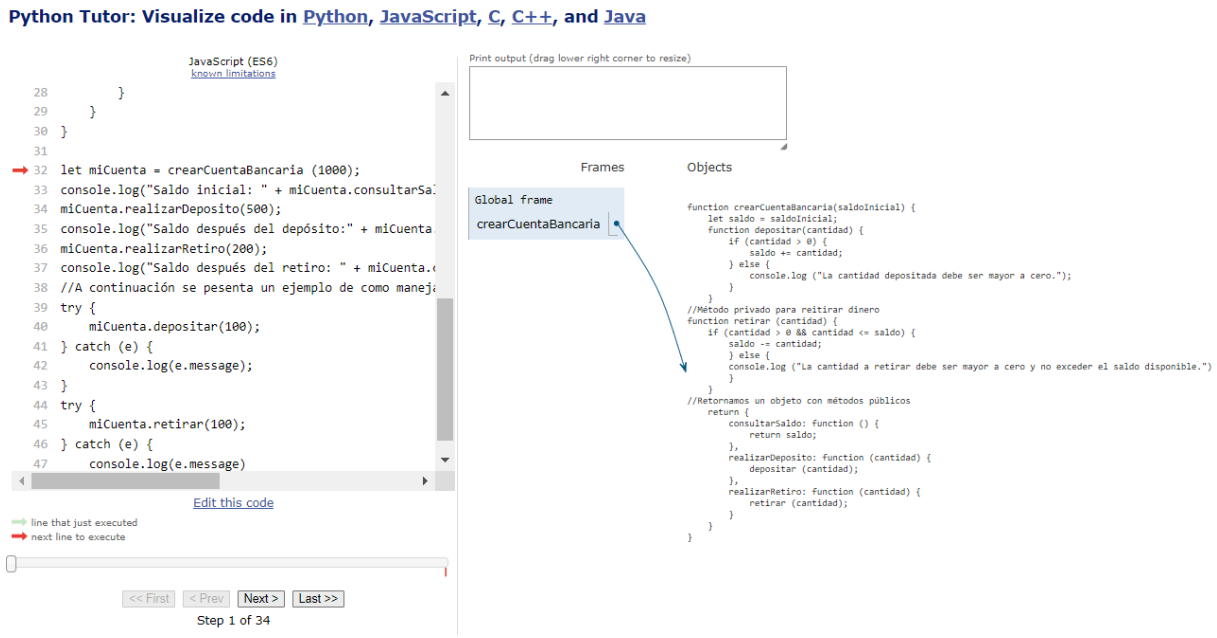
Tarea #8

Se llama a la función crearCuentaBancaria:

- Se invoca la función crearCuentaBancaria definida anteriormente.
- Se le pasa un argumento 1000, que representa el saldo inicial de la cuenta.

Asignación de la cuenta a la variable miCuenta:

- El resultado de la llamada a la función crearCuentaBancaria (que es un objeto que representa la cuenta bancaria) se asigna a la variable miCuenta.



Esta línea inicializa la variable saldo con el valor del saldo inicial de la cuenta.

1. let saldo = saldoInicial;

Esta línea declara una variable llamada saldo y le asigna el valor de la variable saldoInicial.

- let: Es una palabra clave de JavaScript que se utiliza para declarar variables con alcance de bloque. Esto significa que la variable saldo solo estará disponible dentro del bloque de código donde se declara.
- saldo: Es el nombre de la variable que se va a utilizar para almacenar el saldo de la cuenta bancaria.
- saldoInicial: Es una variable que se presume que contiene el valor del saldo inicial de la cuenta bancaria, proporcionado al crear la cuenta.

2. function depositar(cantidad) {

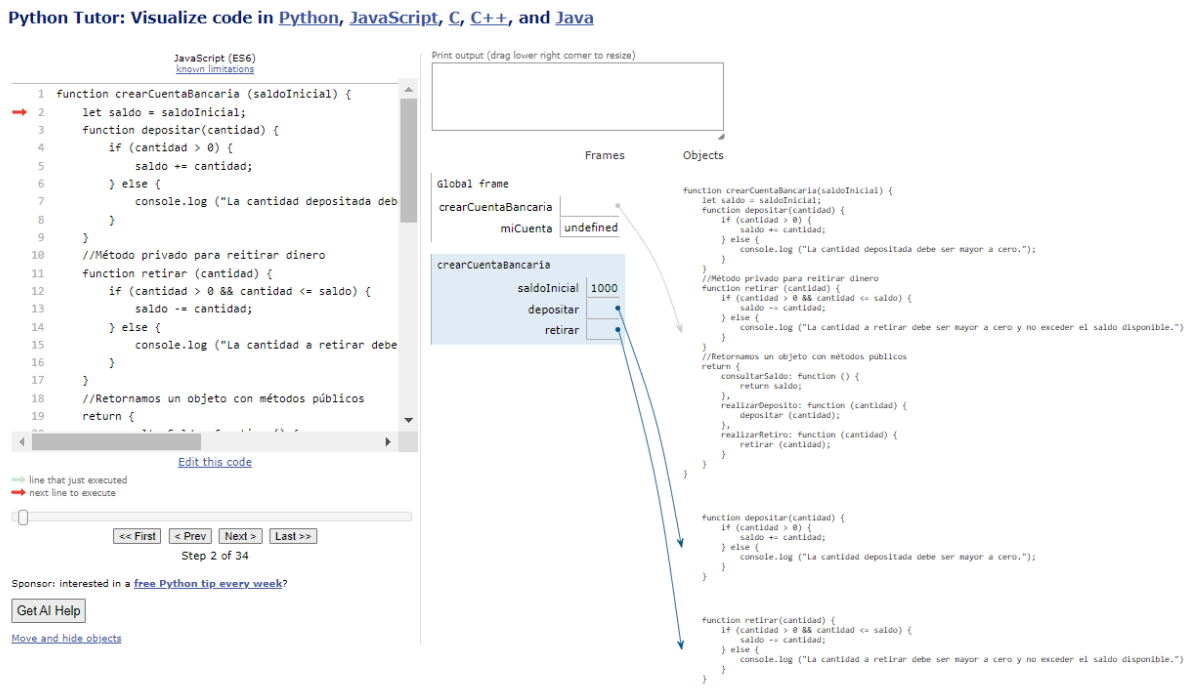
Esta línea define una función llamada depositar que recibe un parámetro llamado cantidad.
3. if (cantidad > 0) {
Esta línea inicia una instrucción condicional if que verifica si la cantidad a depositar (cantidad) es mayor que cero.
4. saldo += cantidad;
Esta línea aumenta el valor del saldo actual (saldo) en la cantidad a depositar (cantidad).

- saldo += cantidad;: Es un operador de asignación compuesto que suma el valor de cantidad al valor actual de saldo y almacena el resultado en saldo.

5. } else {
Esta línea inicia la rama else de la instrucción condicional if. Se ejecuta si la condición cantidad > 0 es false, es decir, si la cantidad a depositar no es mayor que cero.
6. console.log ("La cantidad depositada debe ser mayor a cero.");
Esta línea imprime un mensaje en la consola indicando que la cantidad depositada debe ser mayor a cero.
7. }
Esta línea cierra la instrucción condicional if.
8. function retirar (cantidad) {
Esta línea define una función llamada retirar que recibe un parámetro llamado cantidad. Similar a la función depositar, esta función simula la acción de realizar un retiro de la cuenta bancaria.
9. if (cantidad > 0 && cantidad <= saldo) {
Esta línea inicia una instrucción condicional if que verifica si la cantidad a retirar (cantidad) es mayor que cero y no excede el saldo disponible (saldo).

- cantidad > 0 && cantidad <= saldo: Es la condición que se evalúa. Si es true, significa que la cantidad a retirar es positiva y no supera el saldo actual.

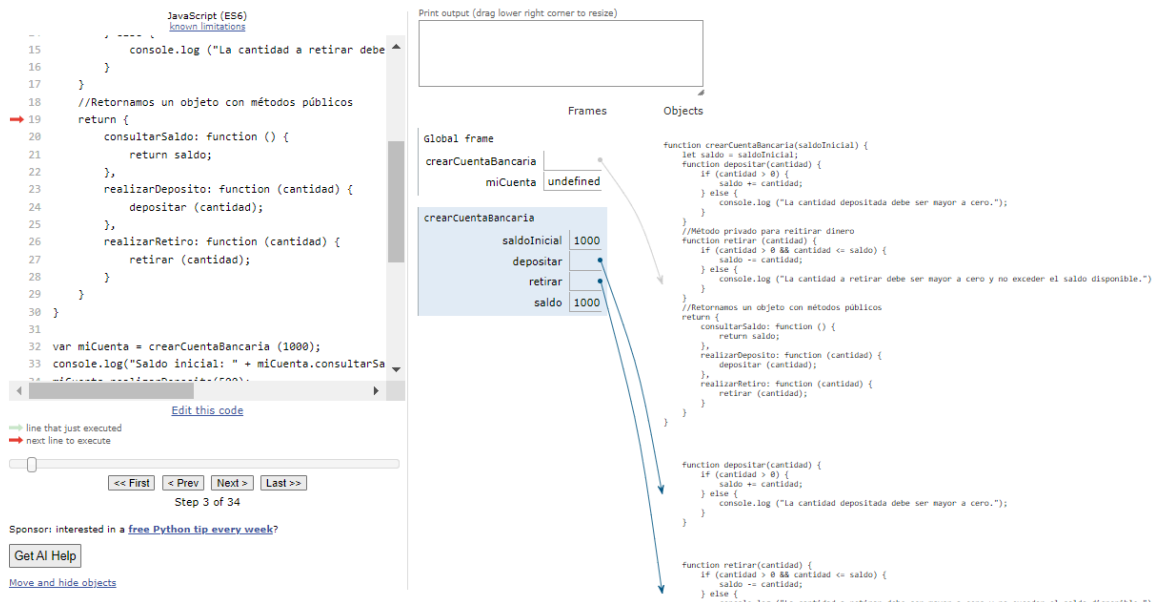
10. saldo -= cantidad;
Esta línea disminuye el valor del saldo actual (saldo) en la cantidad a retirar (cantidad).
11. } else {
Esta línea inicia la rama else de la instrucción condicional if. Se ejecuta si la condición cantidad > 0 && cantidad <= saldo es false, es decir, si la cantidad a retirar no es válida.
12. console.log ("La cantidad a retirar debe ser mayor a cero y no exceder el saldo disponible.")
Esta línea imprime un mensaje en la consola indicando que la cantidad a retirar debe ser mayor a cero y no exceder el saldo disponible.

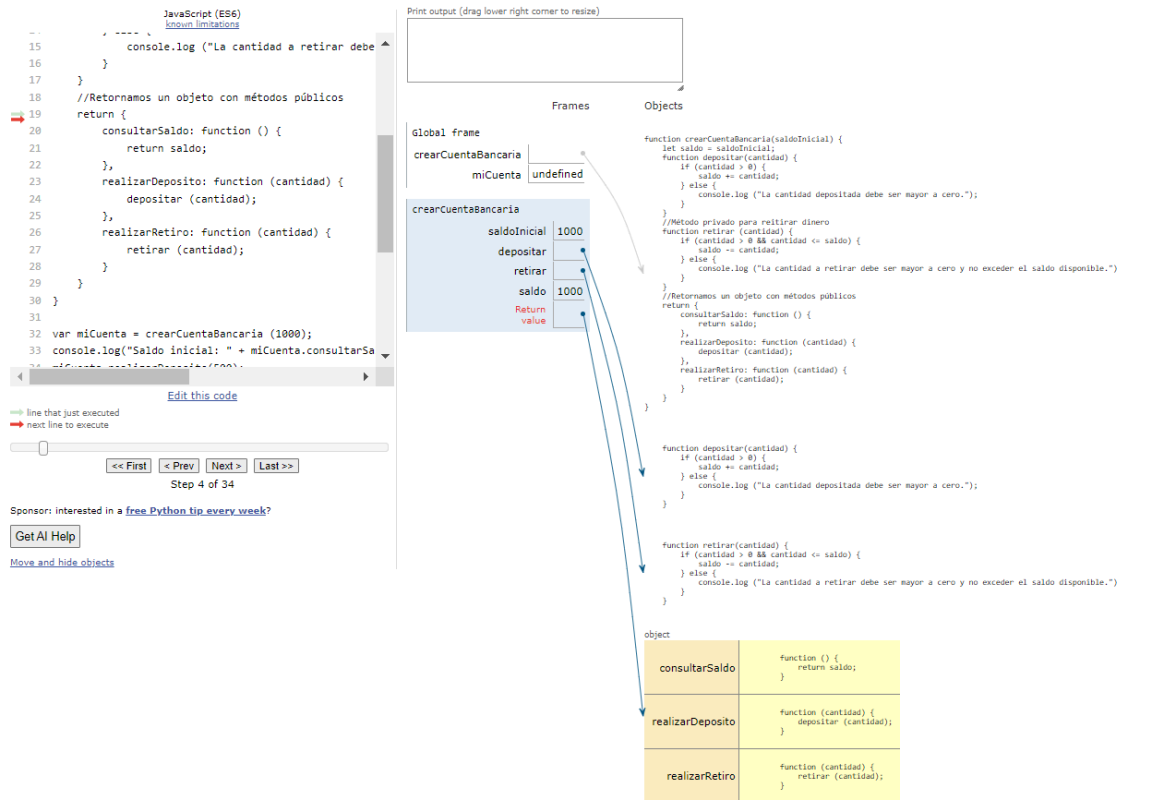


Este objeto literal define tres métodos públicos que permiten a los usuarios:

- Consultar el saldo actual de la cuenta.
- Realizar depósitos en la cuenta.
- Realizar retiros de la cuenta.

