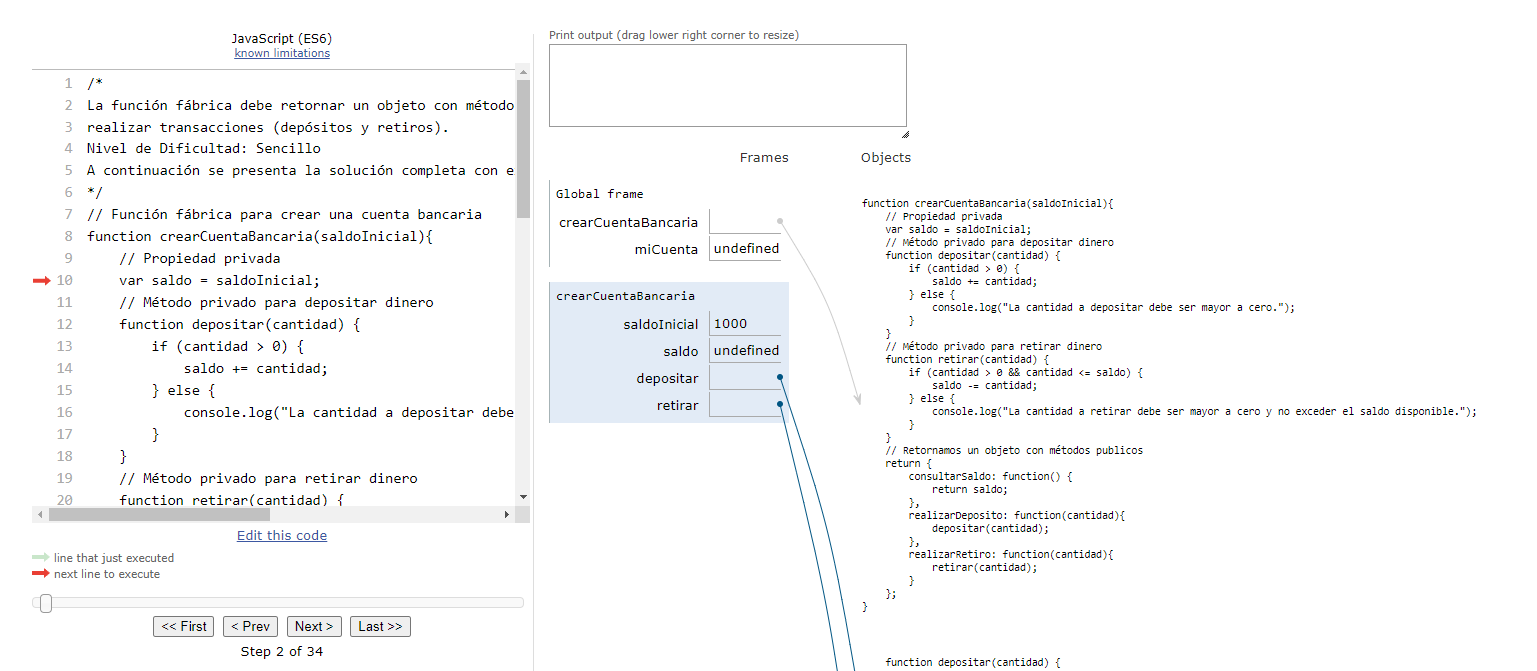
**Paso 1:**



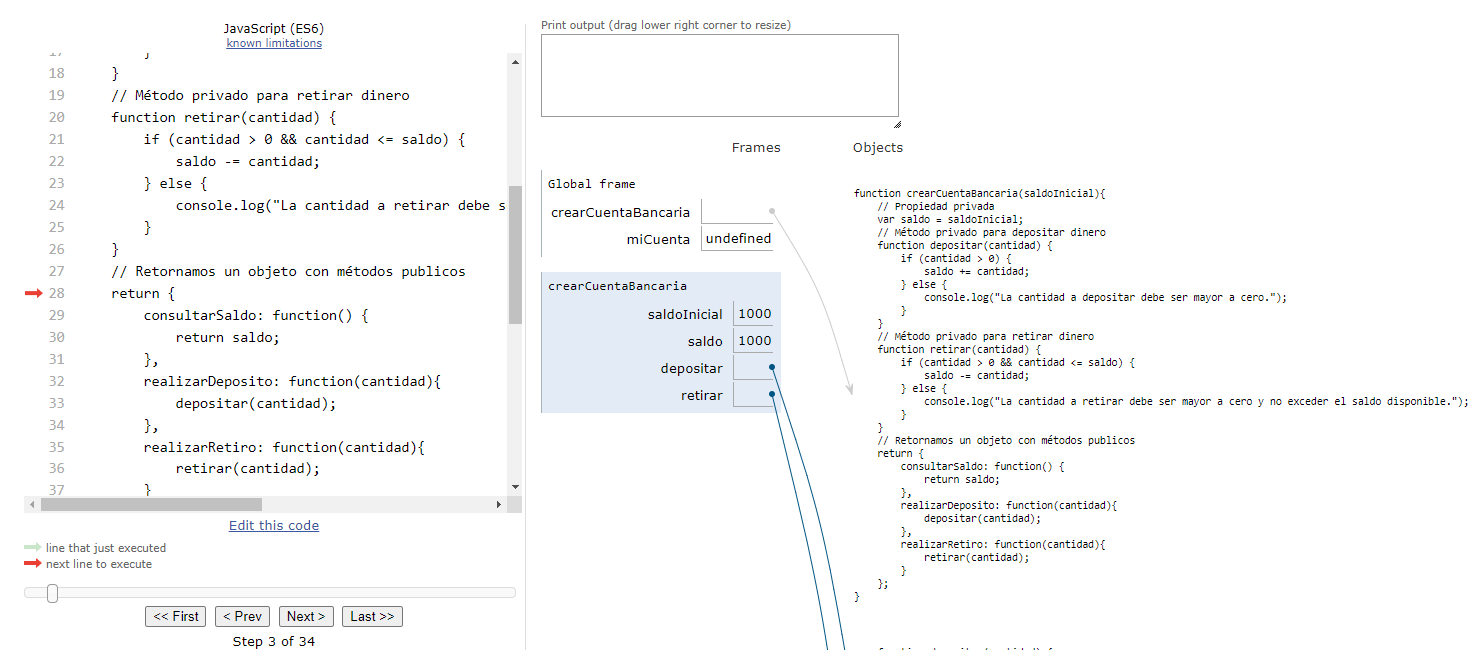
Se crea el objeto “miCuenta” basado en la función crearCuentaBancaria. Hasta el momento, como el objeto no había sido creado, en el Global Frame miCuenta aún no estaba definida.

**Paso 2:**



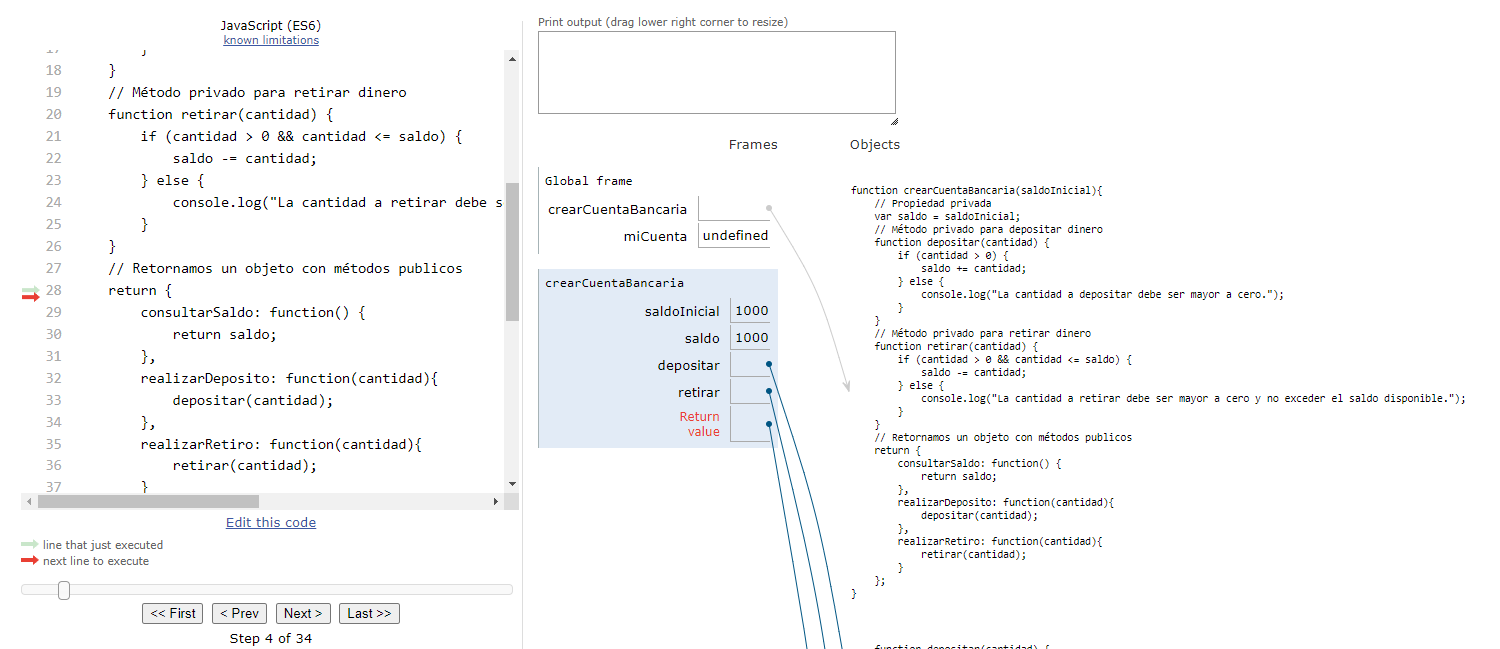
Se le asigna el valor de saldoInicial a el objeto “miCuenta”. Por el momento, sus demás propiedades aún no están definidas.

**Paso 3:**



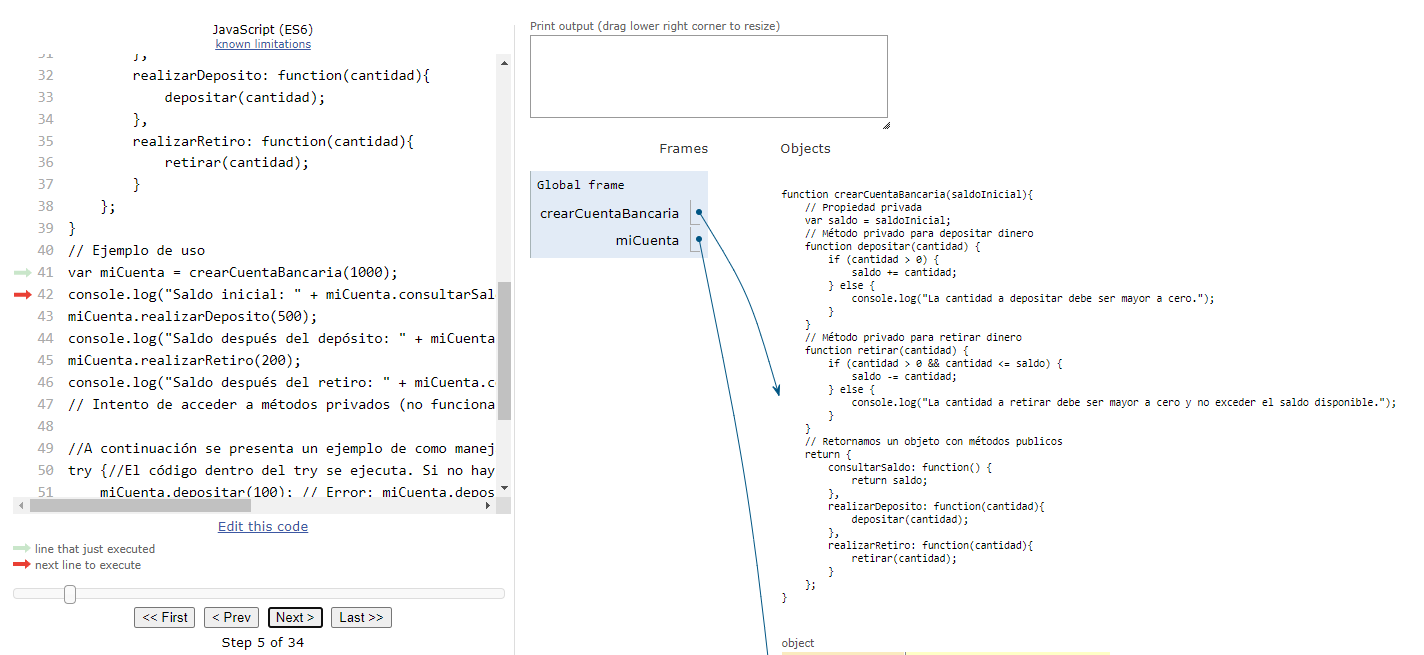
Se crea la función return para el objeto “miCuenta”.

