Tarea 08 javascript

El orden que aparece a continuación es acorde a Python tutor.

Paso 1: Declaramos una variable que será "miCuenta" y la función que es crear cuenta, entre paréntesis está el valor inicial o el string.

JavaScript (ES6)
known limitations

```
34 let miCuenta = crearCuentaBancaria (1000);
35 console.log("Saldo inicial: " + miCuenta.consultarSaldo());
36 miCuenta.realizarDeposito(500);
37
38 console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39 miCuenta.realizarRetiro(200);
40
41 console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
42
43
```

Paso 2: Declaramos una variable que llame a una función el saldo principal.

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

JavaScript (ES6)
known limitations

```
1  function crearCuentaBancaria (saldoInicial) {
2      let saldo = saldoInicial;
3      function depositar(cantidad) {
4          if (cantidad > 0) {
5              saldo += cantidad;
6          } else {
7              console.log ("La cantidad depositada debe ser mayor a cero.")
8          }
9      }
10
11 function retirar (cantidad) {
12     if (cantidad > 0 && cantidad <= saldo) {
13         saldo -= cantidad;
14         } else {
15         console.log ("La cantidad a retirar debe ser mayor a cero y no ex
16         }
17     }
18
```

Print output (drag lower right corner to resize)

Frames

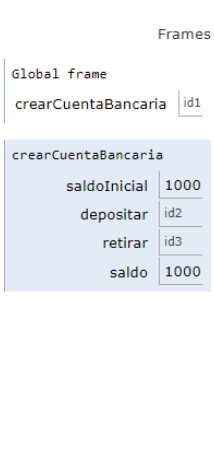
Global frame
crearCuentaBancaria | id1

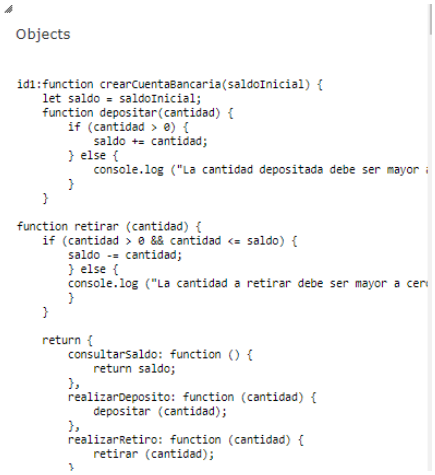
crearCuentaBancaria
saldoInicial | 1000
depositar | id2
retirar | id3

Paso 3: Este paso nos va a indicar qué debe retornar

```
11 function retirar (cantidad) {
12     if (cantidad > 0 && cantidad <= saldo) {
13         saldo -= cantidad;
14         } else {
15         console.log ("La cantidad a retirar debe ser mayor a cero y no ex
16         }
17     }
18
19     return {
20         consultarSaldo: function () {
21             return saldo;
22         },
23         realizarDeposito: function (cantidad) {
24             depositar (cantidad);
25         },
26         realizarRetiro: function (cantidad) {
27             retirar (cantidad);
28         }
```

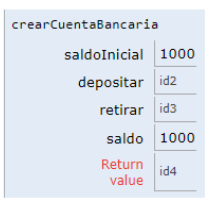
Edit this code

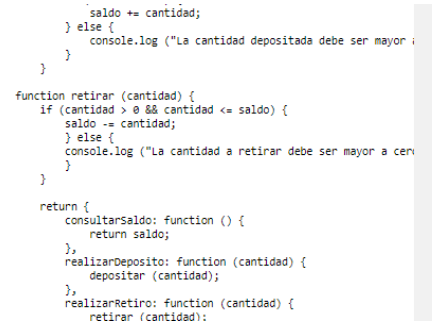




Paso 4: Después de indicarnos qué retorna está la información que va a ser requerida o la función

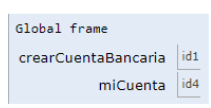
```
16         }
17     }
18
19     return {
20         consultarSaldo: function () {
21             return saldo;
22         },
23         realizarDeposito: function (cantidad) {
24             depositar (cantidad);
25         },
26         realizarRetiro: function (cantidad) {
27             retirar (cantidad);
28         }
```