Al retornar este objeto desde la función crearCuentaBancaria, se proporciona una interfaz para interactuar con la cuenta bancaria simulada.

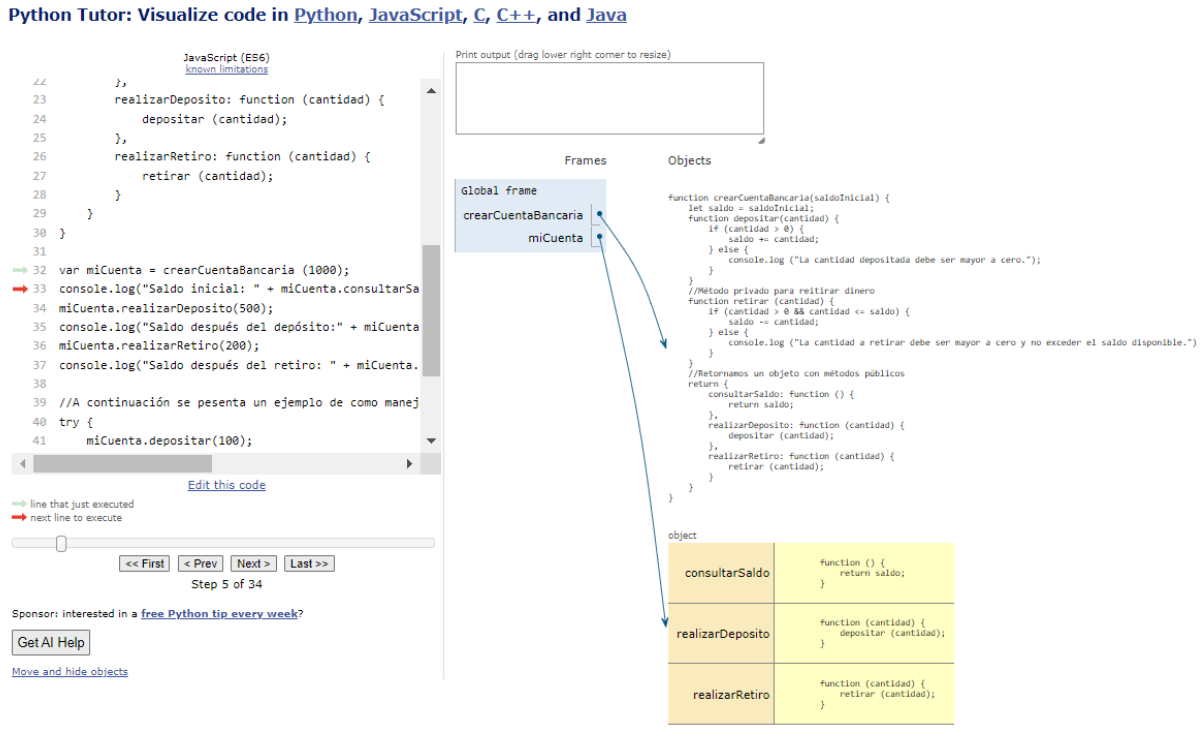




var miCuenta = crearCuentaBancaria(1000);

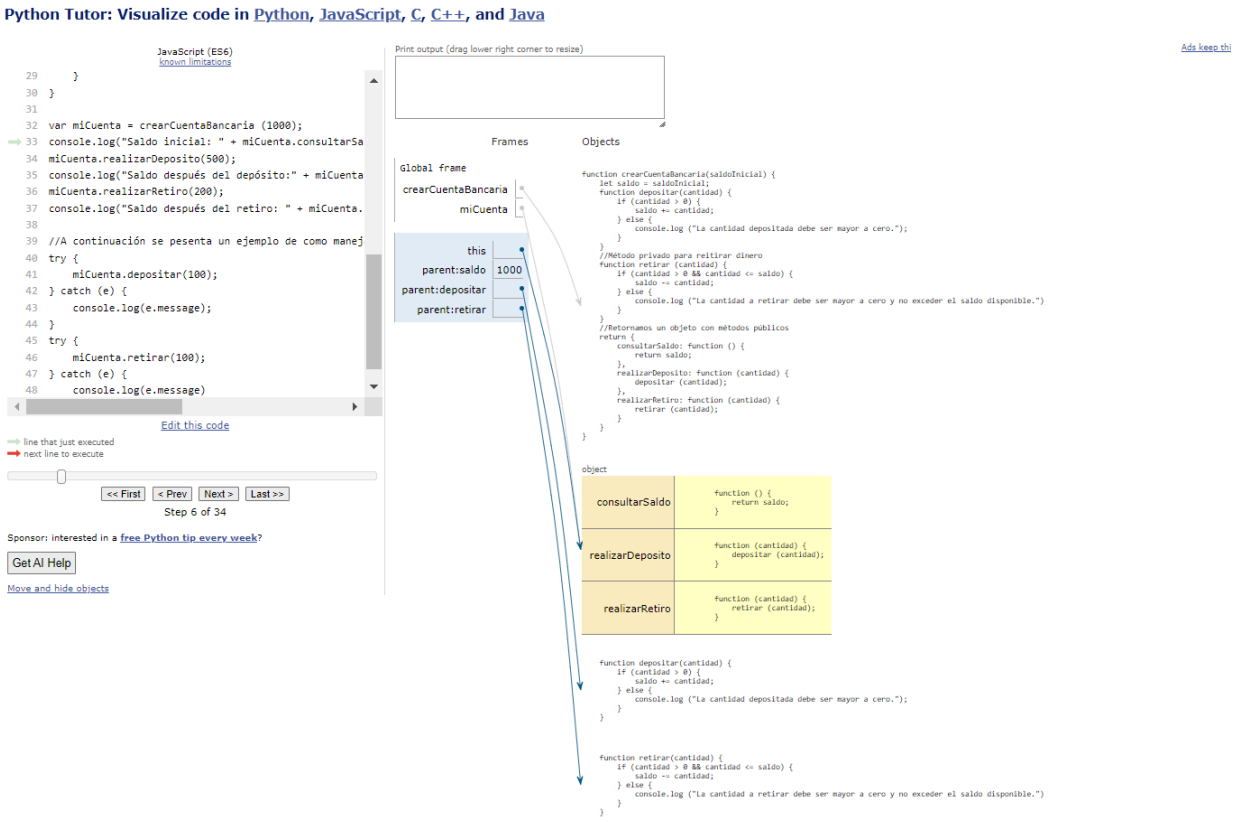
- Esta línea crea una cuenta bancaria utilizando la función crearCuentaBancaria y almacena la referencia a la cuenta en la variable miCuenta.

- crearCuentaBancaria(1000): Se invoca la función crearCuentaBancaria pasándole como argumento el valor inicial del saldo (1000).
- var miCuenta: Se declara una variable llamada miCuenta utilizando la palabra clave var.
- =: Se asigna el valor retornado por la función crearCuentaBancaria (que es la referencia a la cuenta creada) a la variable miCuenta.

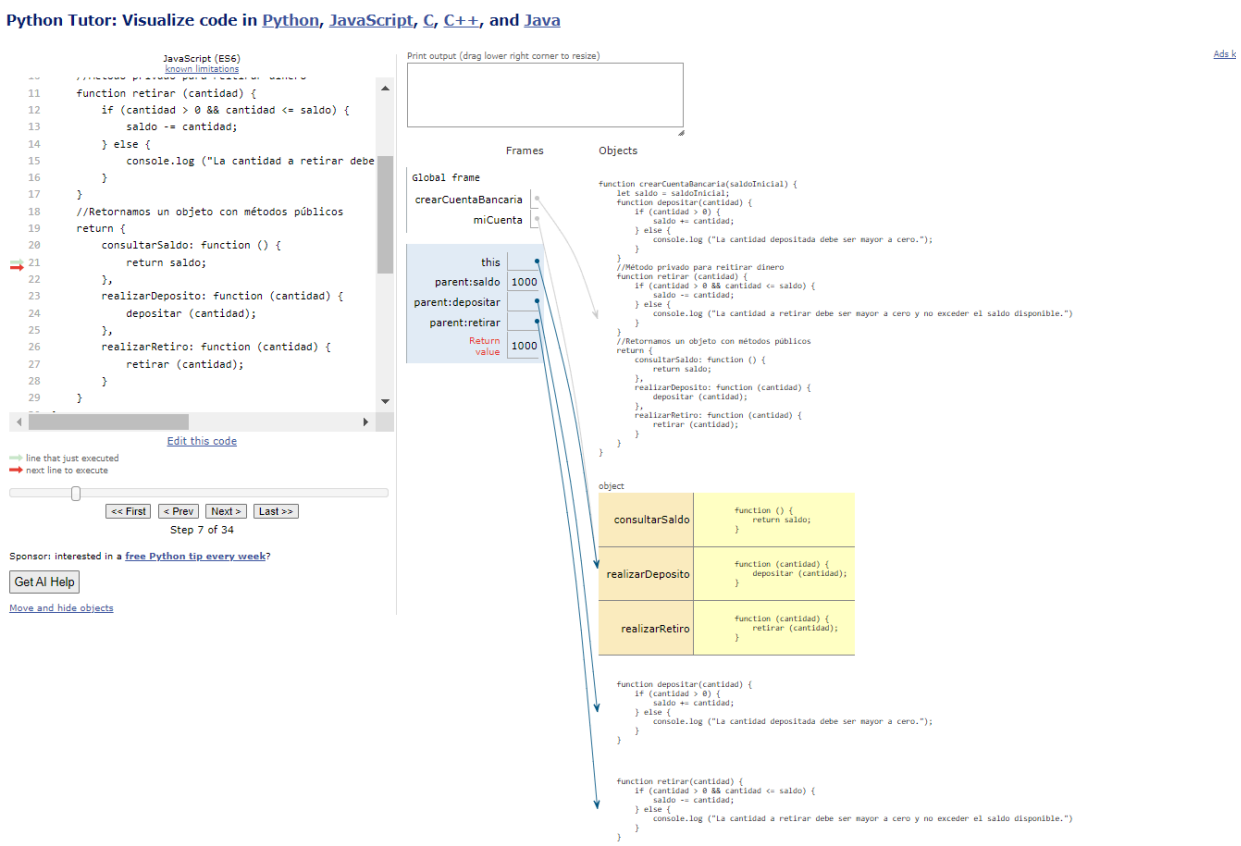


console.log("Saldo inicial: " + miCuenta.consultarSaldo());