**Paso 4:**



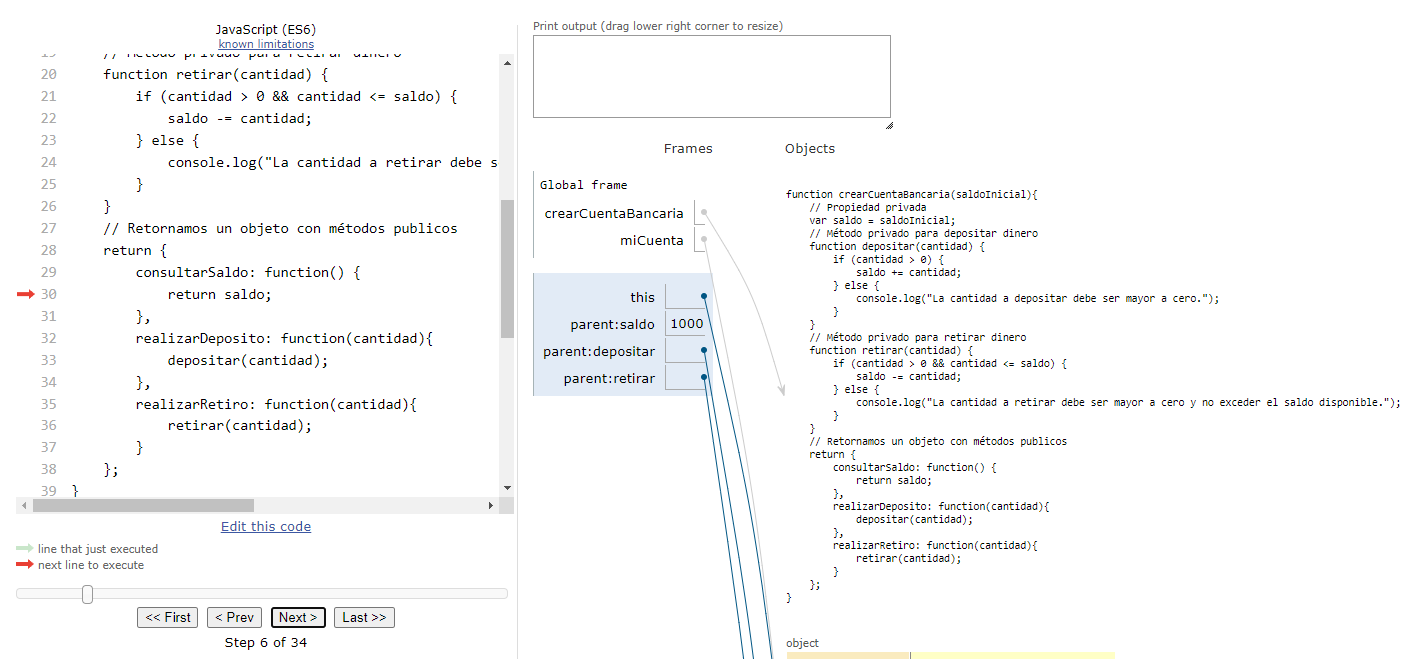
Se reserva espacio para las funciones correspondientes a los valores de return del objeto “miCuenta”.

**Paso 5:**



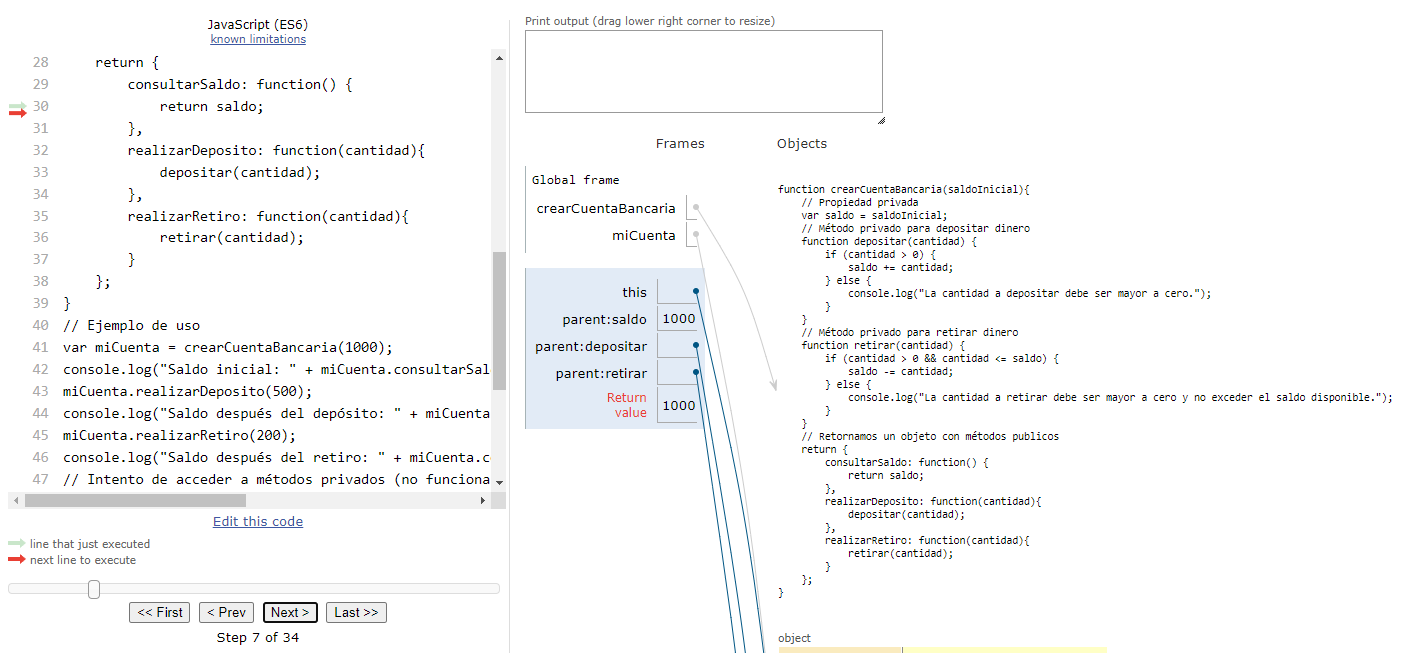
Se reserva espacio para las funciones correspondientes a los valores de return del objeto “miCuenta”.

**Paso 6:**



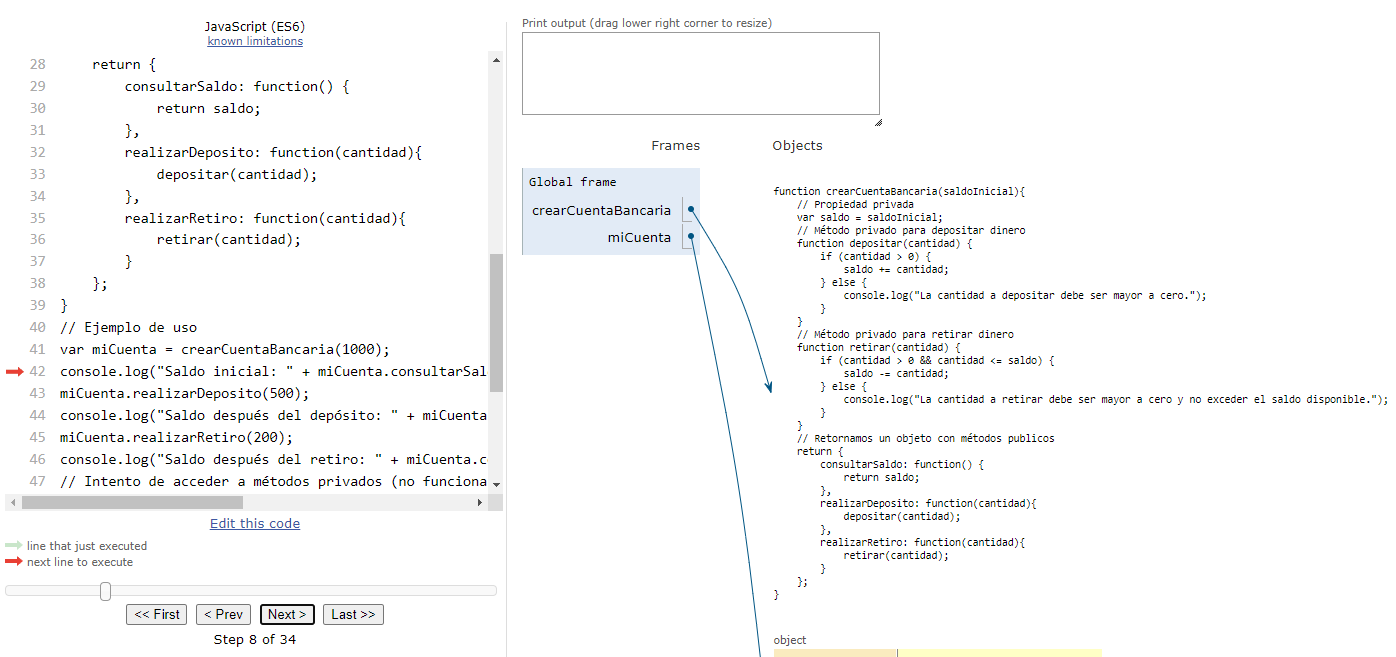
“miCuenta” hereda de la función constructora las funciones depositar y retirar.

**Paso 7:**



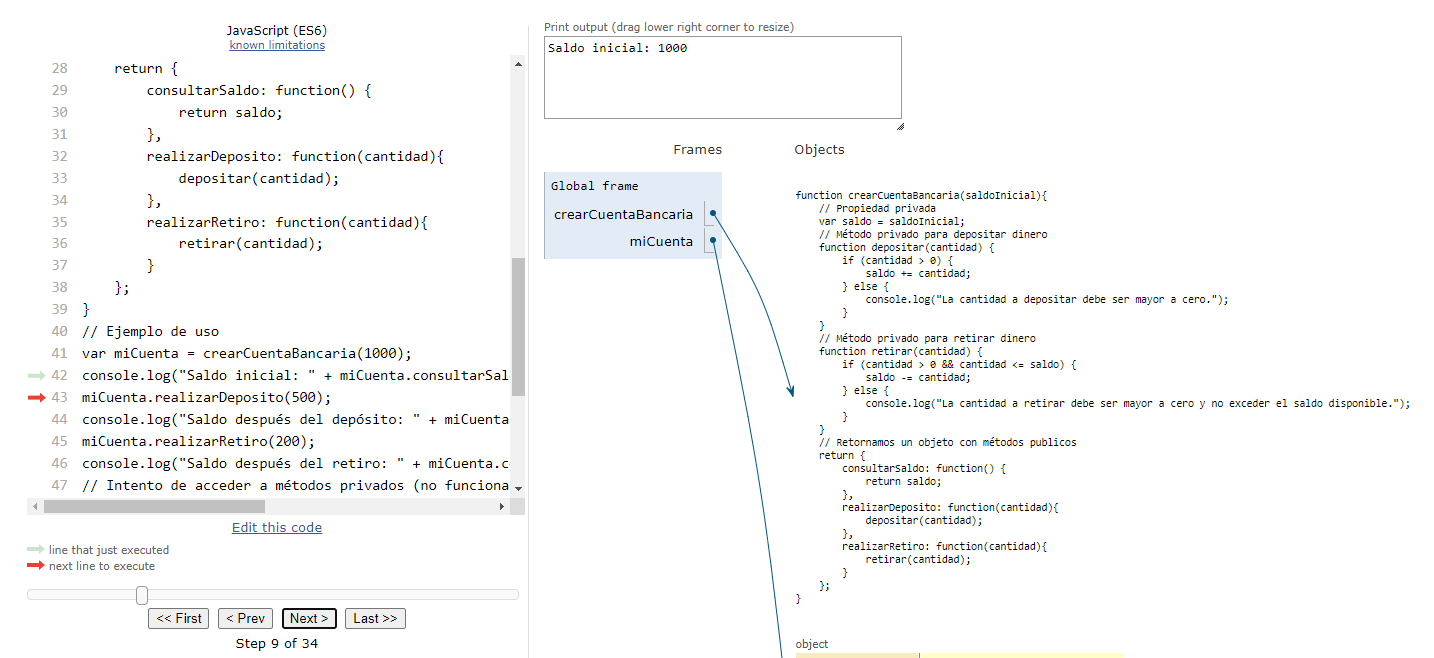
Se asigna el valor 1000 a la función return del objeto “miCuenta”.