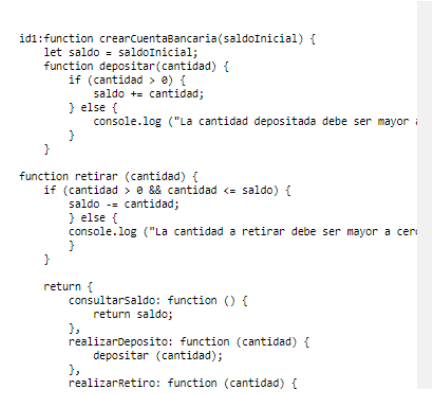




Paso 5: Aquí vamos a verificar lo que va a salir en la consola que sería el saldo inicial más el resultado de la función qué pide consultar el saldo

```
29     }
30 }
31
32
33
34 let miCuenta = crearCuentaBancaria (1000);
35 console.log("Saldo inicial: " + miCuenta.consultarSaldo());
36 miCuenta.realizarDeposito(500);
37
38 console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39 miCuenta.realizarRetiro(200);
40
41 console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
42
43
44 try {
```





Paso 6: Nos va a regresar lo que indique la función de consultar saldo

JavaScript (ES6)
known limitations

```
            console.log ("La cantidad depositada debe ser mayor a cero.")
        }
    }

function retirar (cantidad) {
    if (cantidad > 0 && cantidad <= saldo) {
        saldo -= cantidad;
        } else {
        console.log ("La cantidad a retirar debe ser mayor a cero y no ex
        }
    }

    return {
        consultarSaldo: function () {
            return saldo;
        },
```

Print output (drag lower right corner to resize)

Frames

Global frame	
crearCuentaBancaria	id1
miCuenta	id4

this	id4
parent:saldo	1000
parent:depositar	id2
parent:retirar	id3

Paso 7: Aquí la función que nos sirve para consultar el saldo ya nos dio la cantidad de 1000.

```
function retirar (cantidad) {
    if (cantidad > 0 && cantidad <= saldo) {
        saldo -= cantidad;
        } else {
        console.log ("La cantidad a retirar debe ser mayor a cero y no ex
        }
    }

    return {
        consultarSaldo: function () {
            return saldo;
        },
```

Global frame	
crearCuentaBancaria	id1
miCuenta	id4

this	id4
parent:saldo	1000
parent:depositar	id2
parent:retirar	id3
Return value	1000

Paso 8: En consola lo que vamos a ver es el saldo inicial de la variable mi cuenta más la función de consultar saldo.

```
        },
        realizarRetiro: function (cantidad) {
            retirar (cantidad);
        }
    }
}



let miCuenta = crearCuentaBancaria (1000);
console.log("Saldo inicial: " + miCuenta.consultarSaldo());
miCuenta.realizarDeposito(500);

console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
miCuenta.realizarRetiro(200);
```