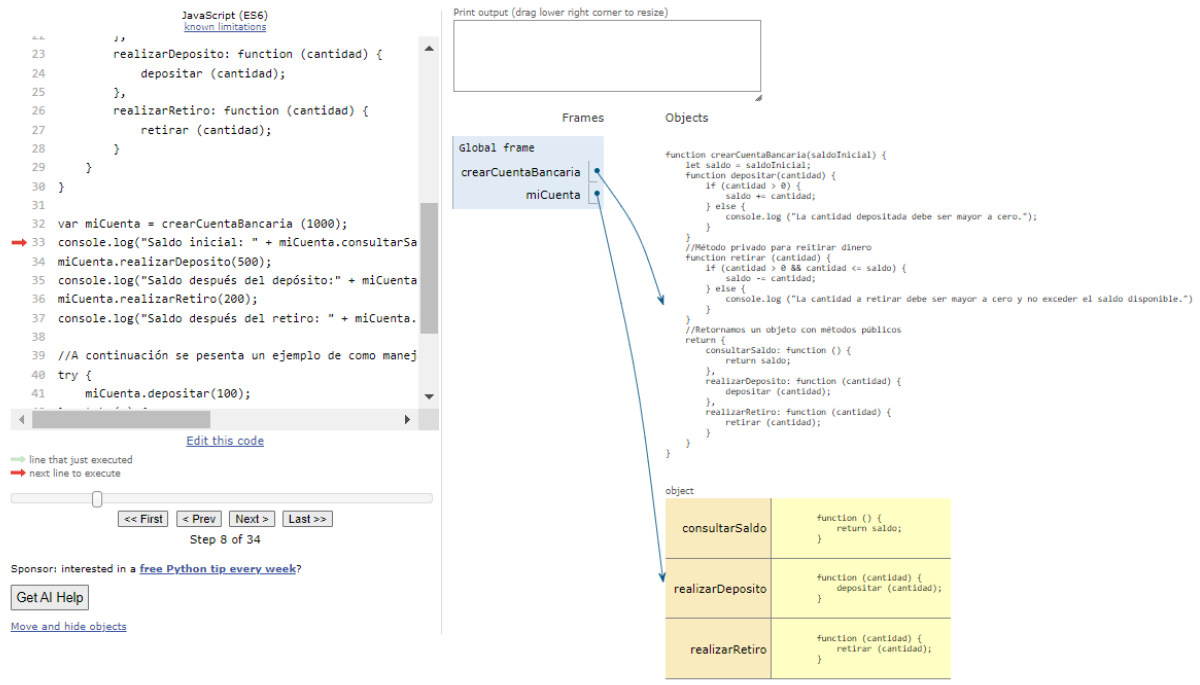
- Esta línea imprime el saldo inicial de la cuenta en la consola del navegador.



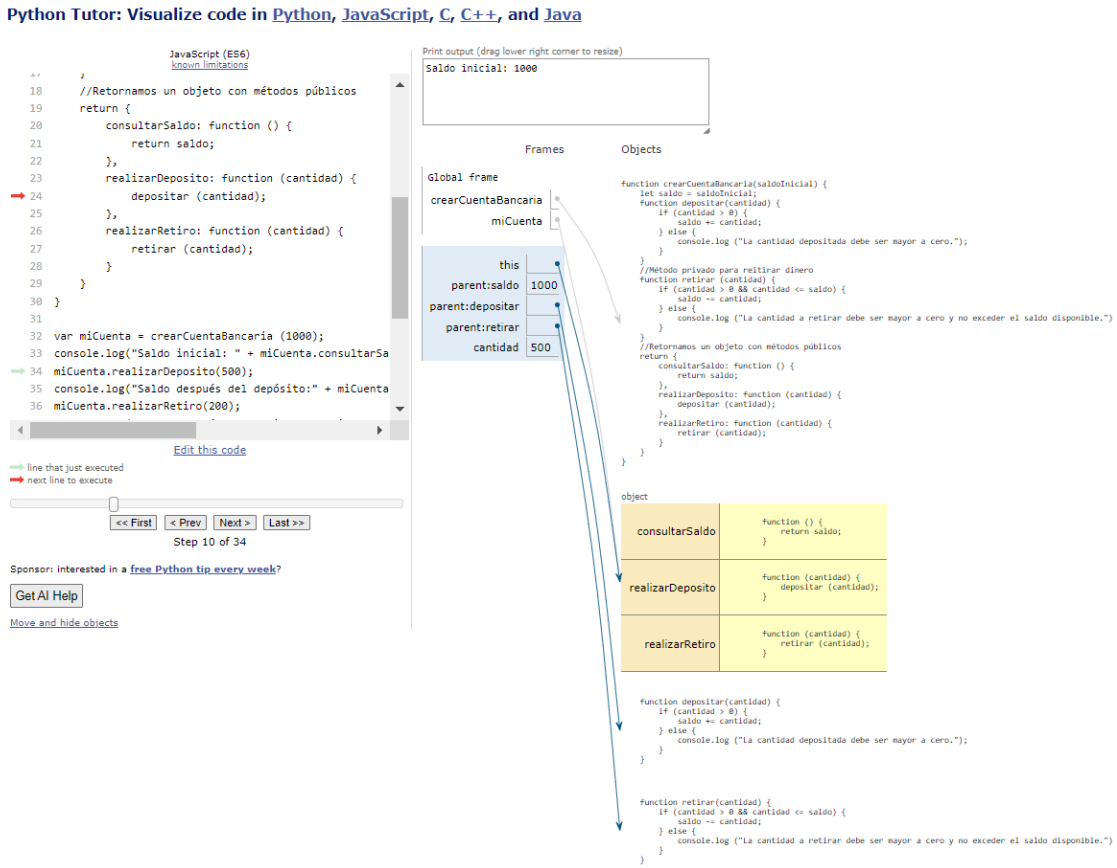
Se retorna el saldo



Se imprime saldo inicial, saldo después del depósito y saldo después del retiro

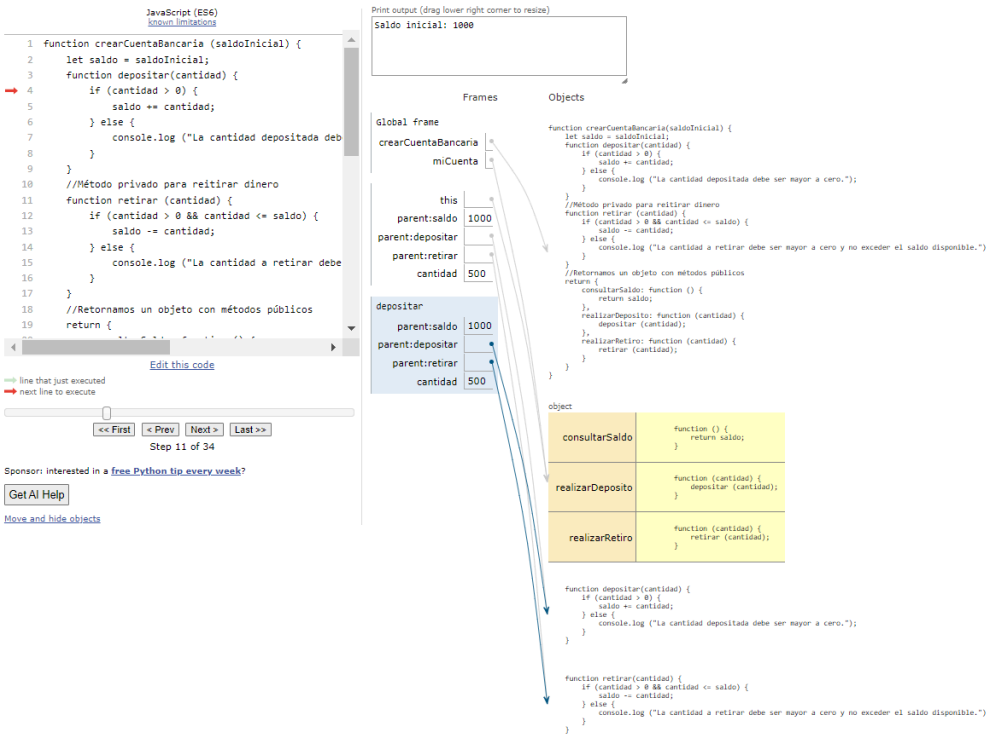


Se realiza un depósito de 500 a la cuenta.

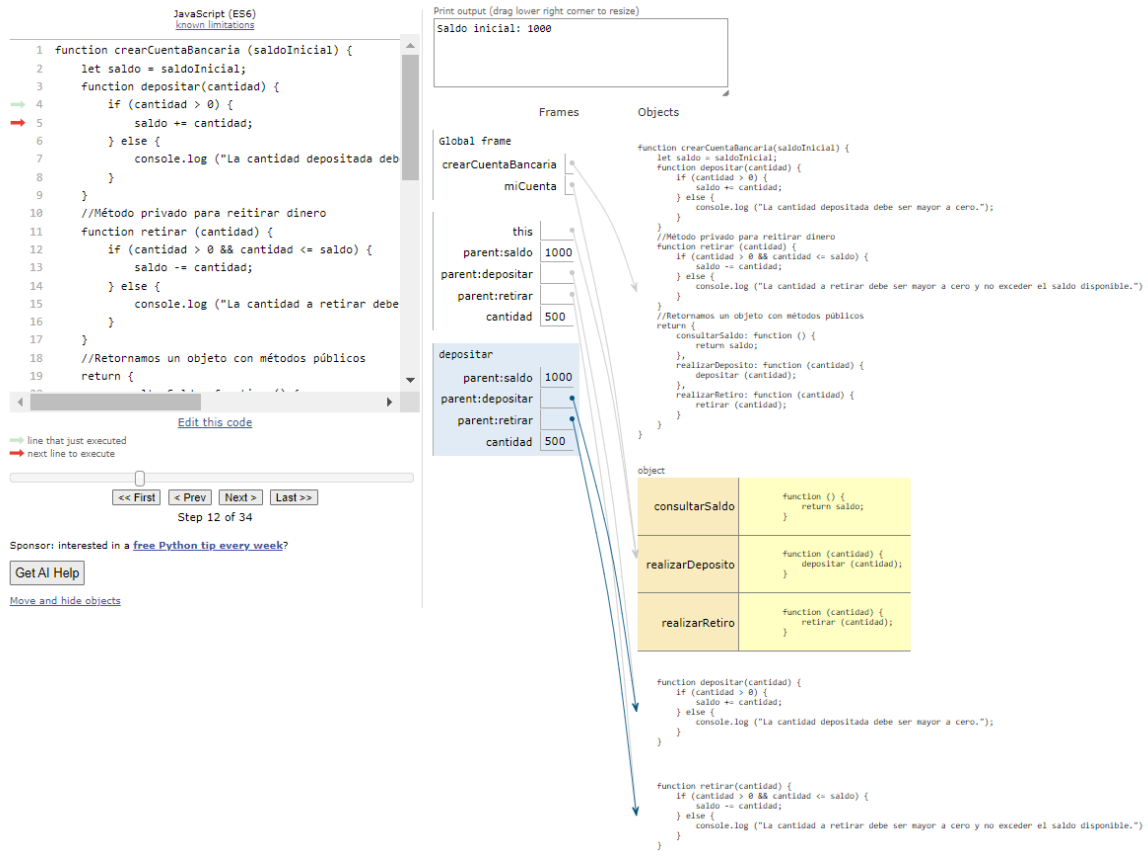


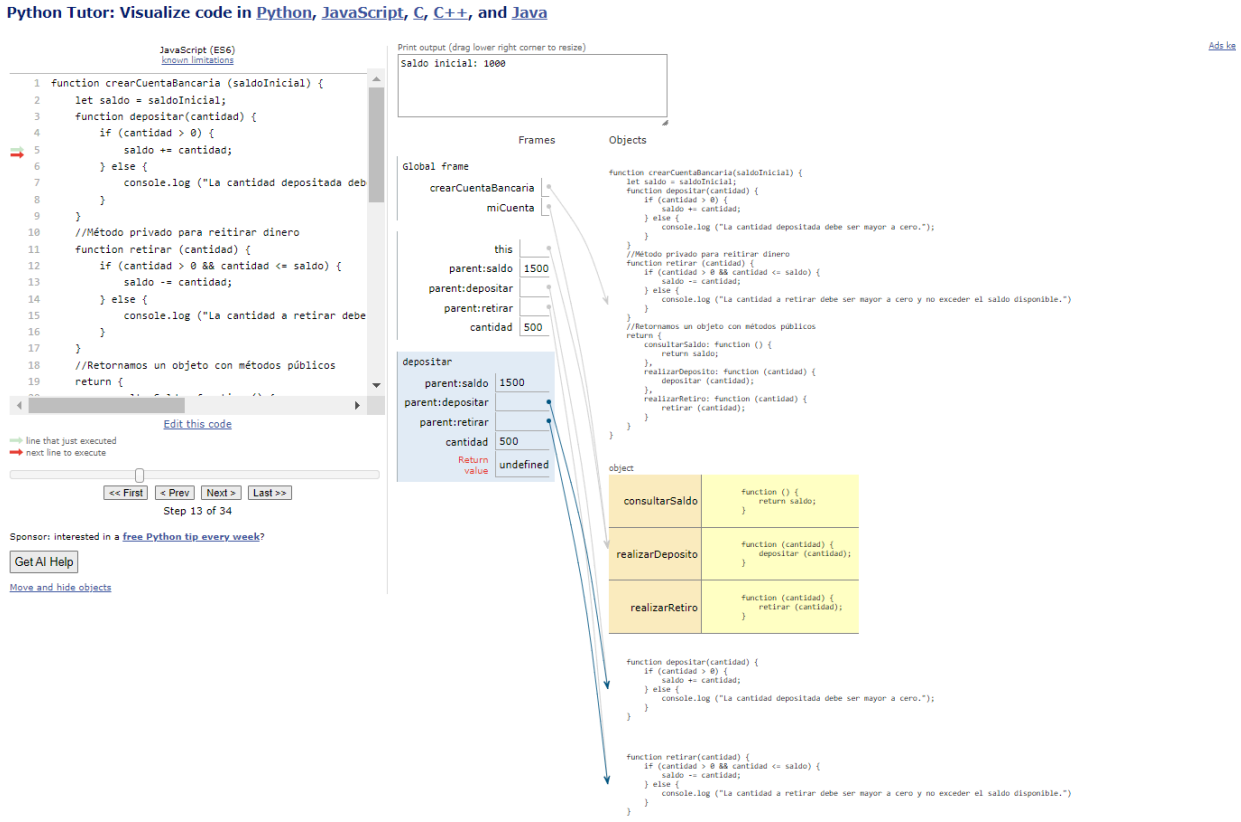
Si la cantidad es mayor a 0 suma se actualiza a suma mas cantidad y si no se imprime "La cantidad depositada debe ser mayor a cero.".

Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java



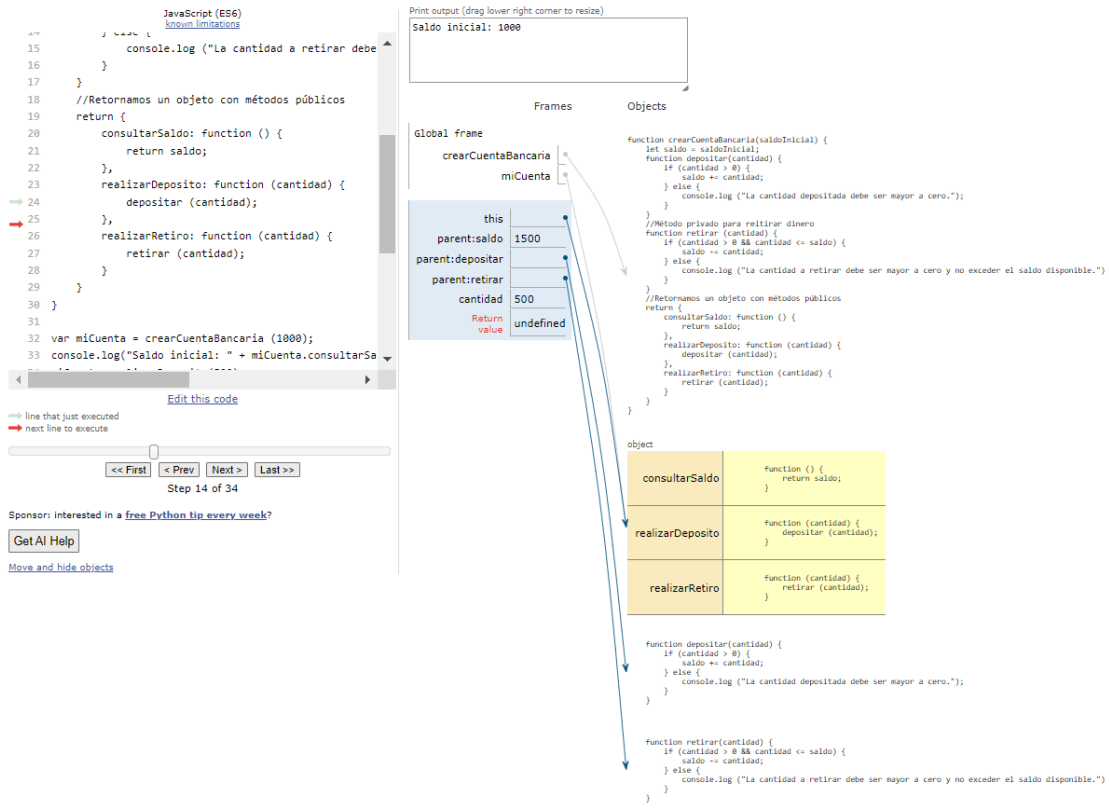
Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java



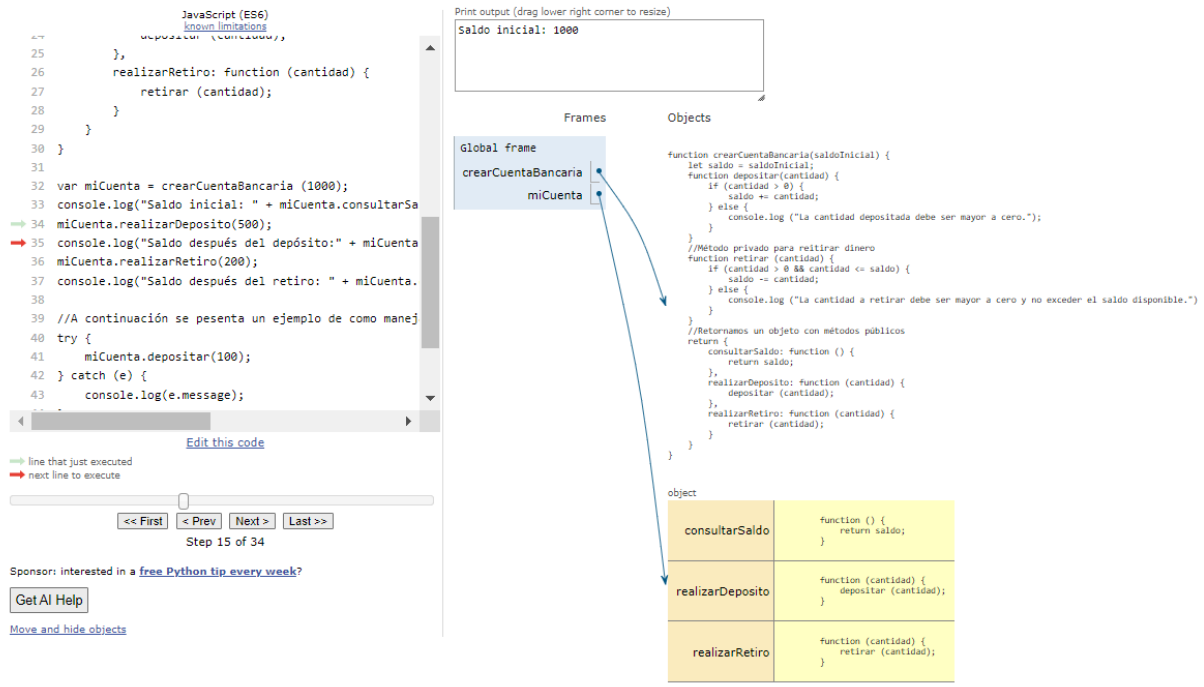


Se realiza el deposito exitosamente:

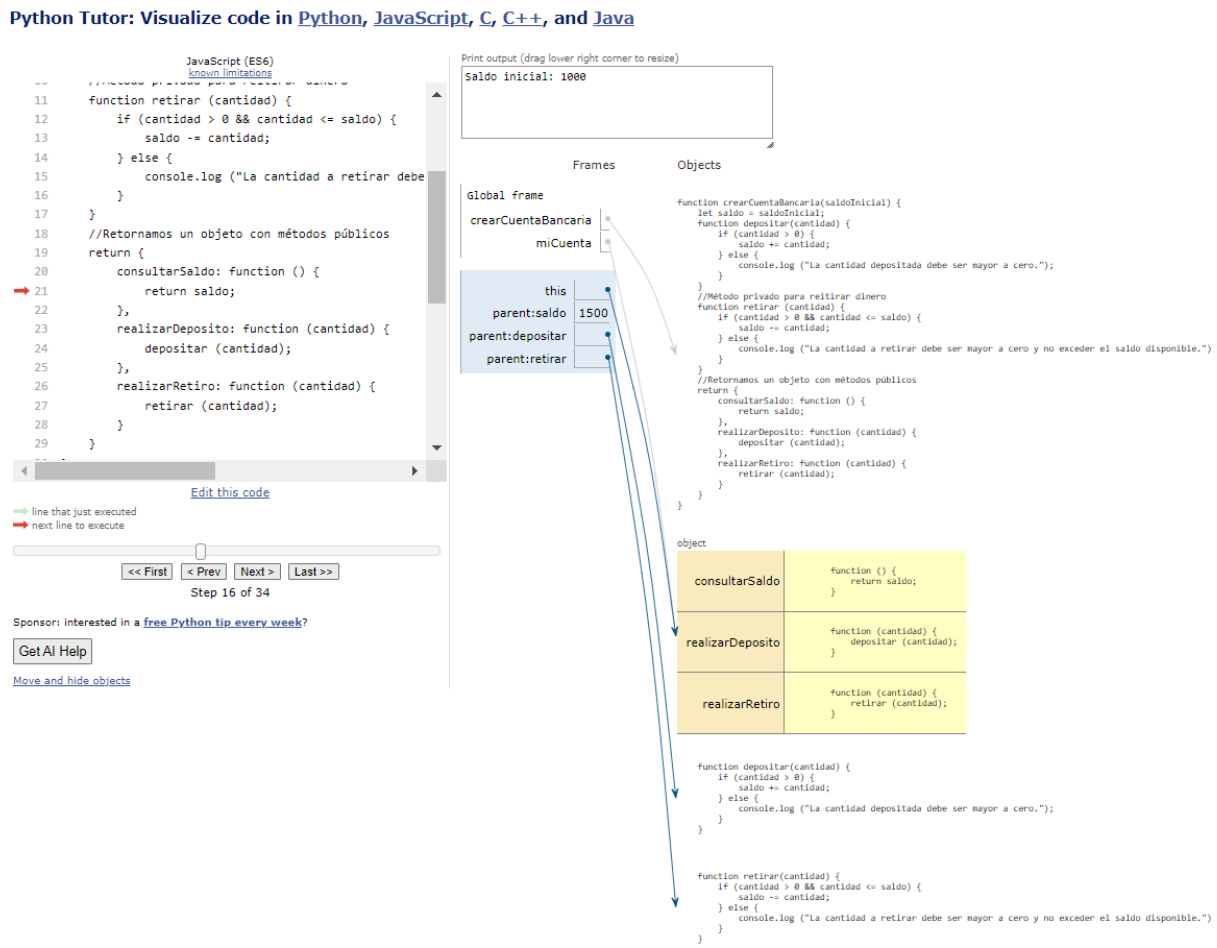
Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java