**Paso 8:**



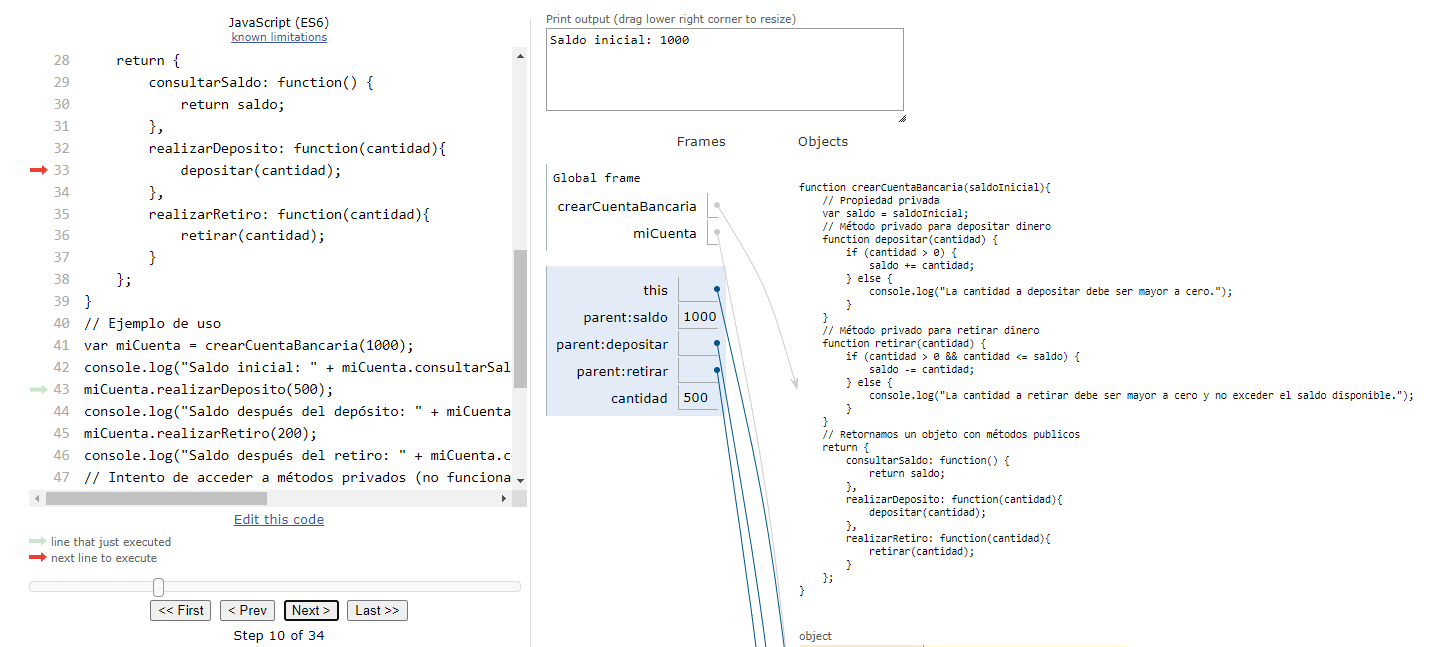
Se hace el display a pantalla con console.log.

**Paso 9:**



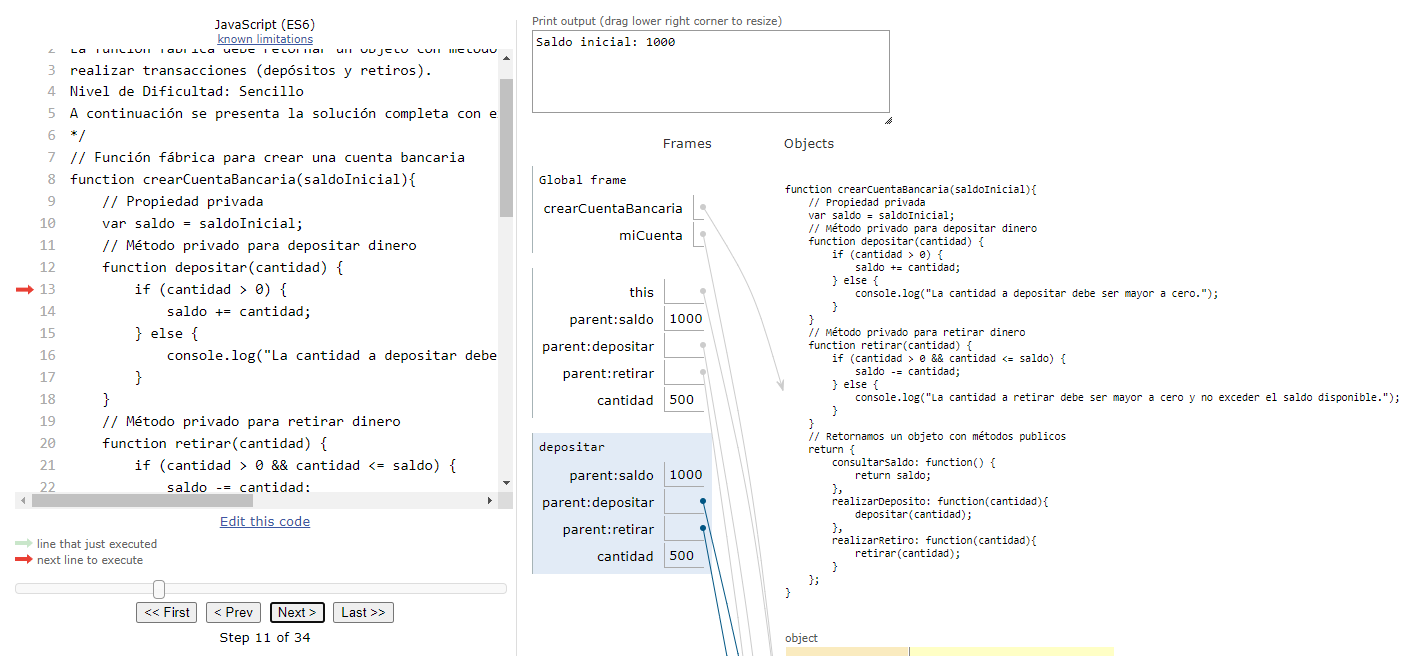
Se realiza la función .realizarDeposito con el valor de 500 al objeto “miCuenta”.

**Paso 10:**



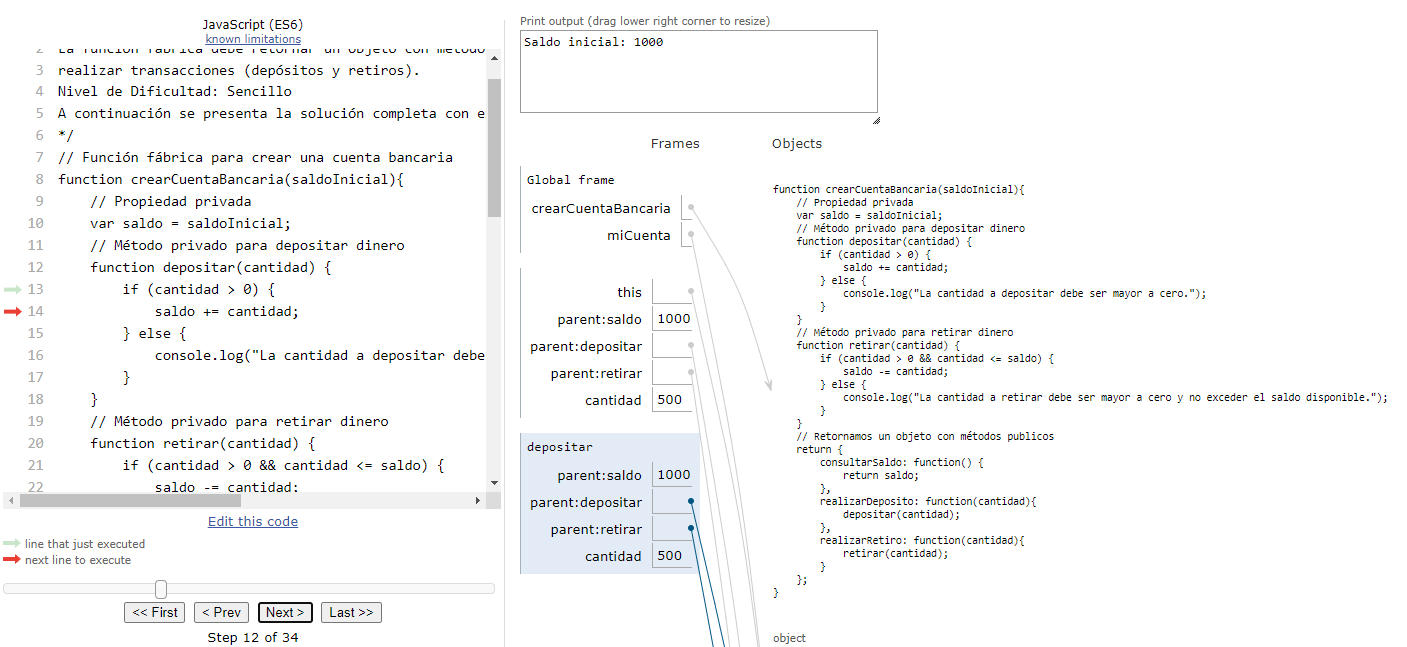
El programa entiende que la cantidad que necesita operar son 500, en este caso los agregará al saldo.

**Paso 11:**



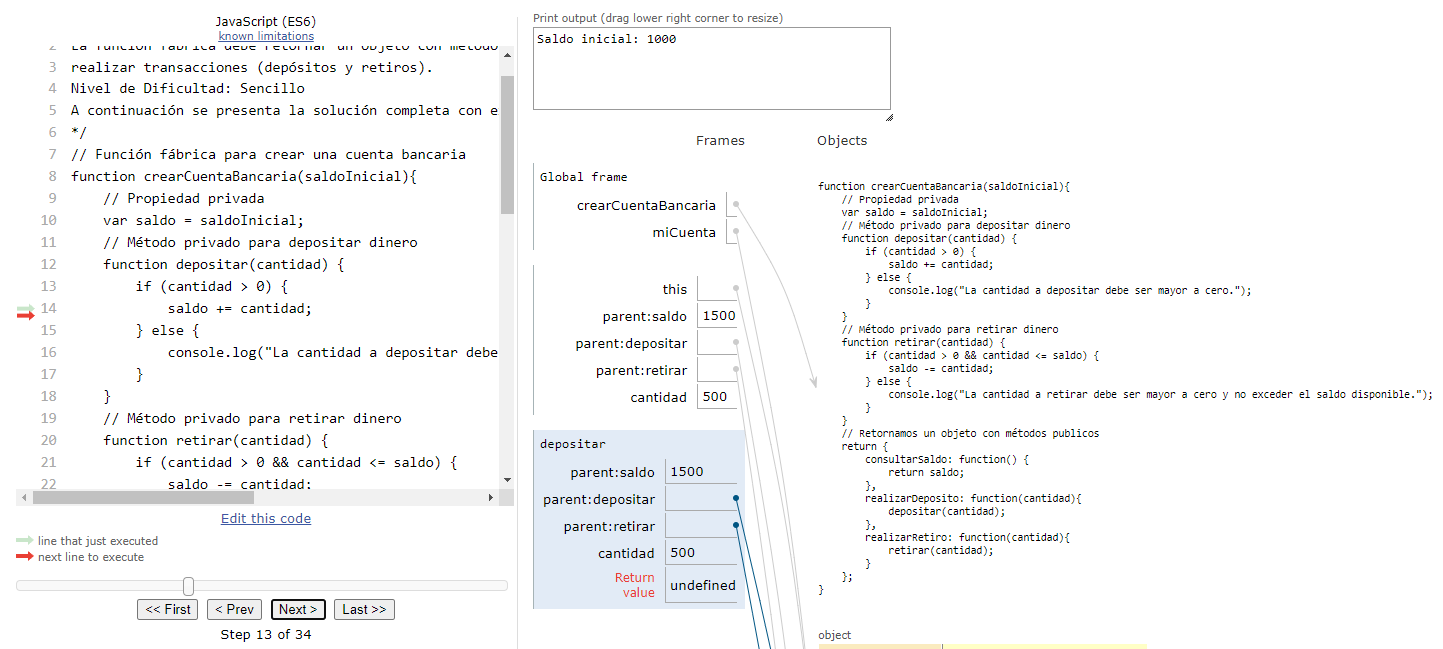
El programa va a revisar la función depositar.