Edit this code

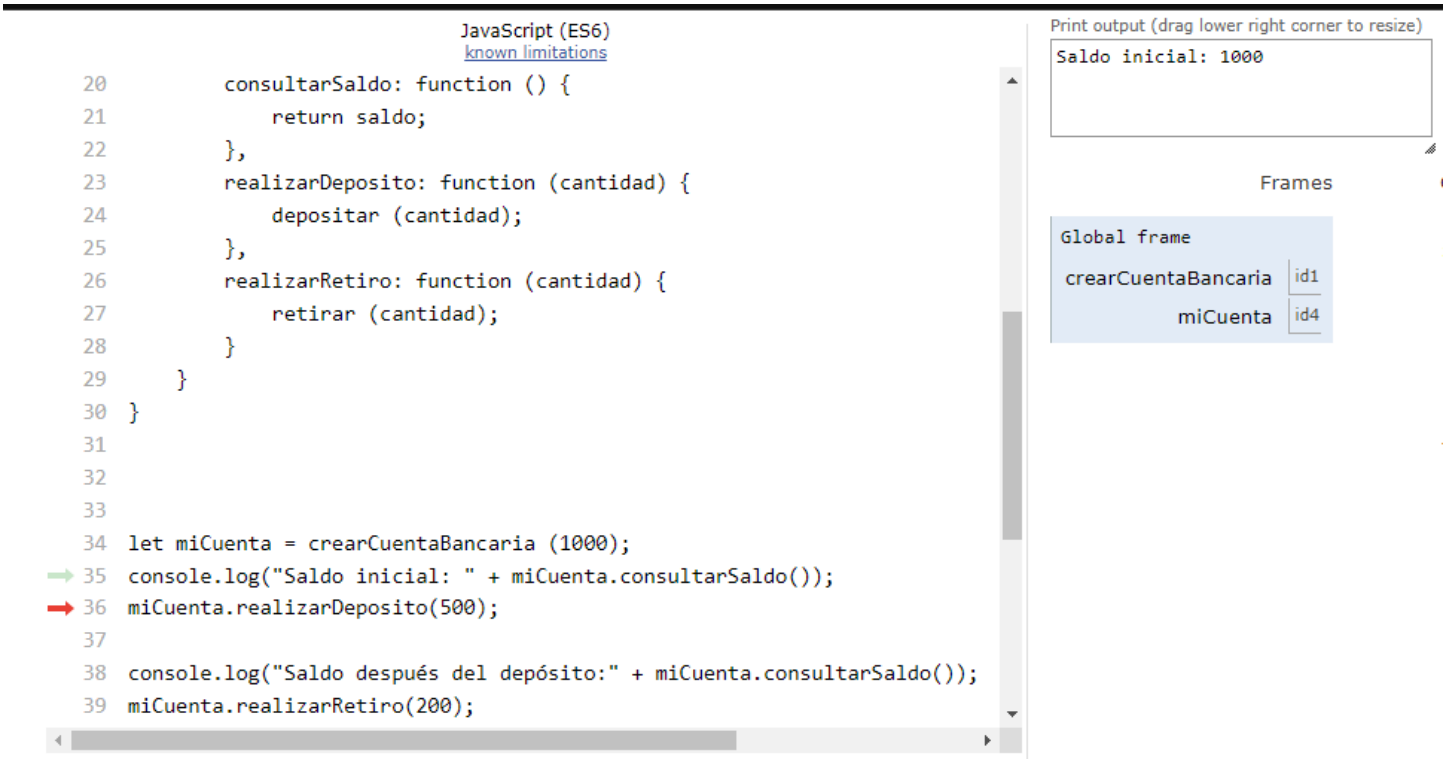
line that just executed
next line to execute