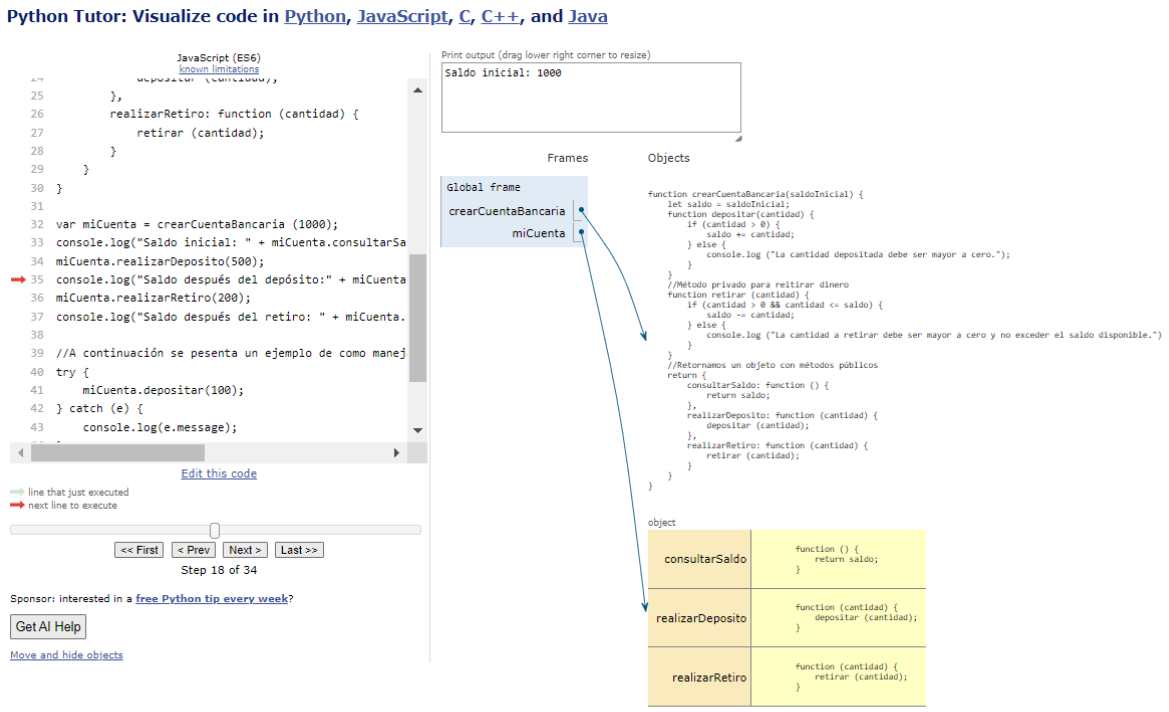
Deposito de 500



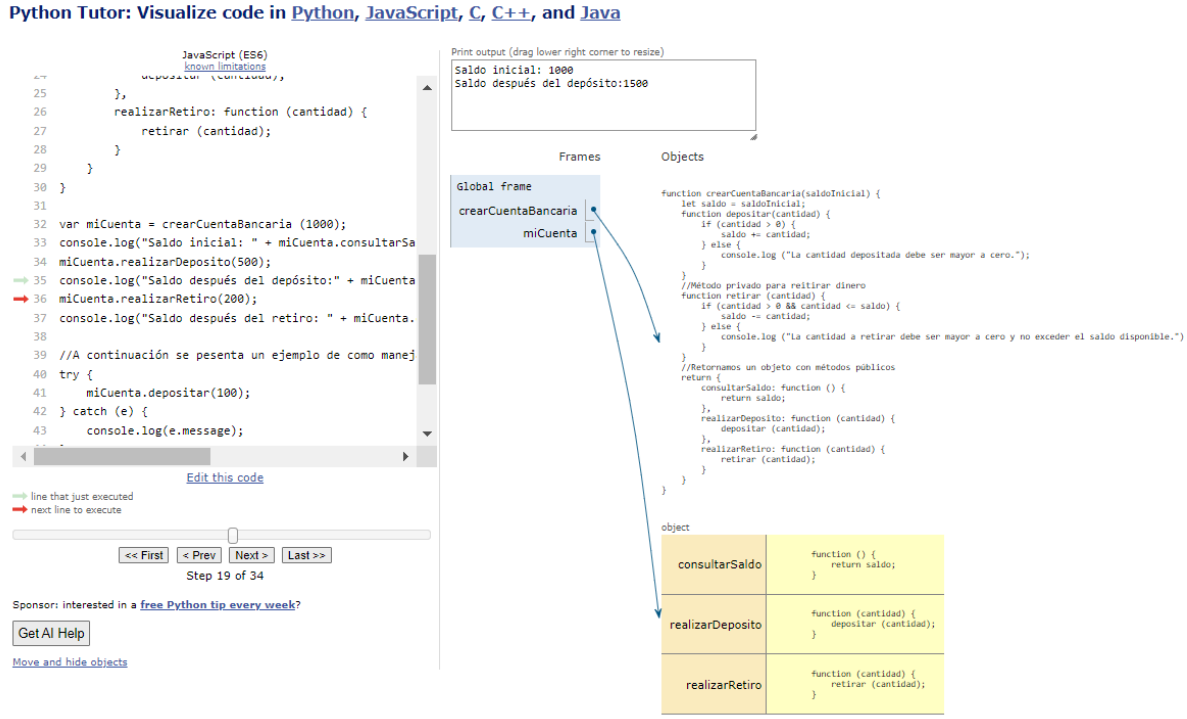
Se retorna el saldo



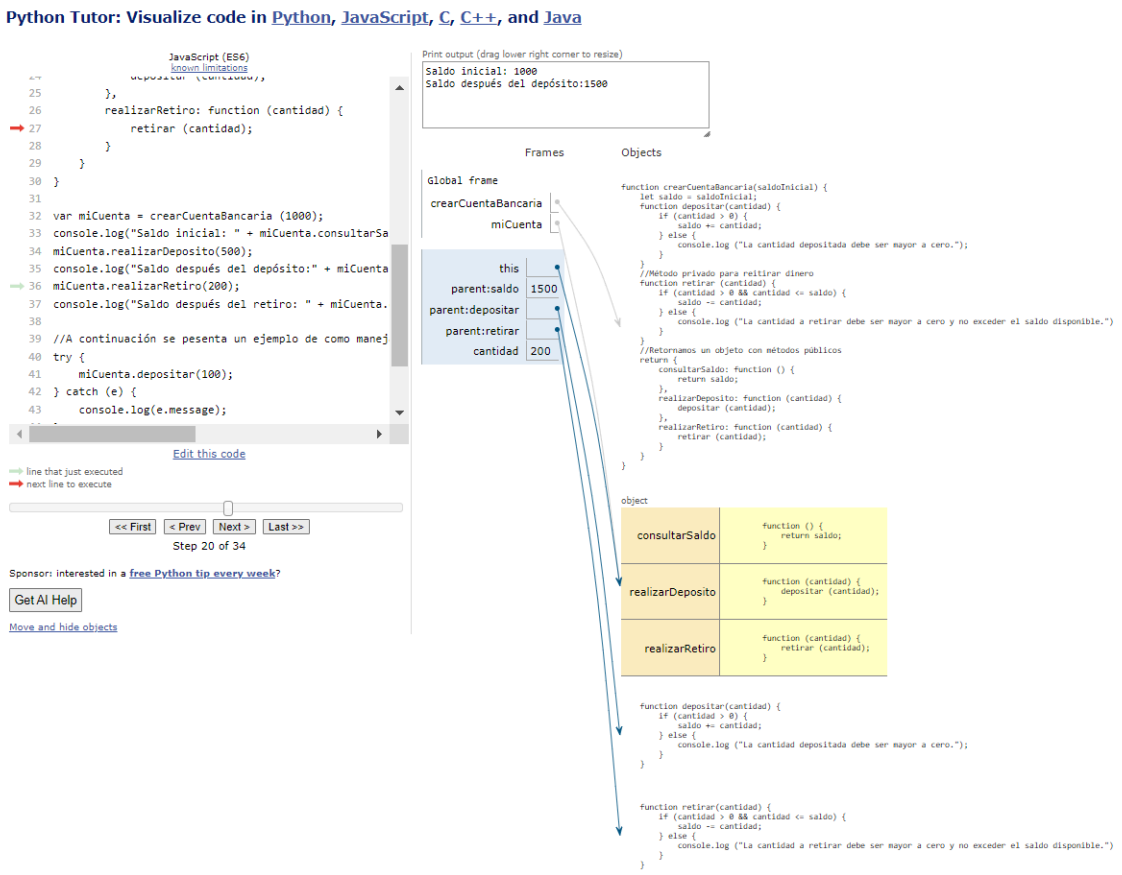
Se imprime saldo despues de deposito



Retiro de 200

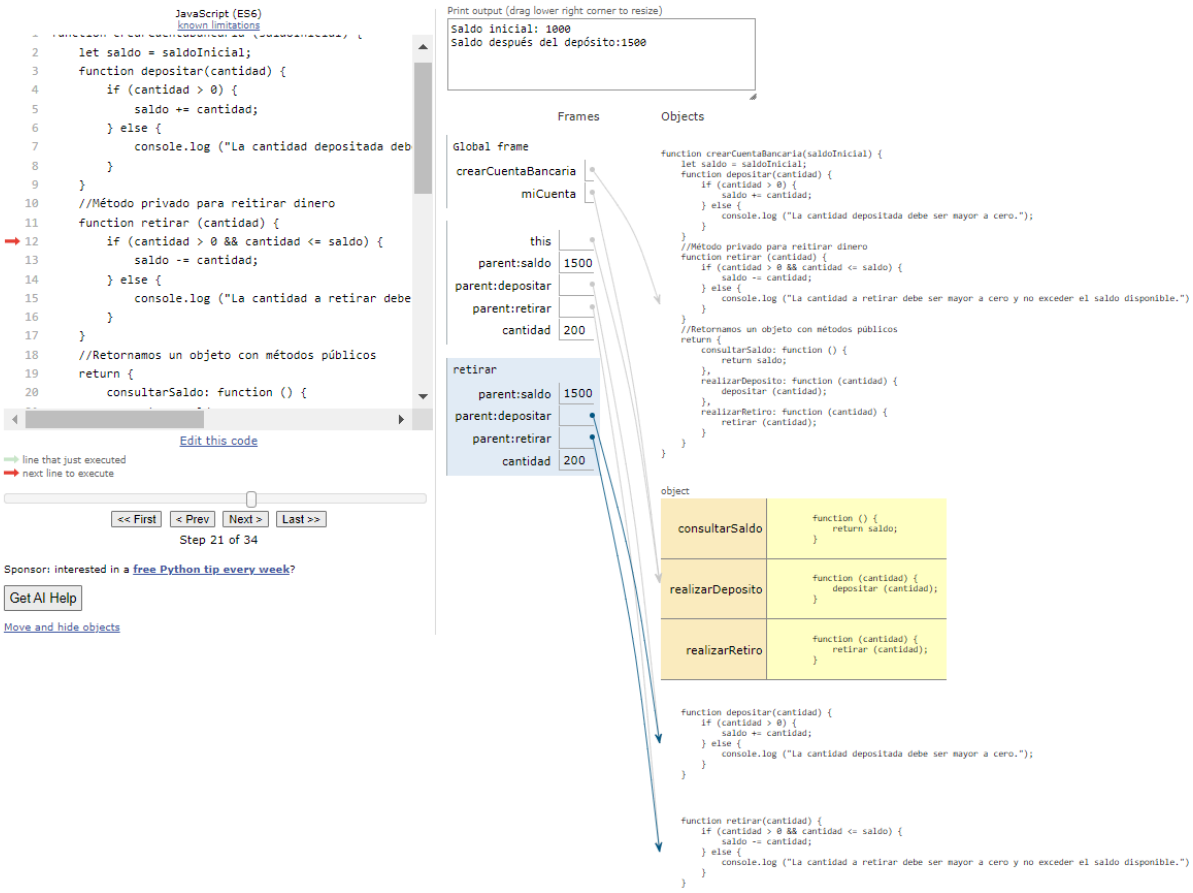


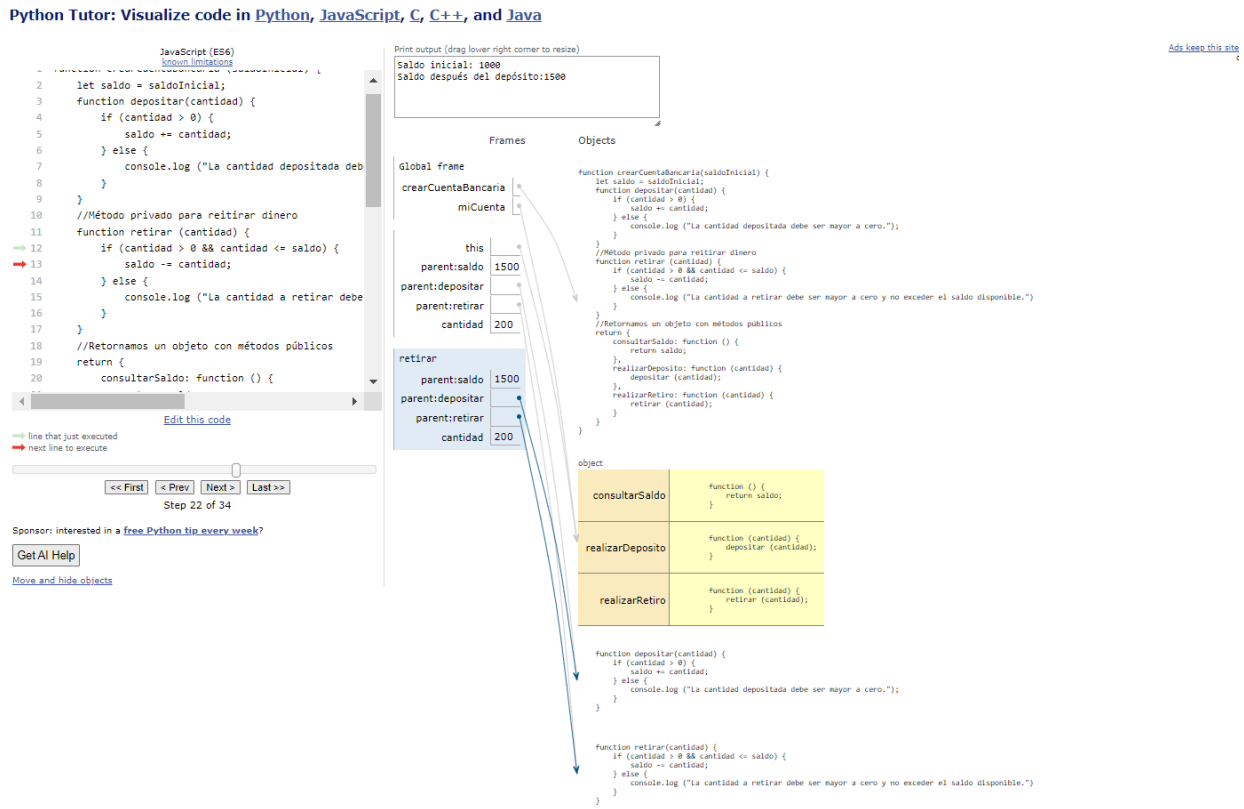
Funcion de retiro

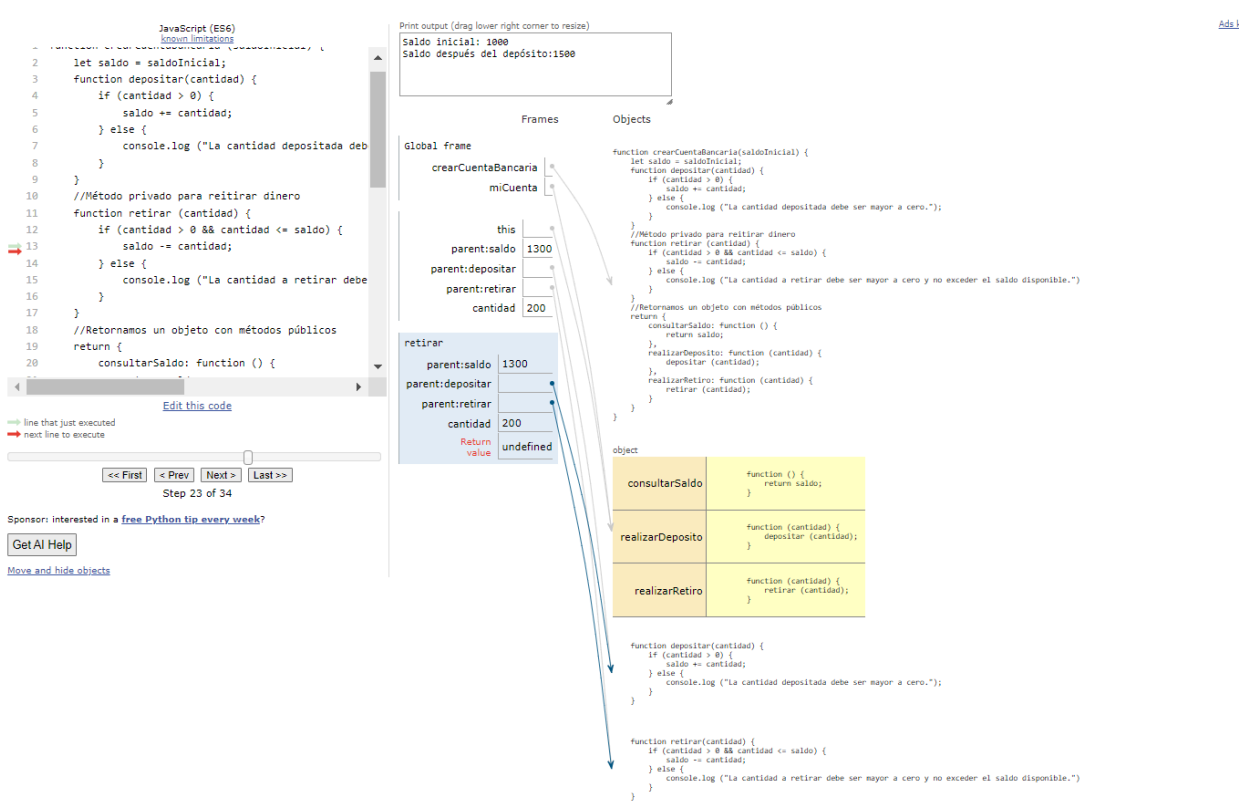


Si la cantidad es mayor que cero y menor o igual que saldo entonces se actualiza