**Paso 12:**



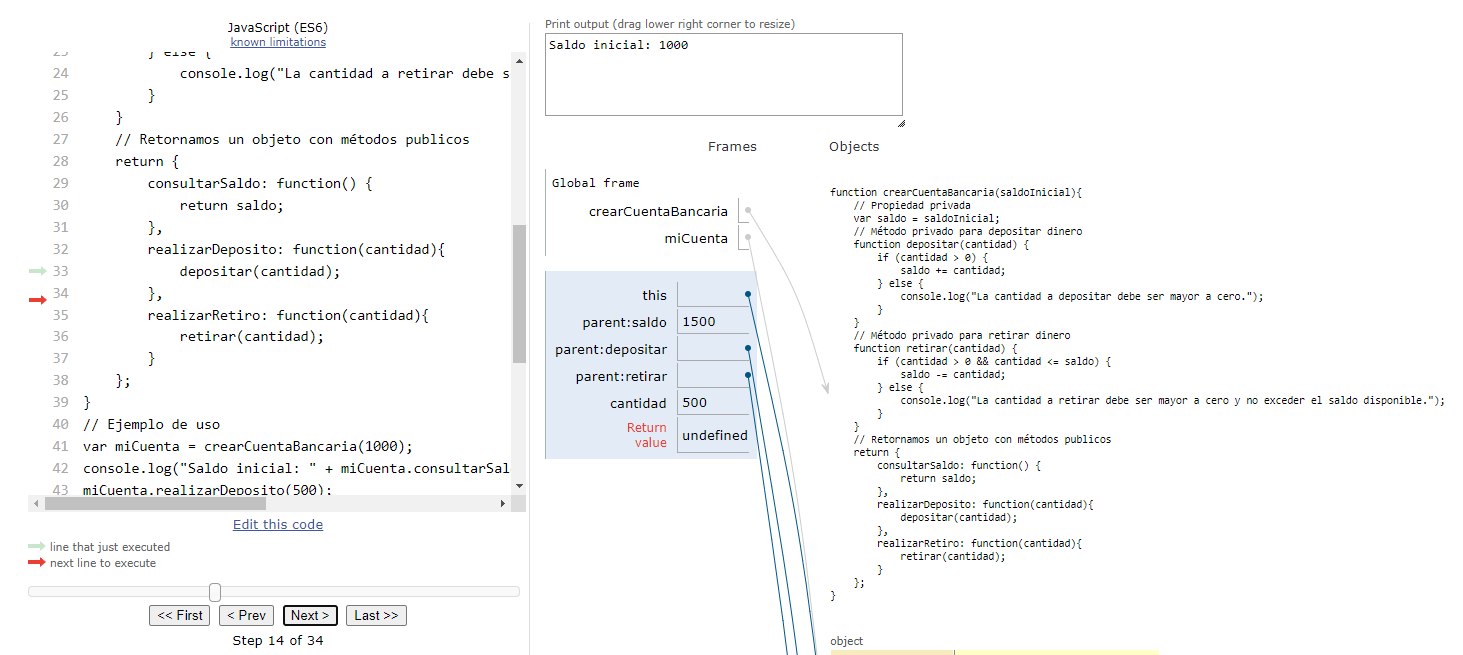
El programa encuentra y entiende que lo que tiene que realizar es una suma.

**Paso 13:**



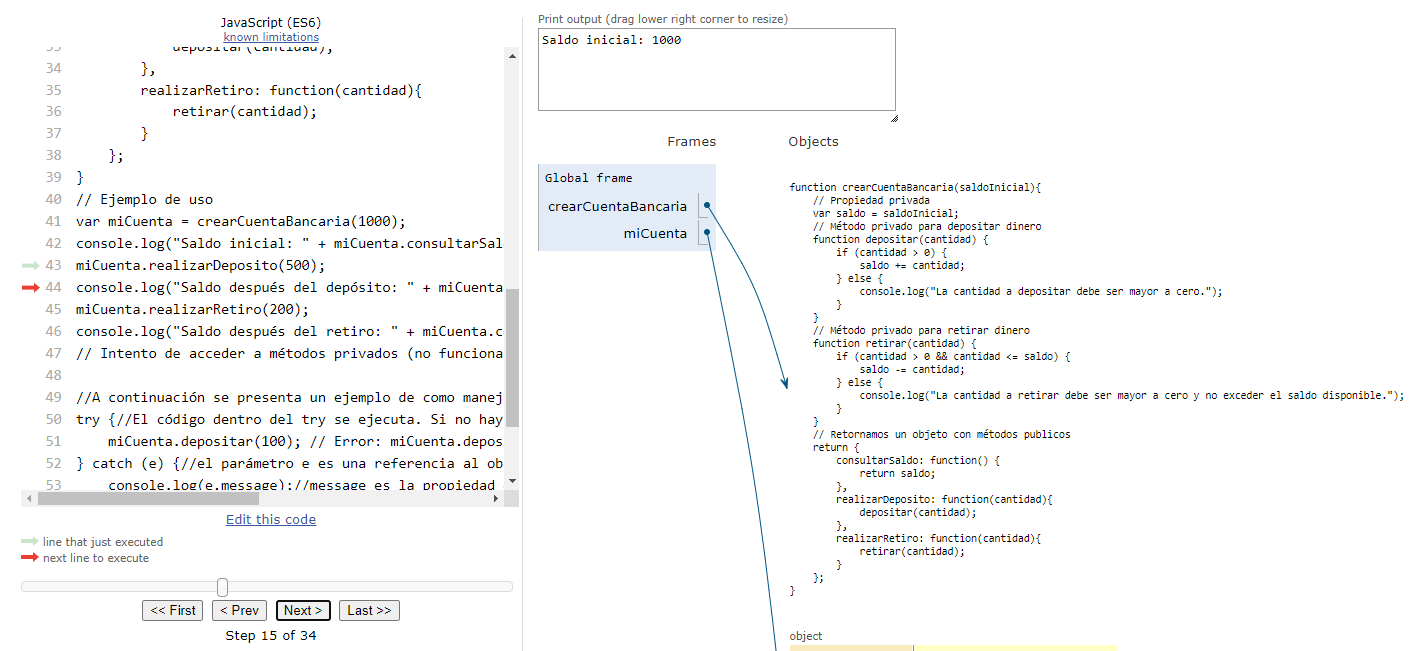
El programa reserva el espacio de memoria para el return como aún no ha realizado la operación este es undefined por el momento.

**Paso 14:**



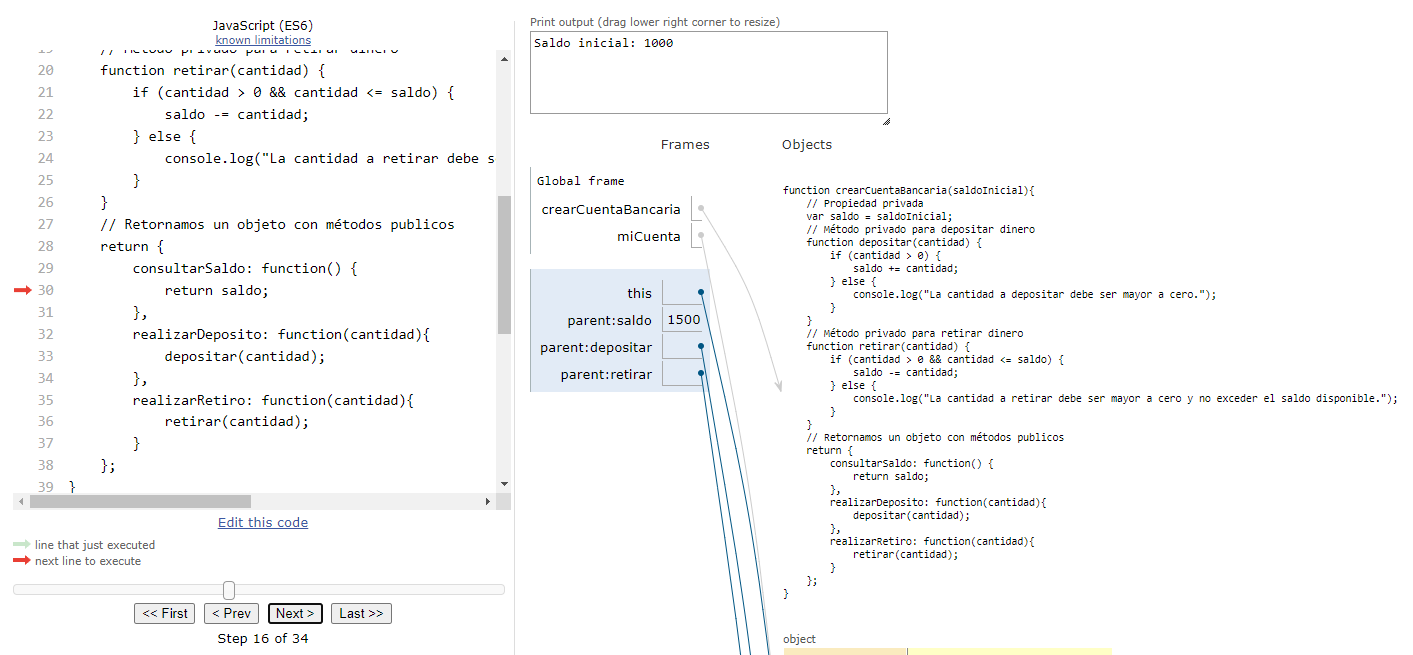
El programa revisa si hay alguna otra función que incorporar a los cálculos.

**Paso 15:**



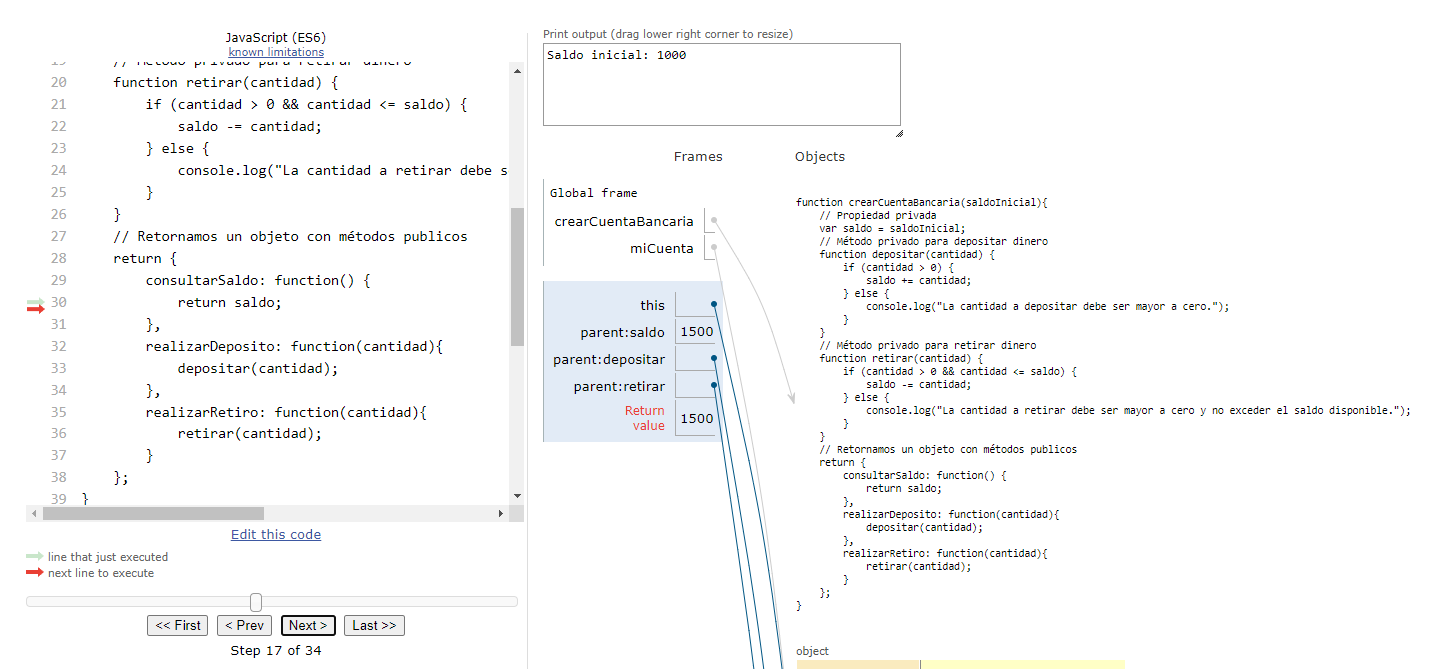
El programa operó la función depósito y saltó a la siguiente línea.