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

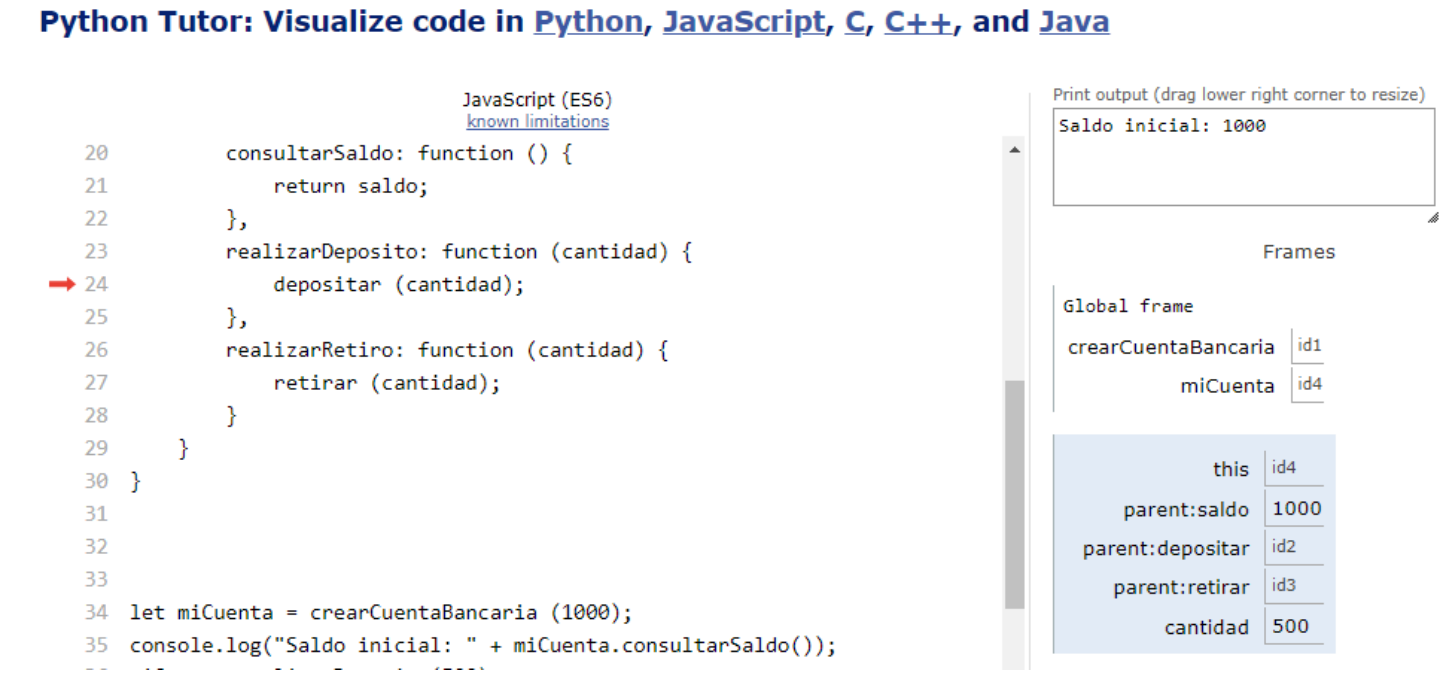
Step 8 of 34

Global frame	
crearCuentaBancaria	id1
miCuenta	id4

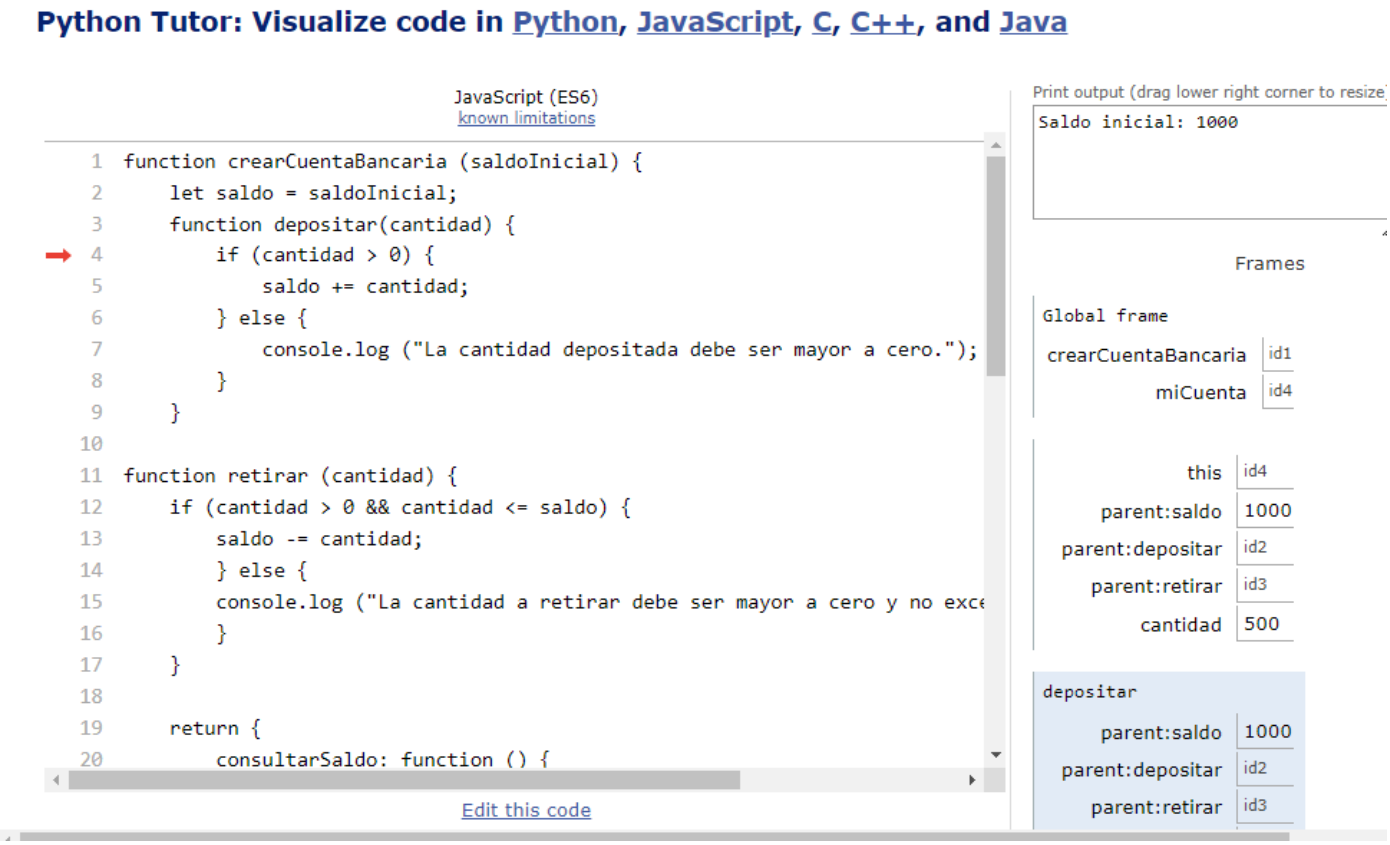
Paso 9: Ahora podemos ver que la función realizar depósito suma una cantidad de 500 a la suma actual de la cuenta que eran 1000, inicialmente.