el saldo a saldo menos cantidad de otra manera se imprime: "La cantidad a retirar debe ser mayor a cero y no exceder el saldo disponible."

Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java

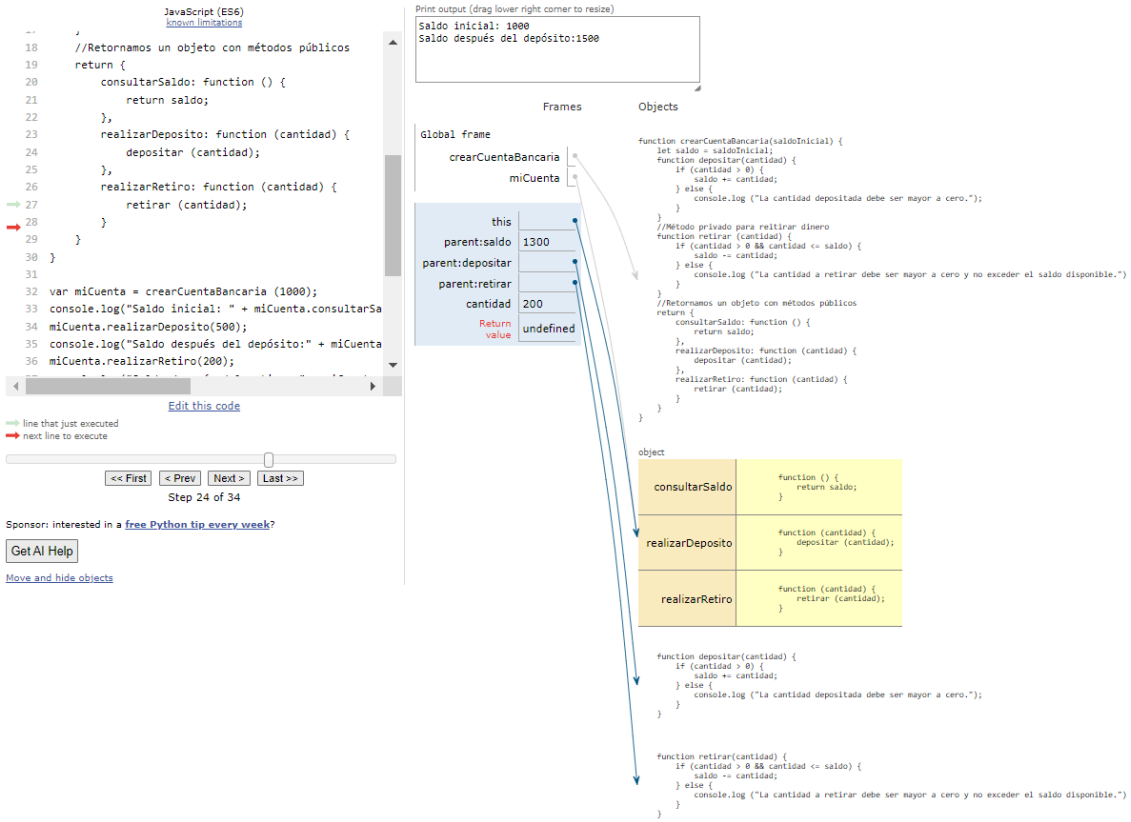




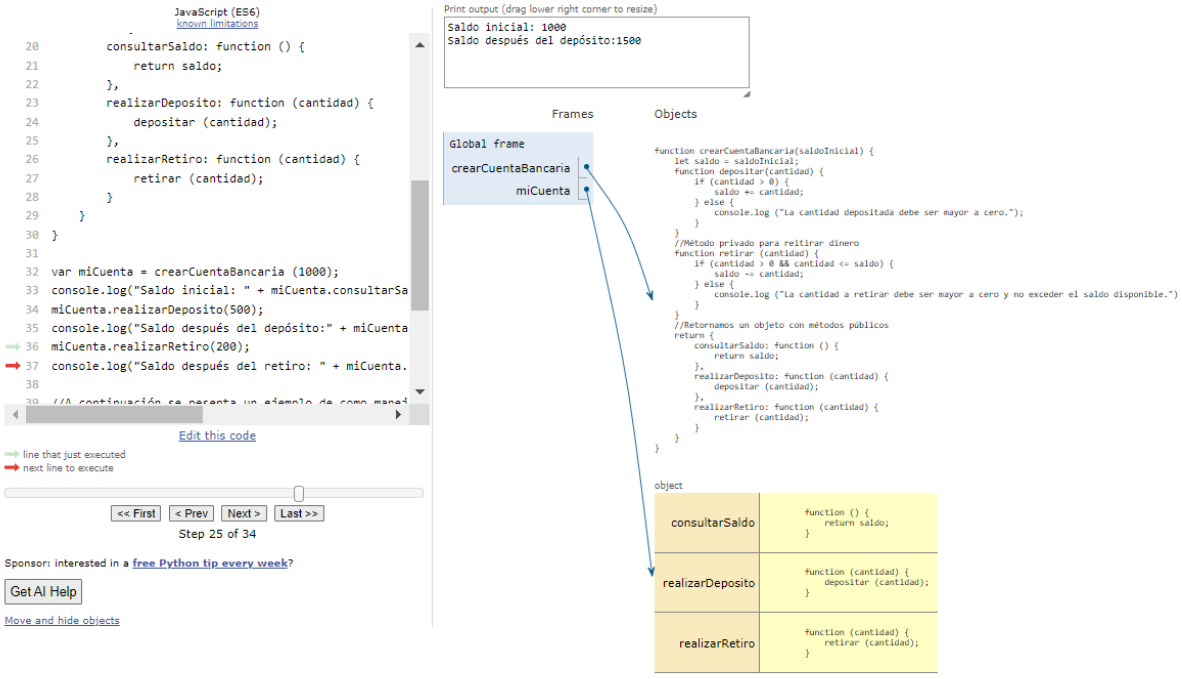


Se retira la cantidad y se imprime saldo despues de retiro

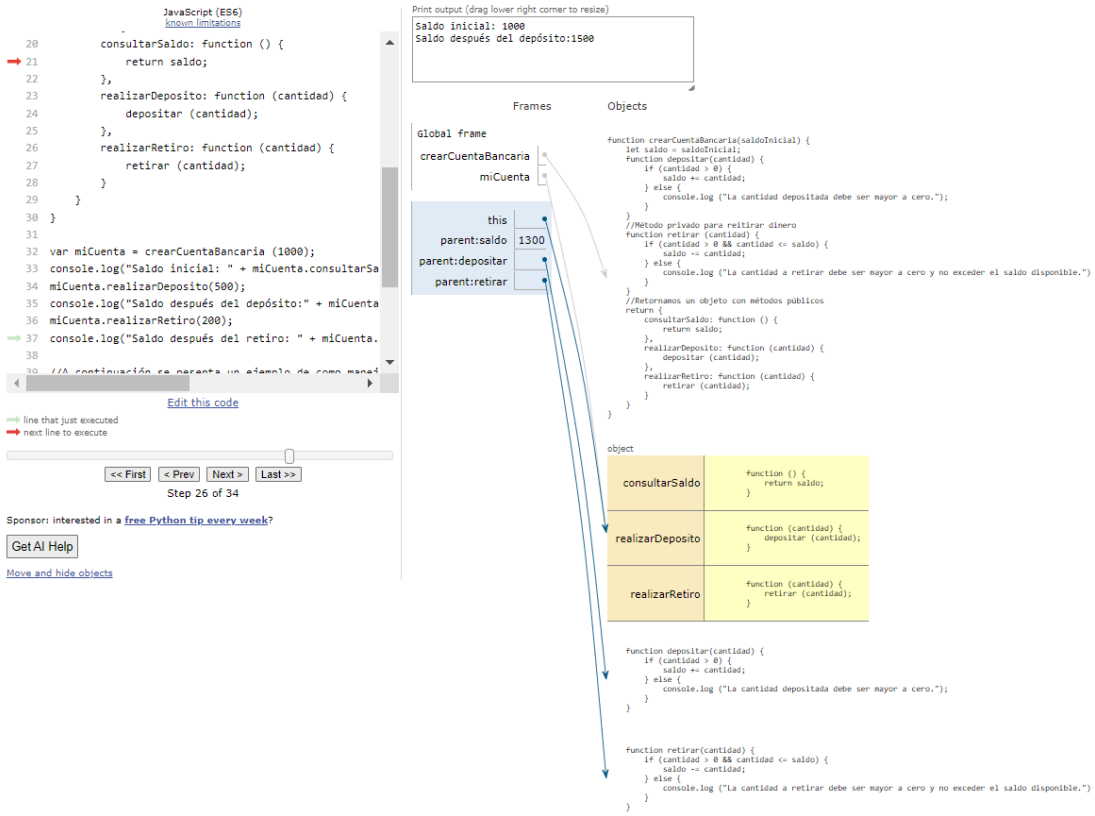
Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java