**Paso 16:**



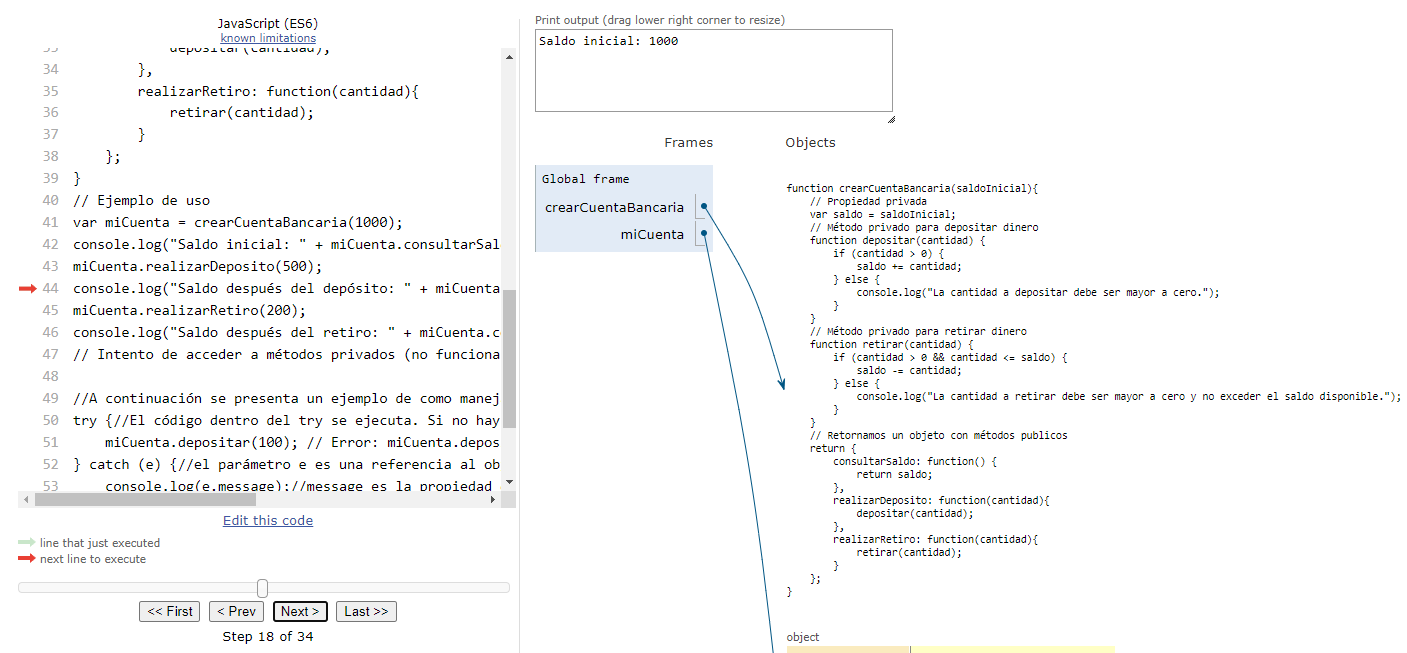
El programa verifica la función consultarSaldo.

**Paso 17:**



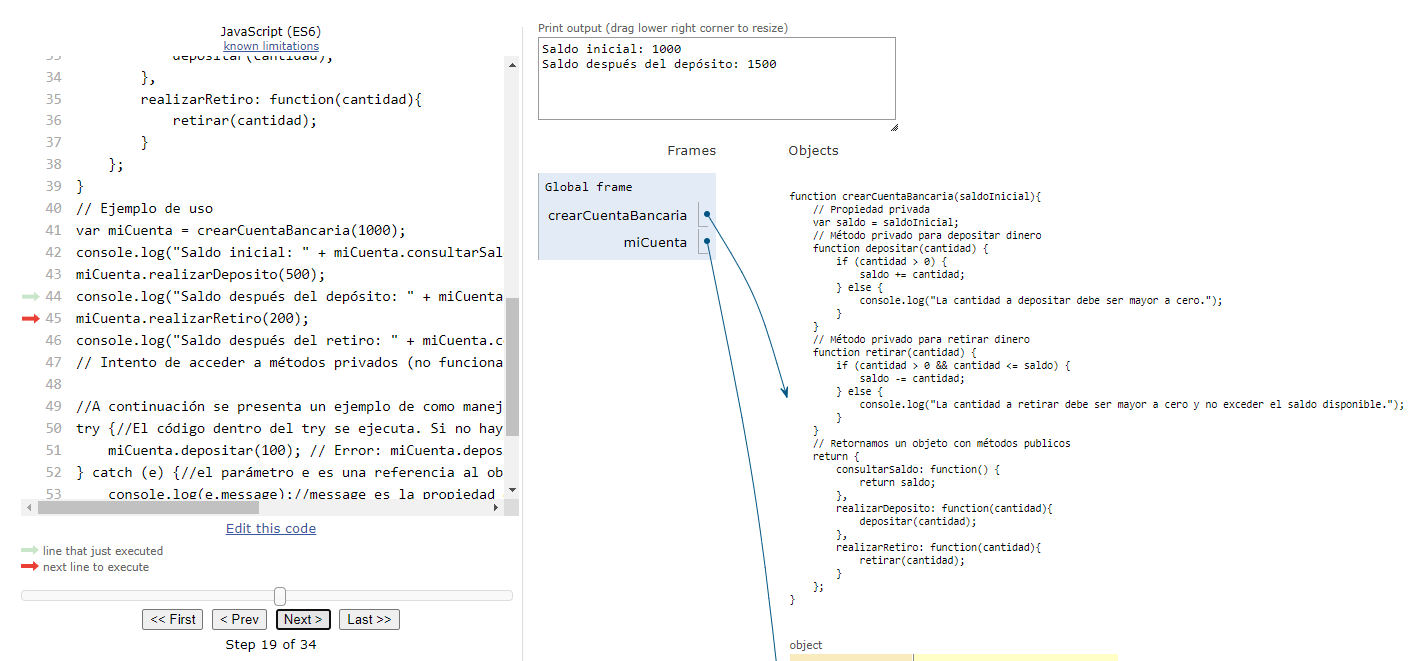
El programa extrae el return value de saldo, actualmente 1500.

**Paso 18:**



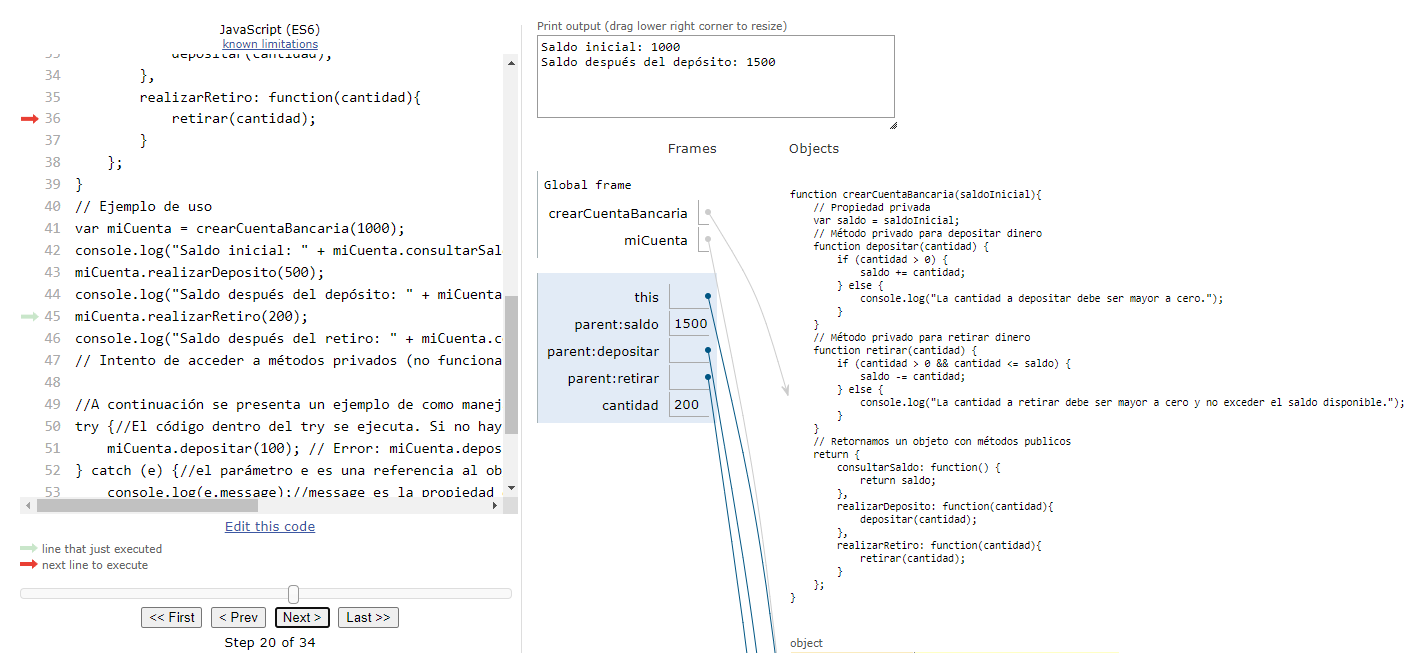
Despliega la información en consola.

**Paso 19:**



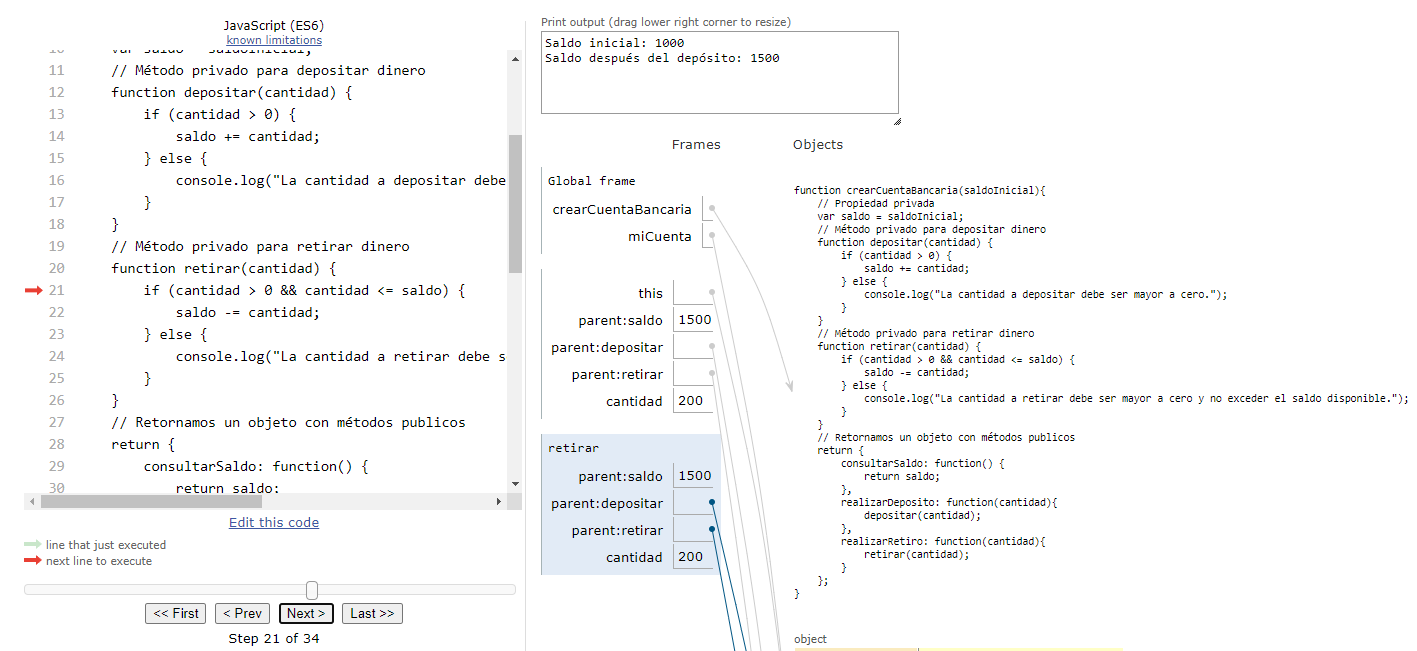
El programa salta a la siguiente línea.

**Paso 20:**



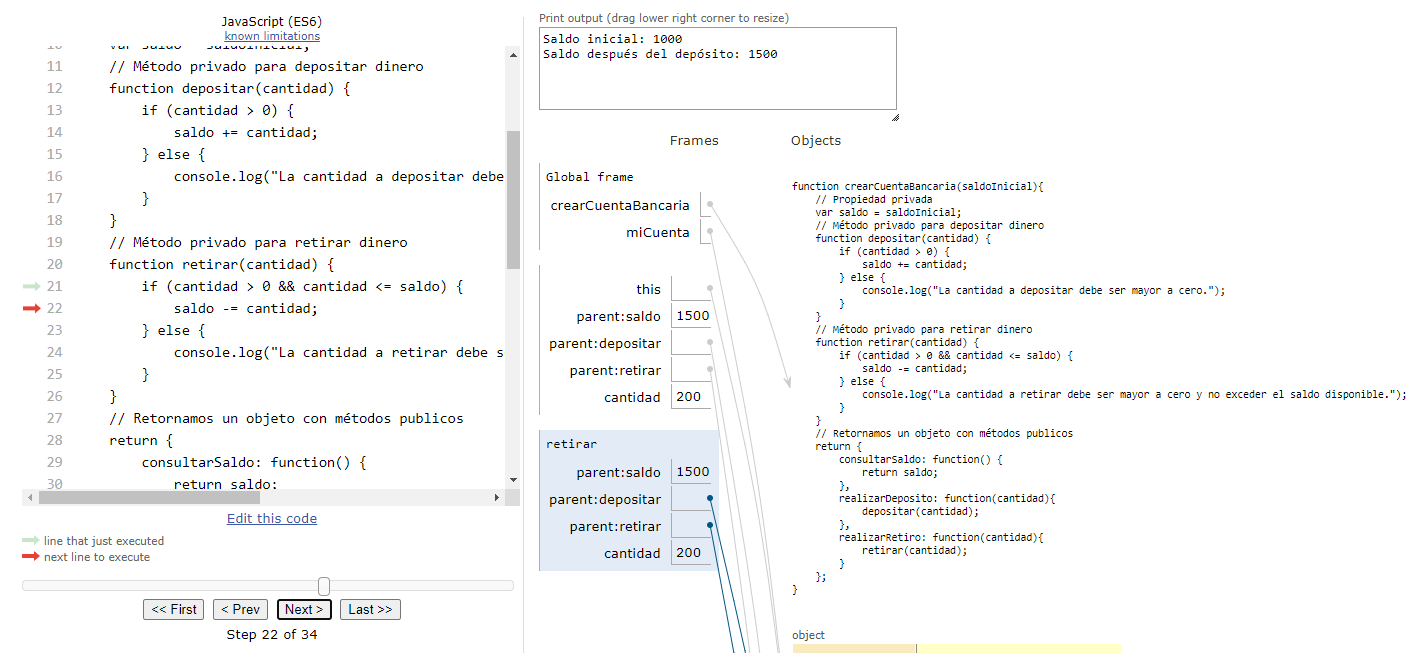
Como ahora se va a operar un retiro, el programa obtiene su saldo y la cantidad a operar.