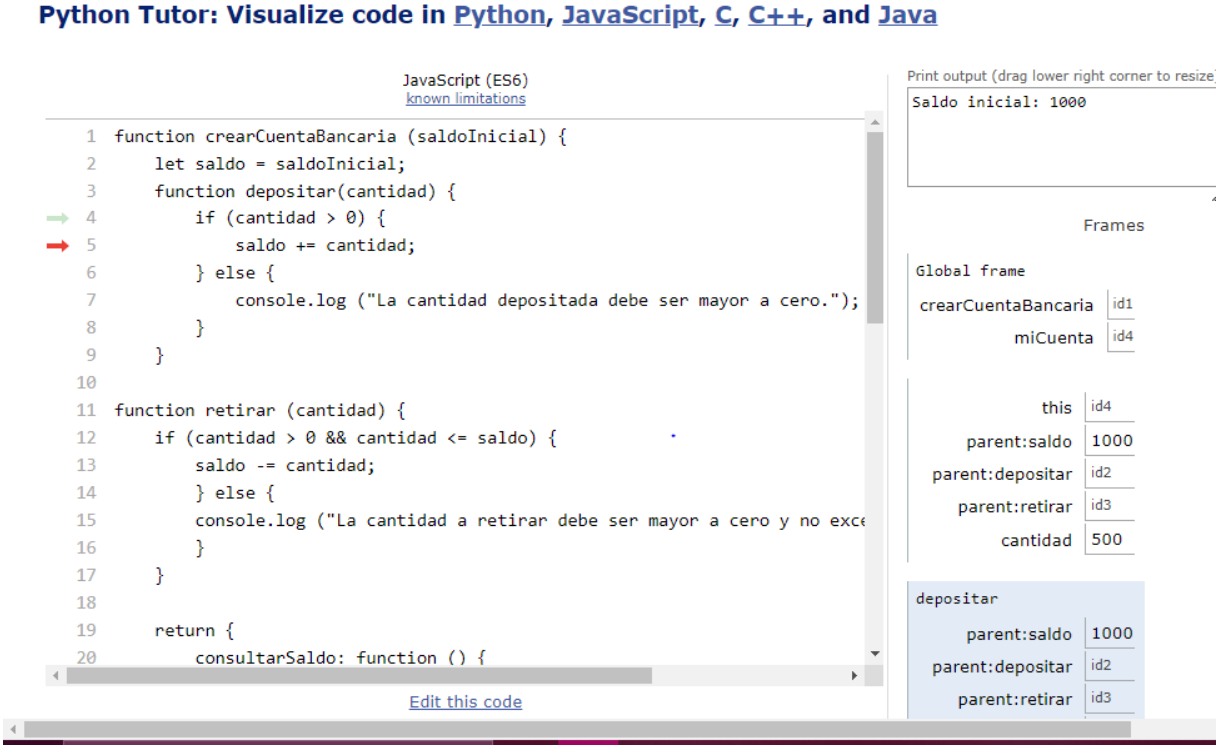
Paso 10: En este paso ya se usa la función depositar y entre paréntesis se puede ver el string que sería una cantidad.