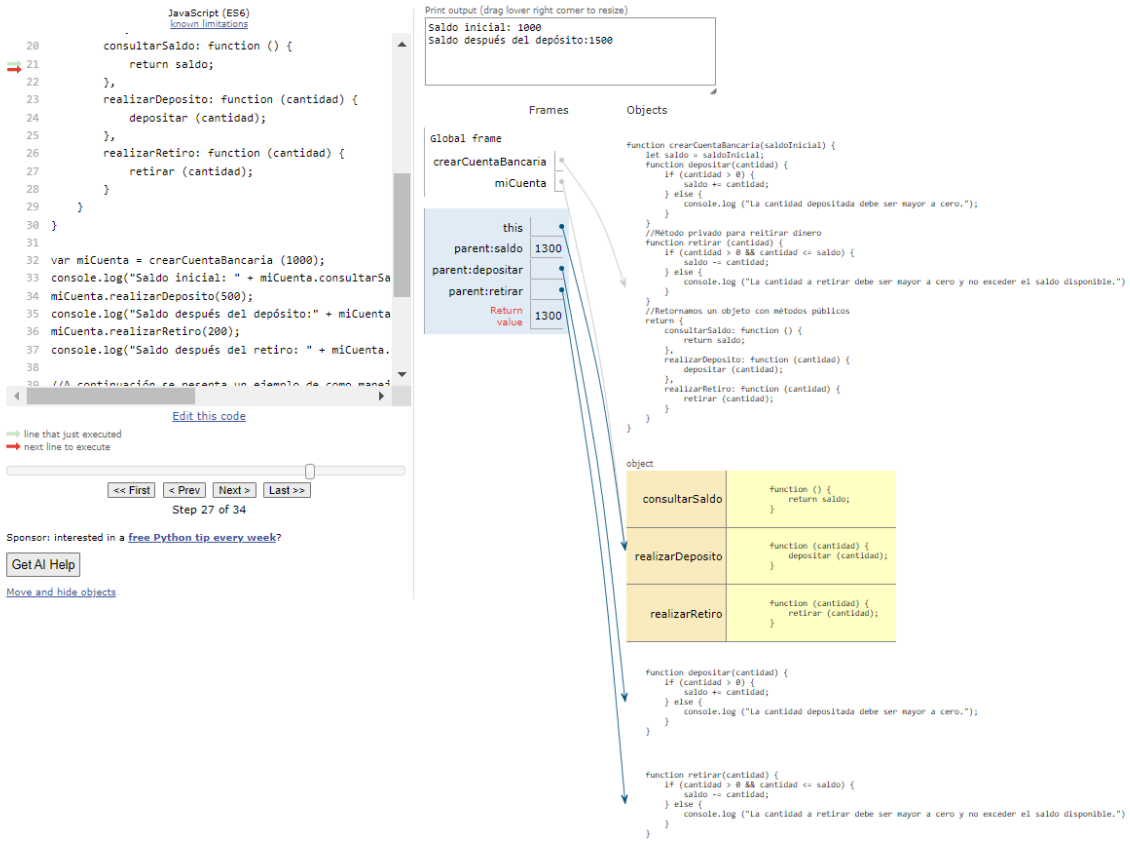
Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java



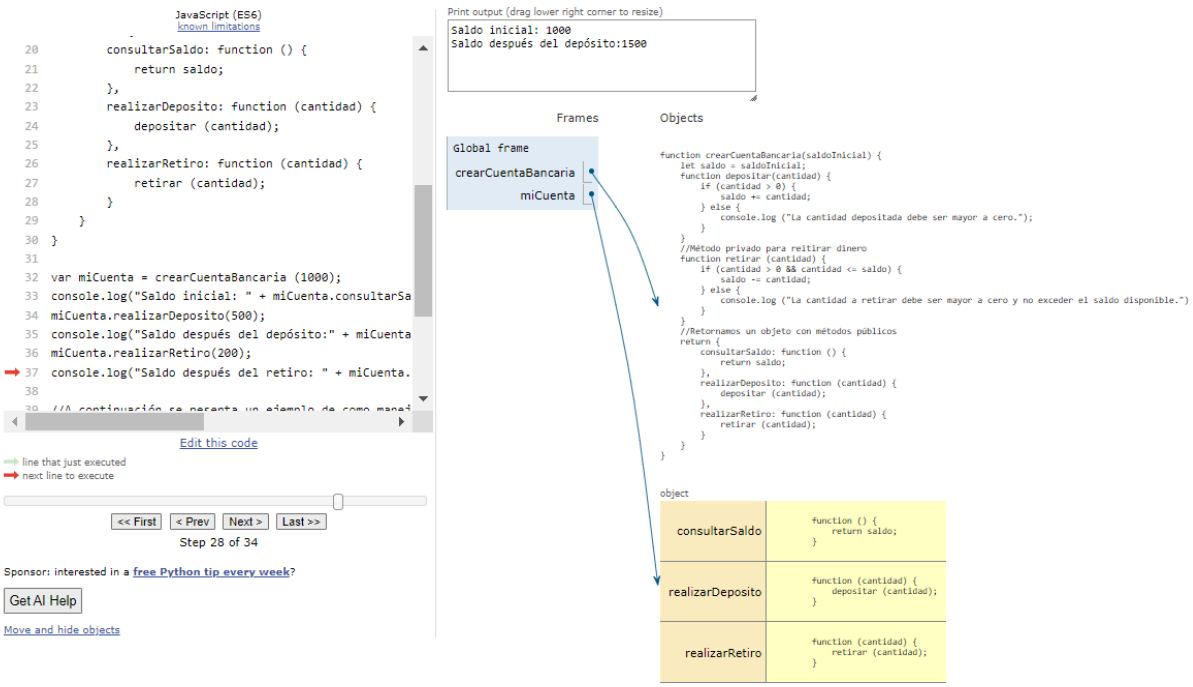
Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java



Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java



Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java

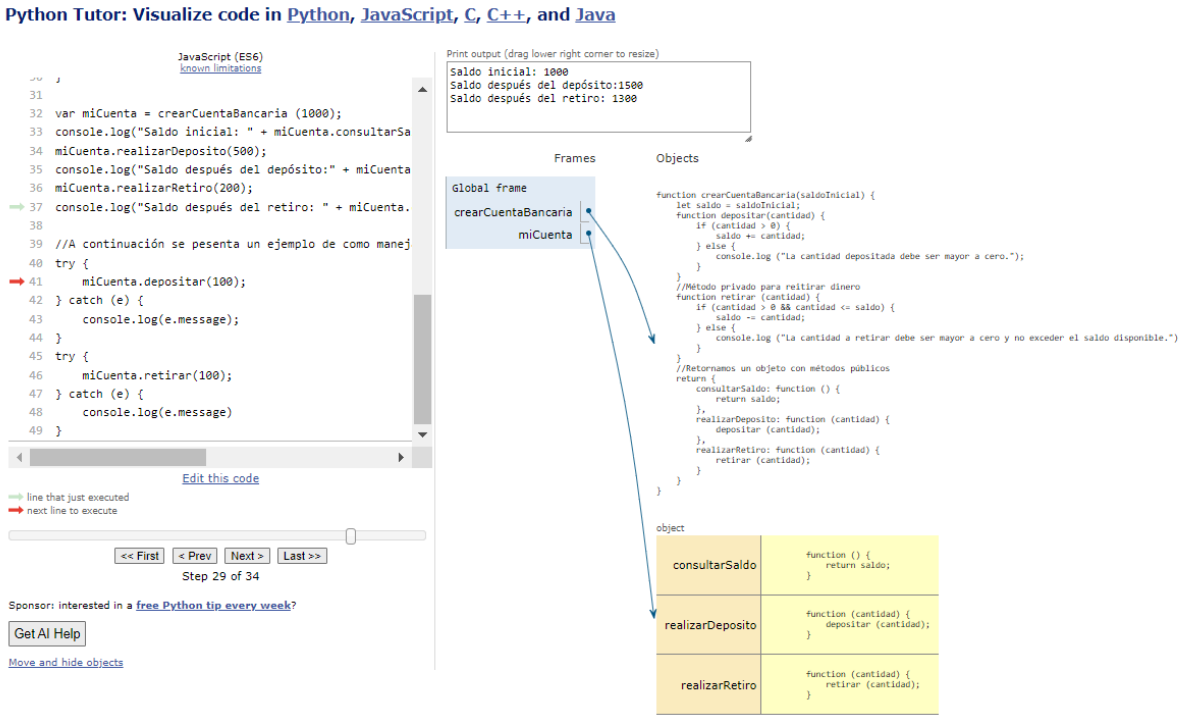


Se intenta realizar un depósito de 100 a la cuenta utilizando el método miCuenta.depositar(100).

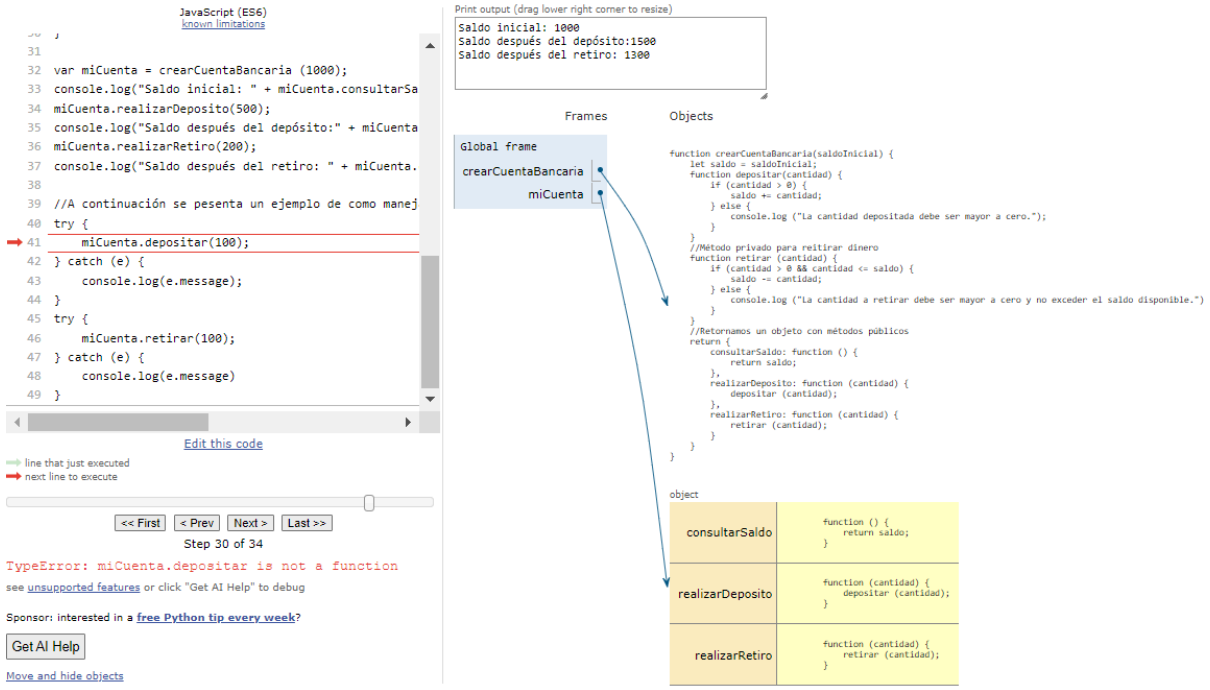
- Si la operación se realiza correctamente, el código continúa su ejecución sin entrar al bloque catch.
- Si la operación falla por una excepción (por ejemplo, la cantidad a depositar no es válida), se genera una excepción y el flujo del código salta al bloque catch.