**Paso 21:**



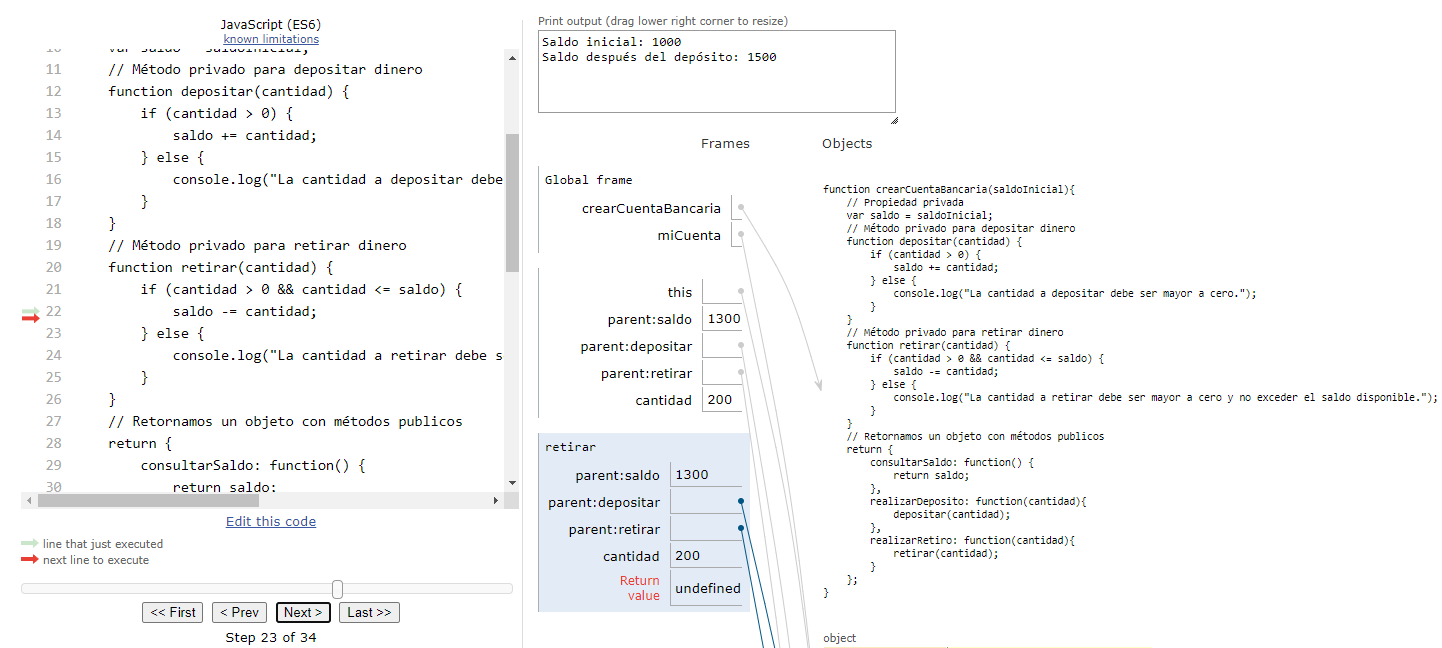
El programa revisa la función y valida condiciones.

**Paso 22:**



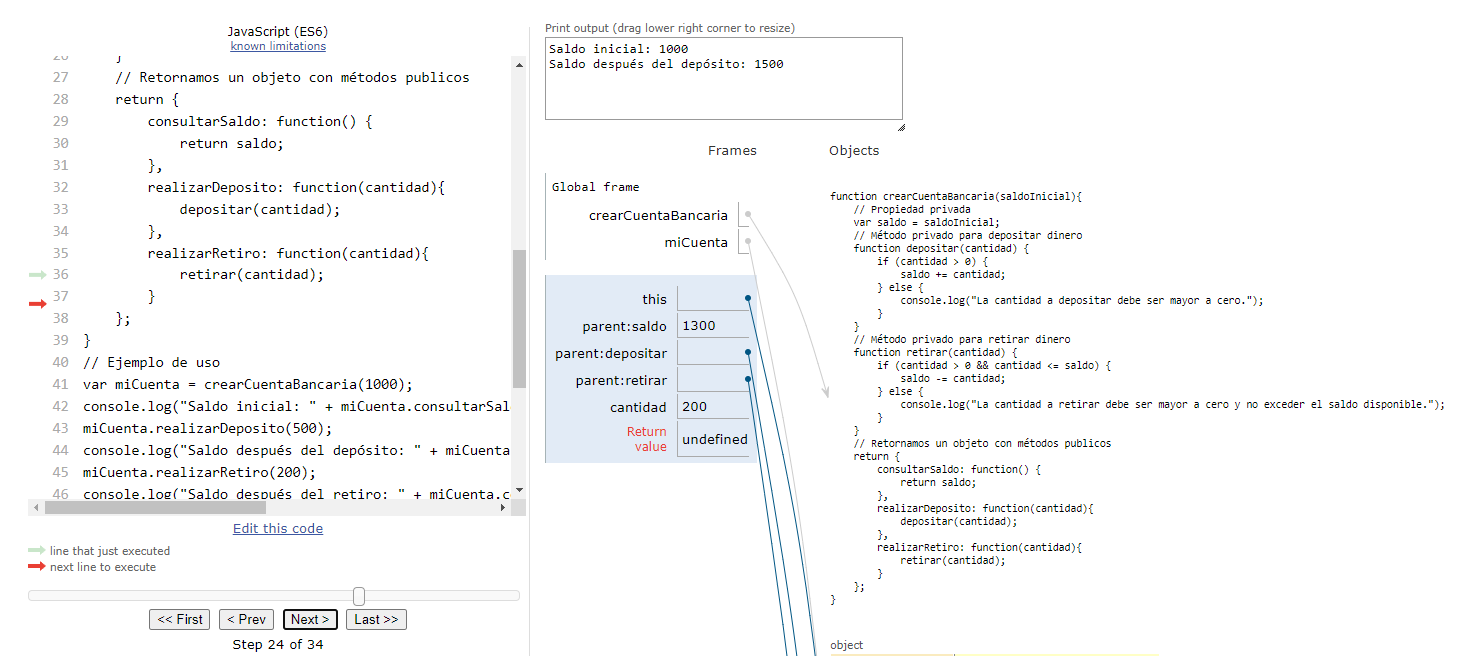
Como las condiciones evaluadas si cumplen, pasa a revisar la operación que debe realizar.

**Paso 23:**



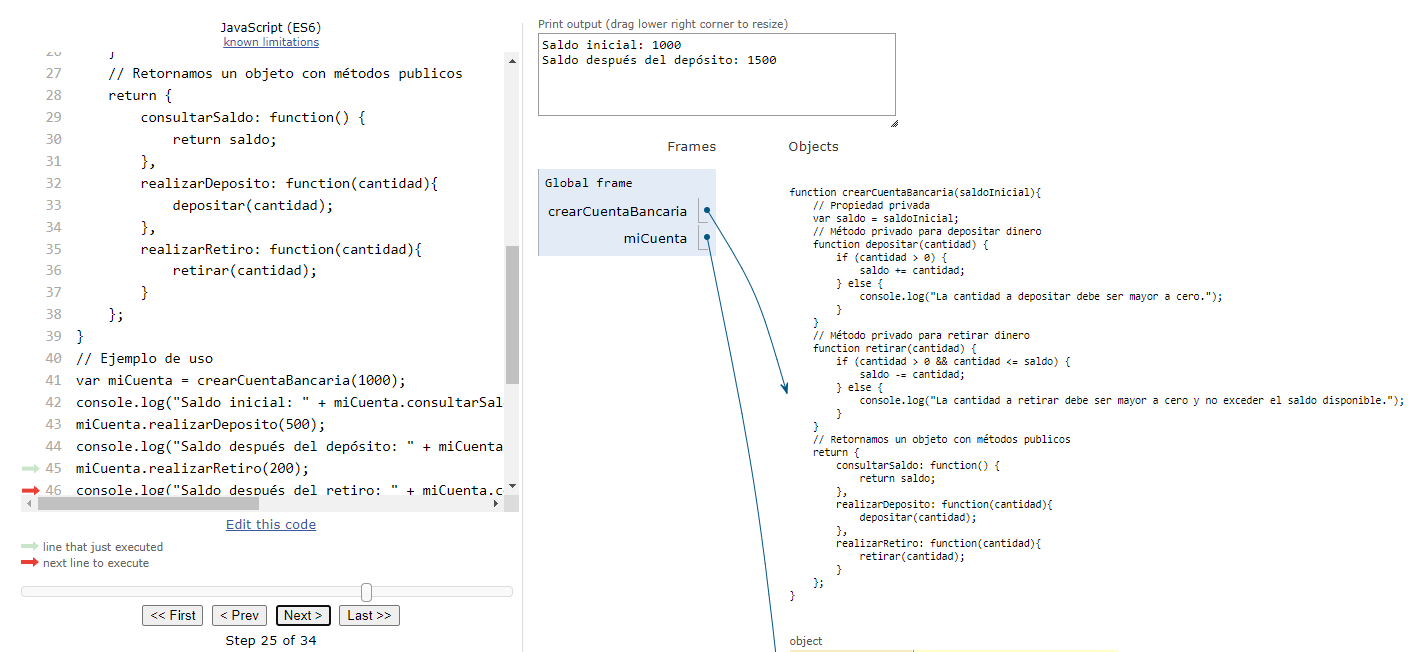
Reserva espacio en memoria para el return ya que aún no se ha realizado la operación.

**Paso 24:**



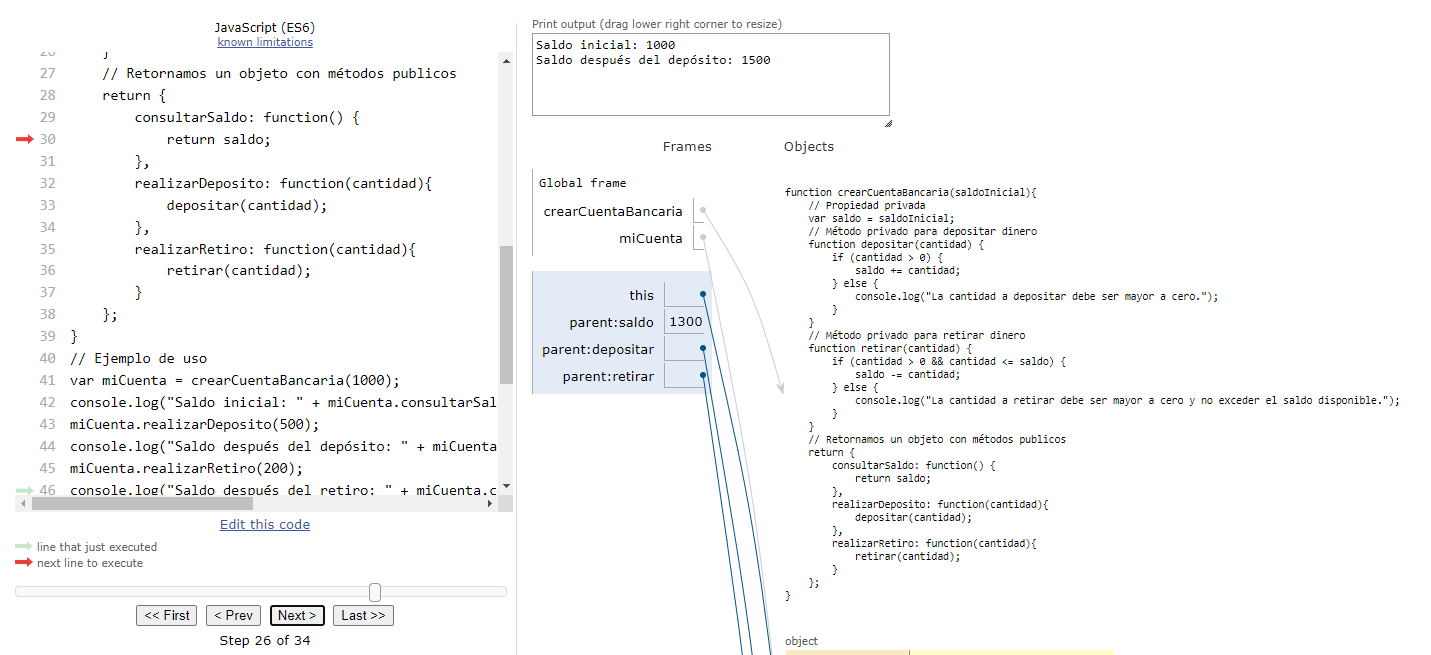
Termina de revisar funciones y ejecuta.

**Paso 25:**



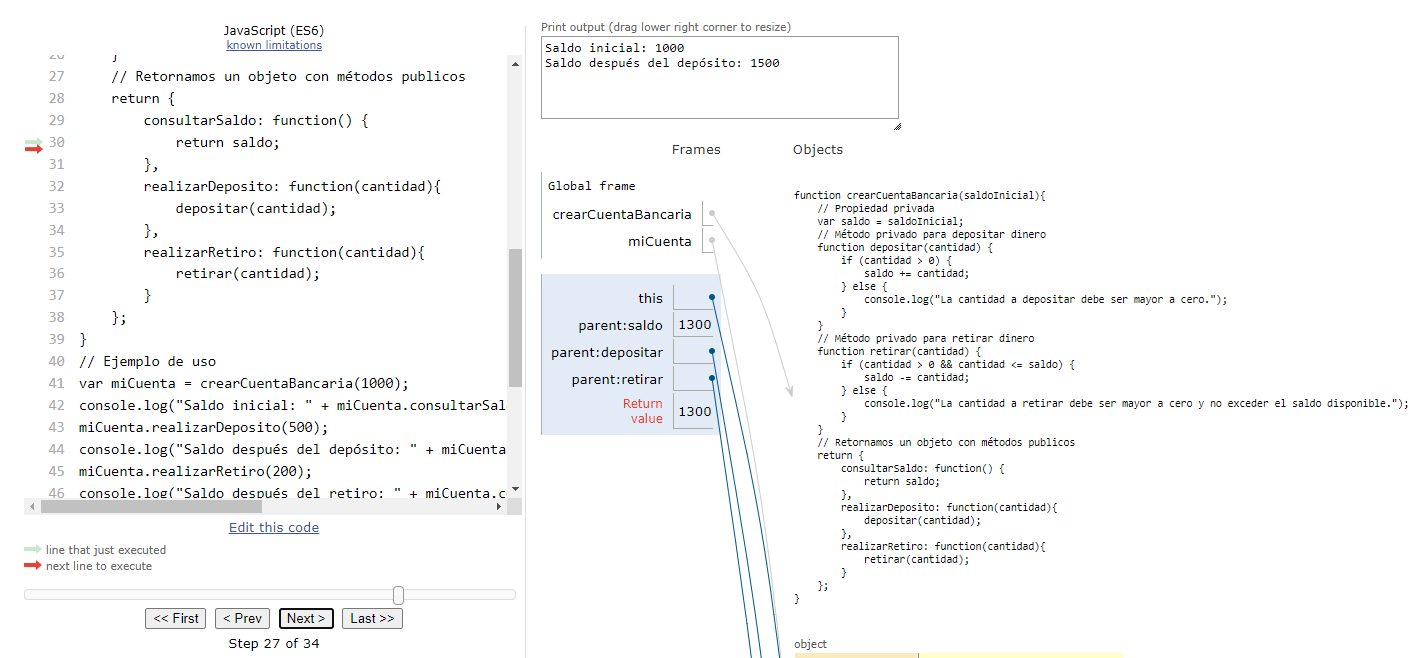
Pasa a la siguiente línea.

**Paso 26:**



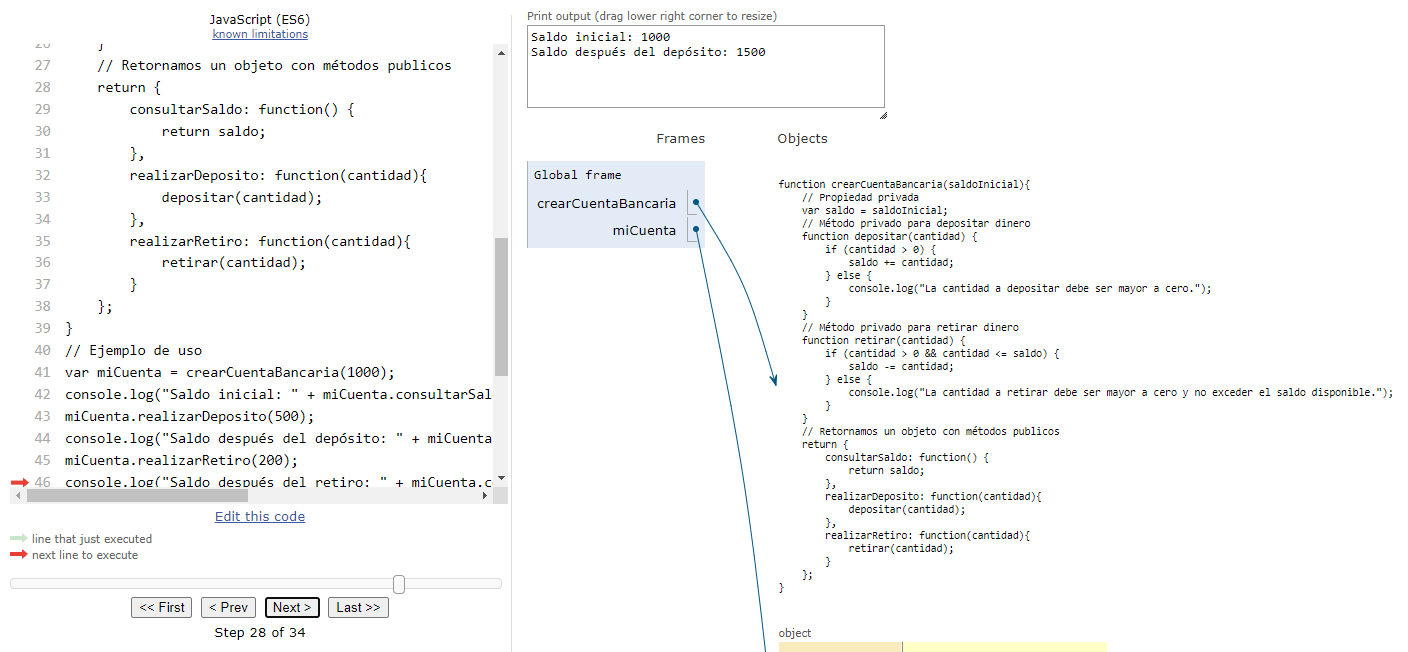
Busca la función consultarSaldo y obtiene el valor de 1300 de saldo.

**Paso 27:**



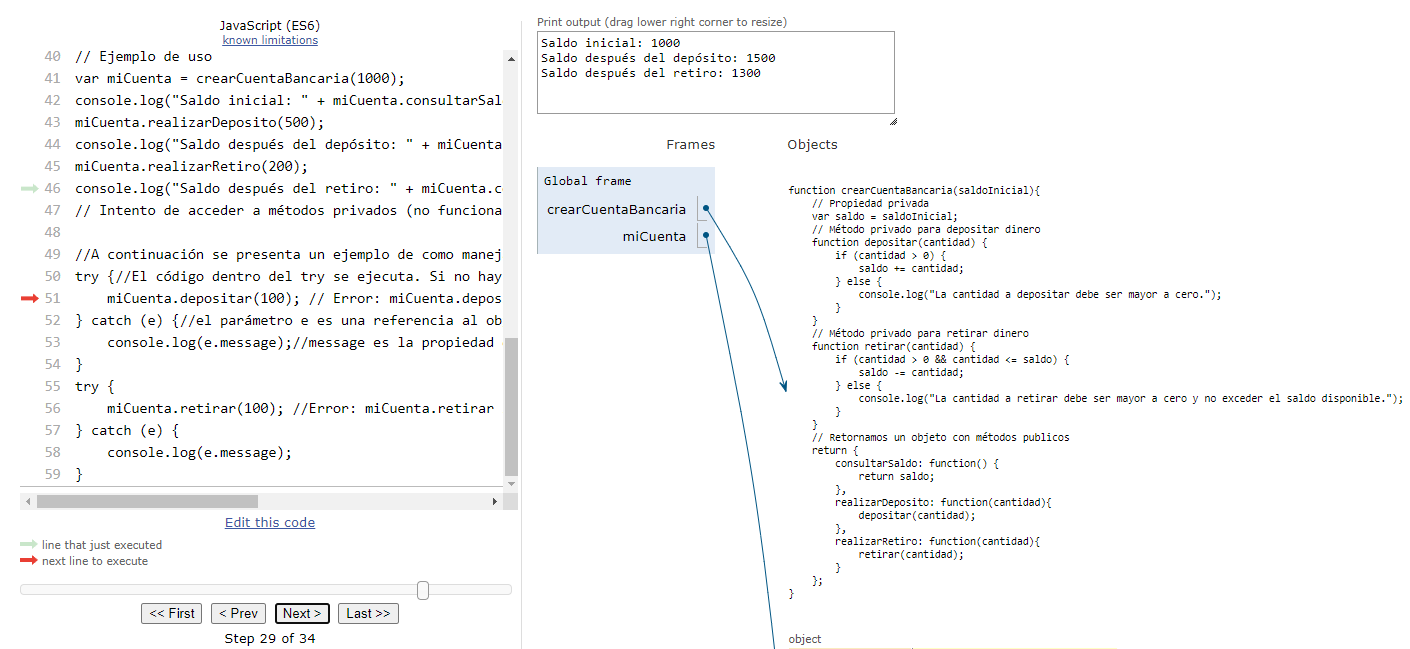
Return value obtiene el valor de Saldo.

**Paso 28:**



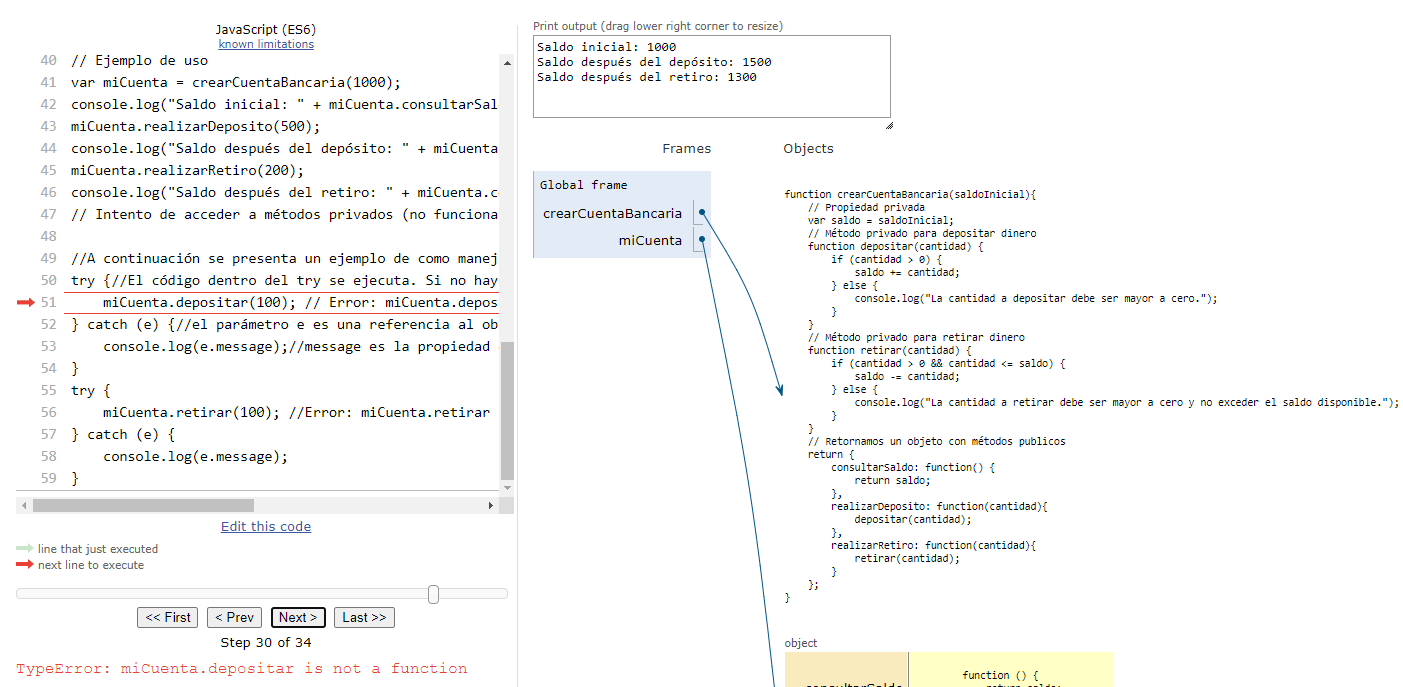
Despliega la información y cambia de línea.