Dentro del bloque catch, se captura la excepción (e) y se imprime su mensaje (e.message) en la consola.

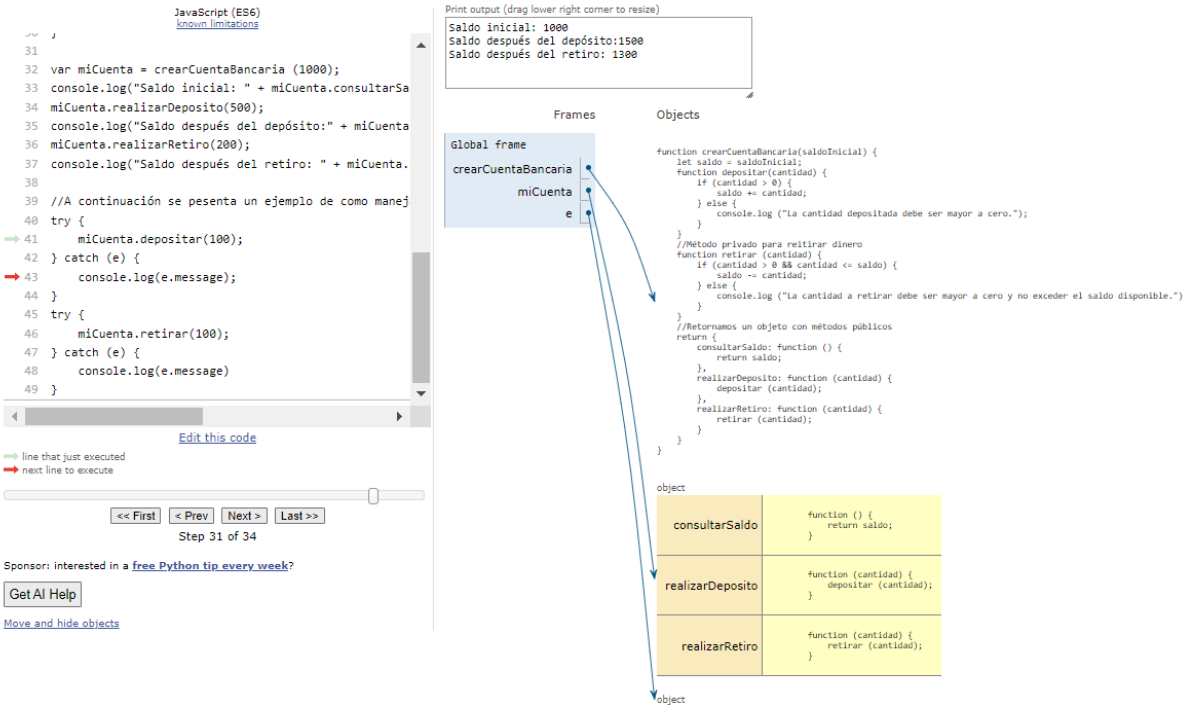
- El mensaje de error proporciona información sobre la causa de la excepción, lo que puede ser útil para depurar el código y detectar posibles errores en la lógica de las operaciones.

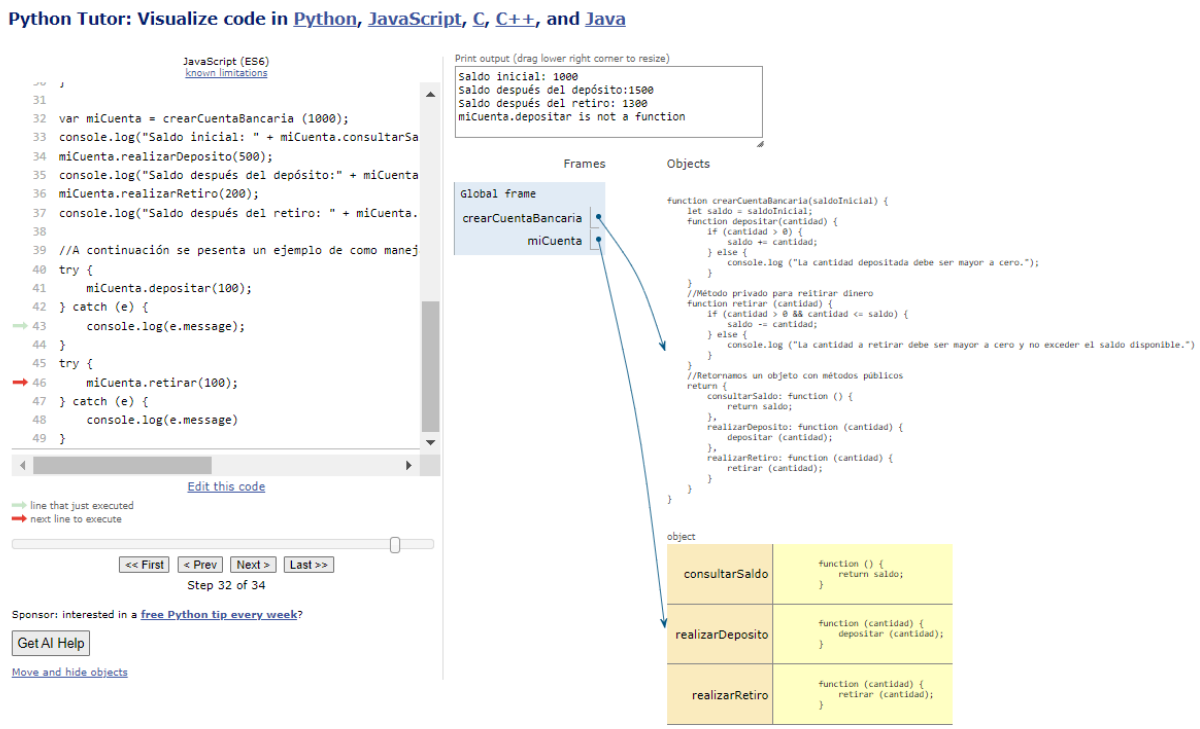


Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java



Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java





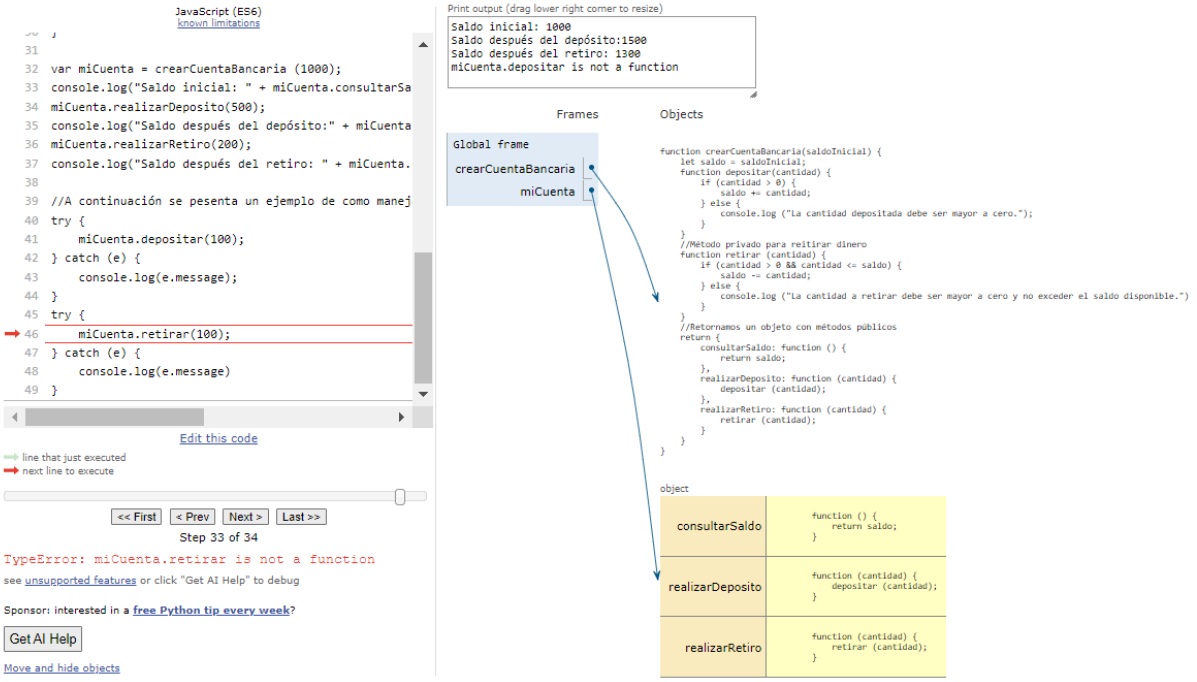
Se intenta realizar un retiro de 100 de la cuenta utilizando el método miCuenta.retirar(100).

- Similar al caso anterior, si la operación se realiza correctamente, el código continúa sin entrar al bloque catch.
- Si la operación falla por una excepción (por ejemplo, saldo insuficiente), se genera una excepción y el flujo del código salta al bloque catch.

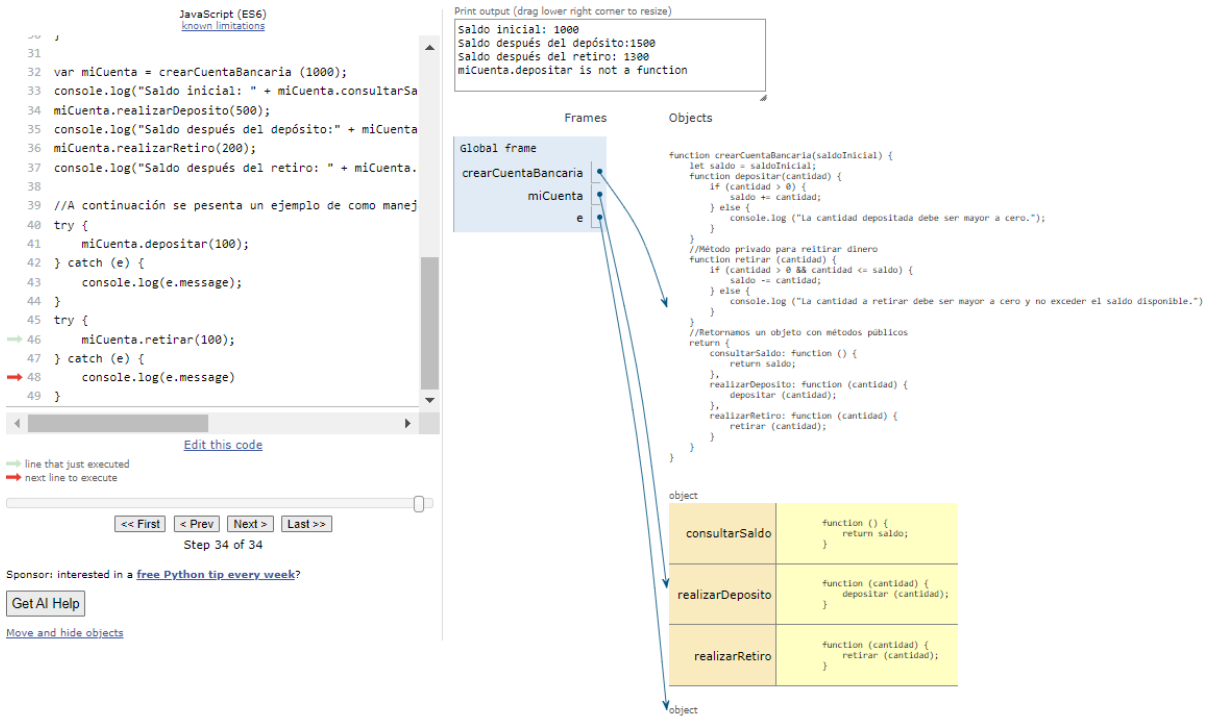
Dentro del bloque catch, se captura la excepción (e) y se imprime su mensaje (e.message) en la consola.

- Al igual que en el caso del depósito, el mensaje de error proporciona información sobre la causa de la excepción, permitiendo identificar el problema.

Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java



Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java



Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java