**Paso 29:**



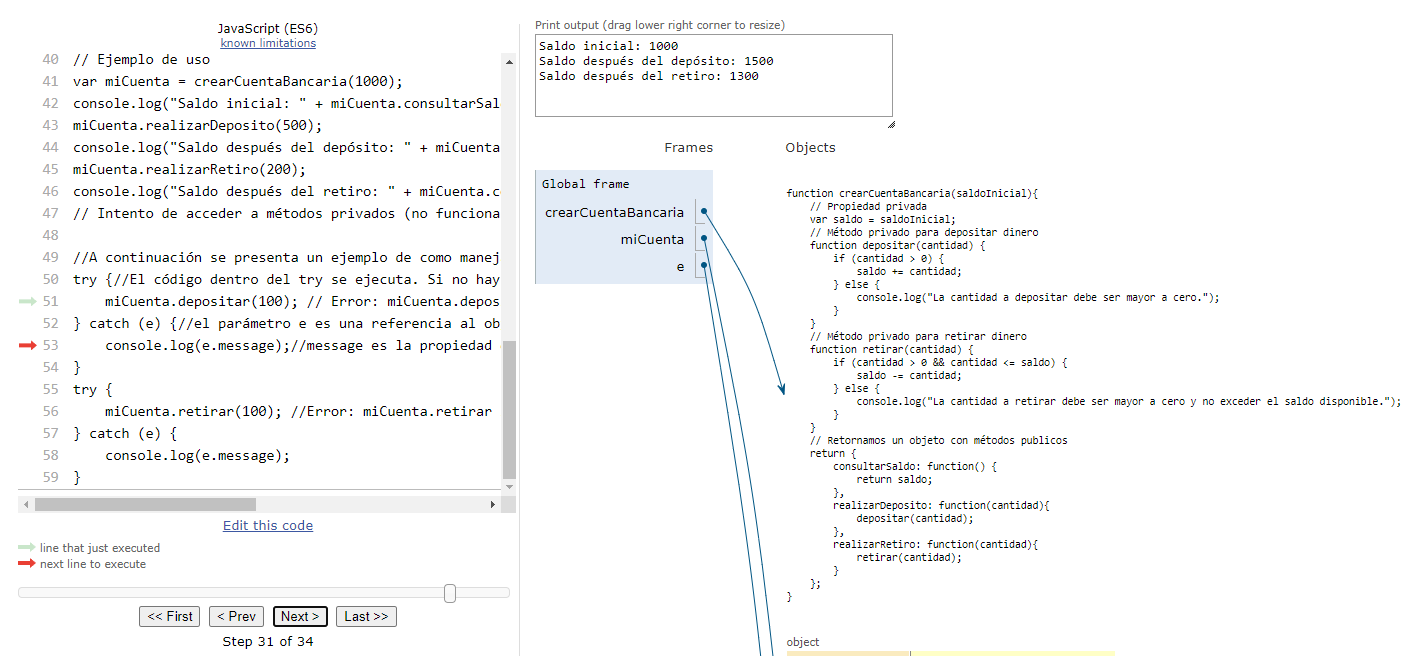
Intenta ejecutar el Try.

**Paso 30:**



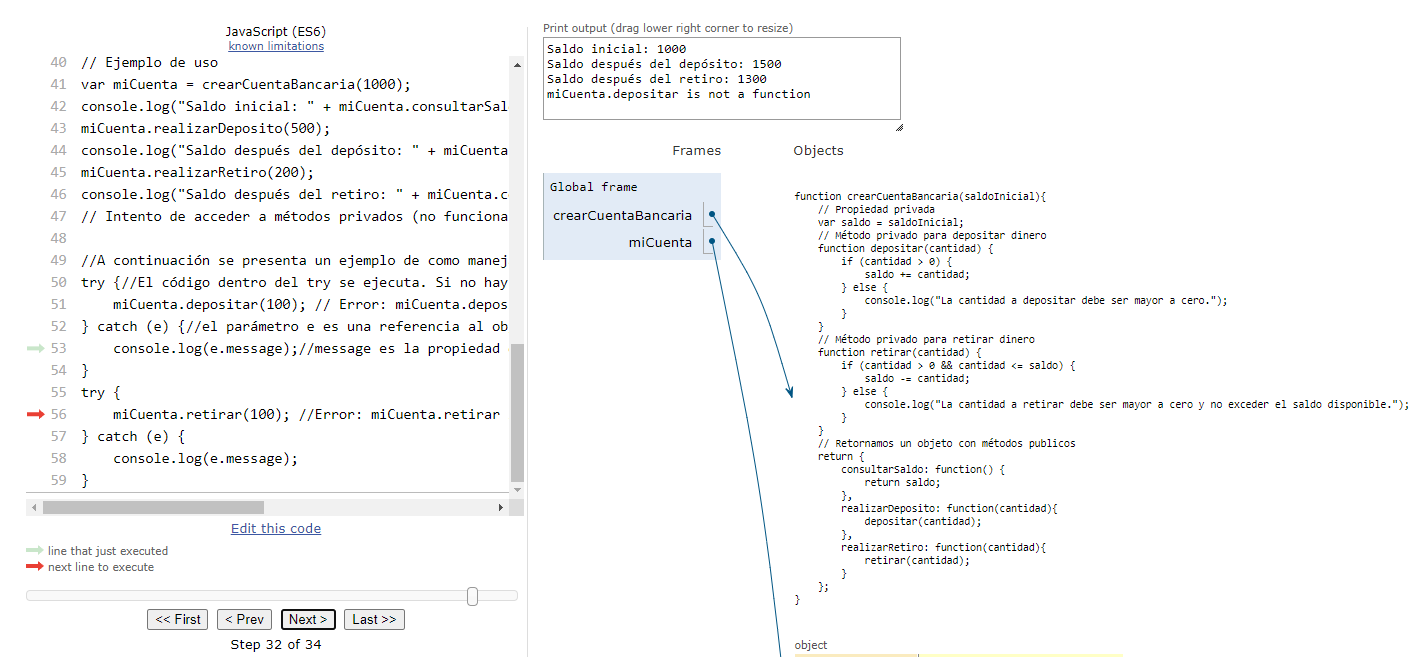
Obtenemos un error porque la función depositar nunca se creó en la función constructora y por lo tanto el objeto “miCuenta” nunca lo heredó.

**Paso 31:**



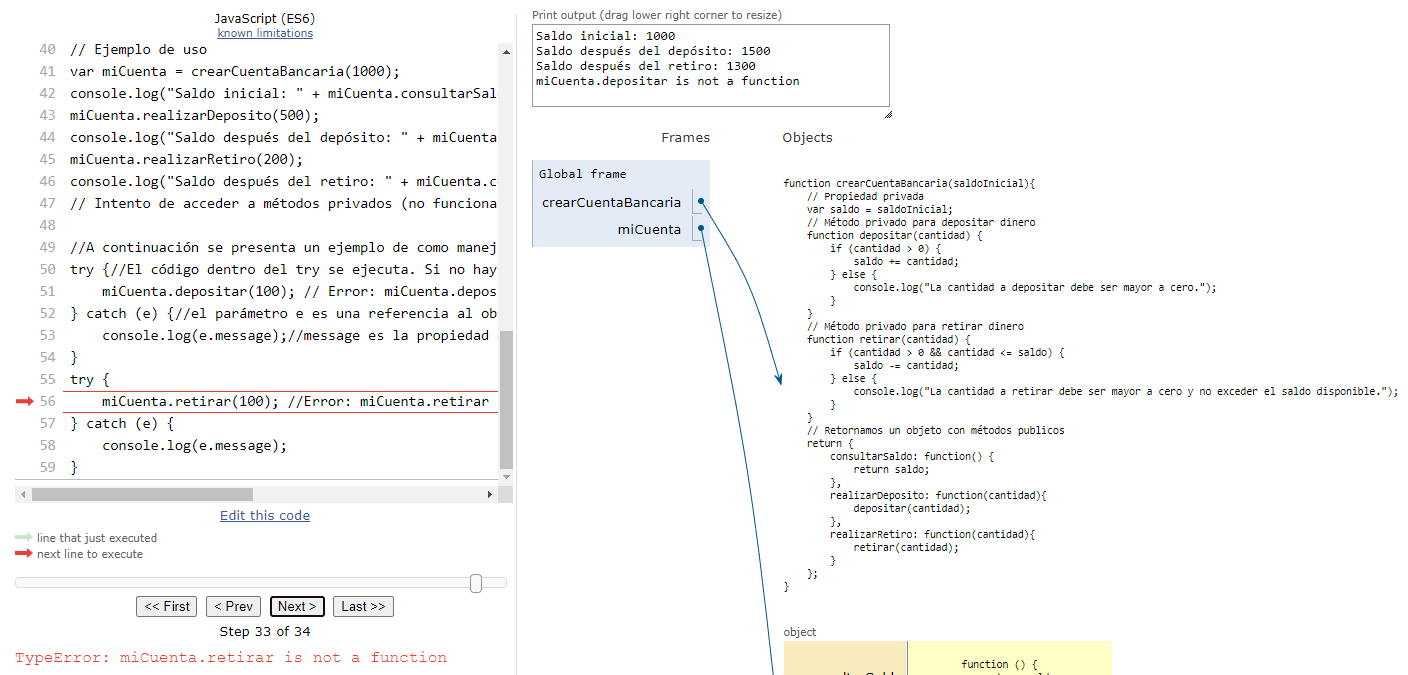
Intenta ejecutar el catch.

**Paso 32:**



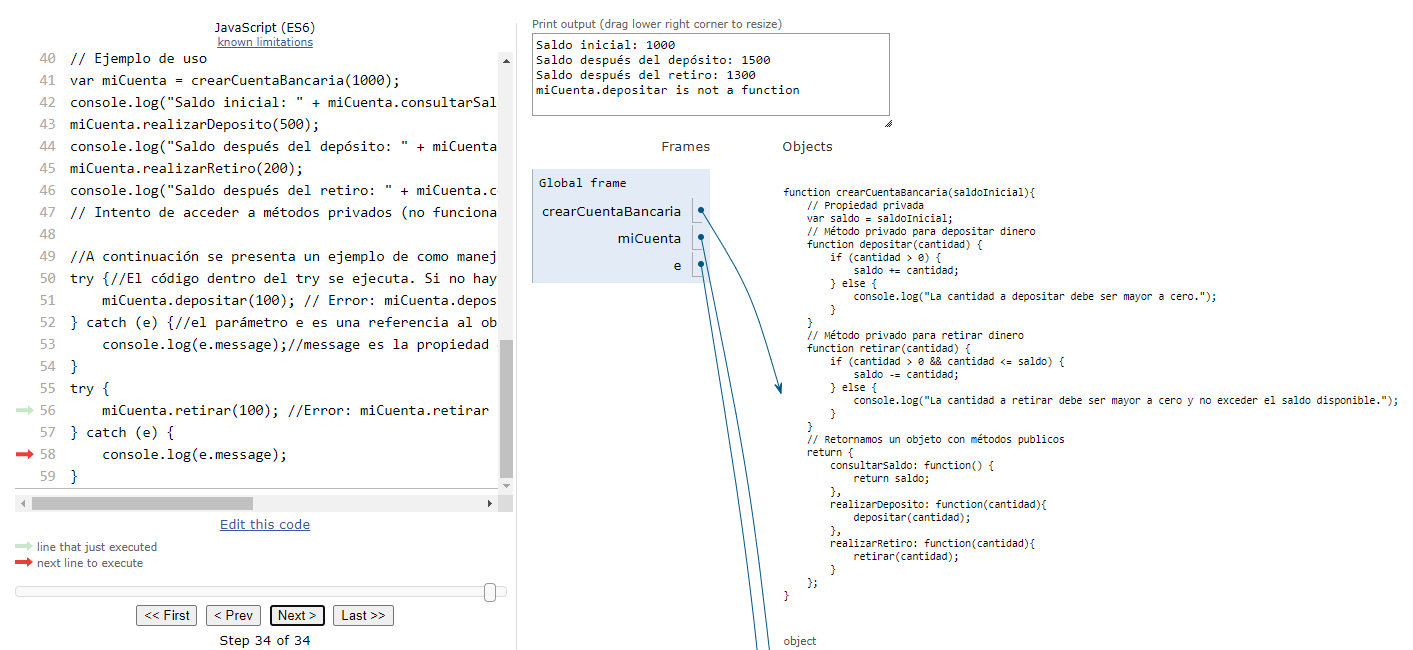
Ejecuta el catch, despliega la información en la consola y pasa a la siguiente línea.

**Paso 33:**



Intenta ejecutar el try y también obtiene un error porque retirar no es una función que “miCuenta” haya heredado.

**Paso 34:**



Ejecuta el catch, despliega la información en la consola.