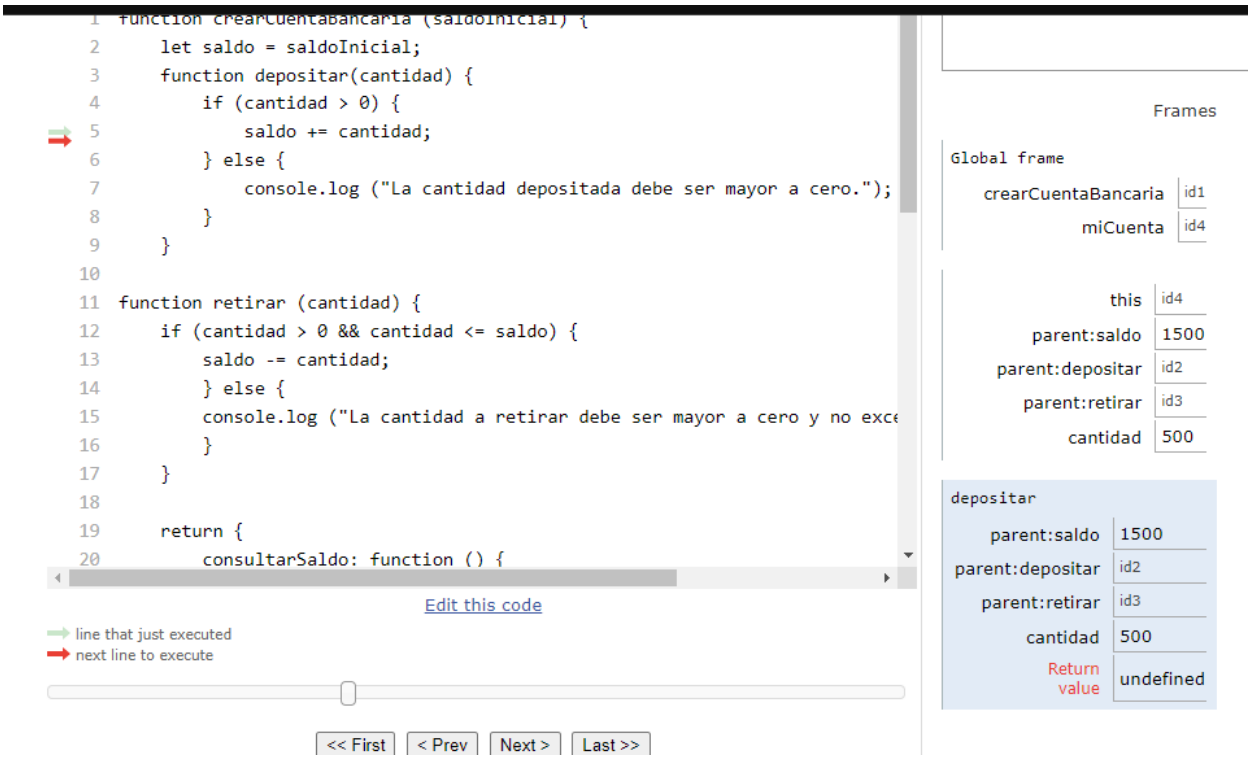
Paso 11: Aquí empieza la función de if-else, esta función indica que si la cantidad depositada es mayor a cero



Paso 12: se suma el saldo más la cantidad que se depositó

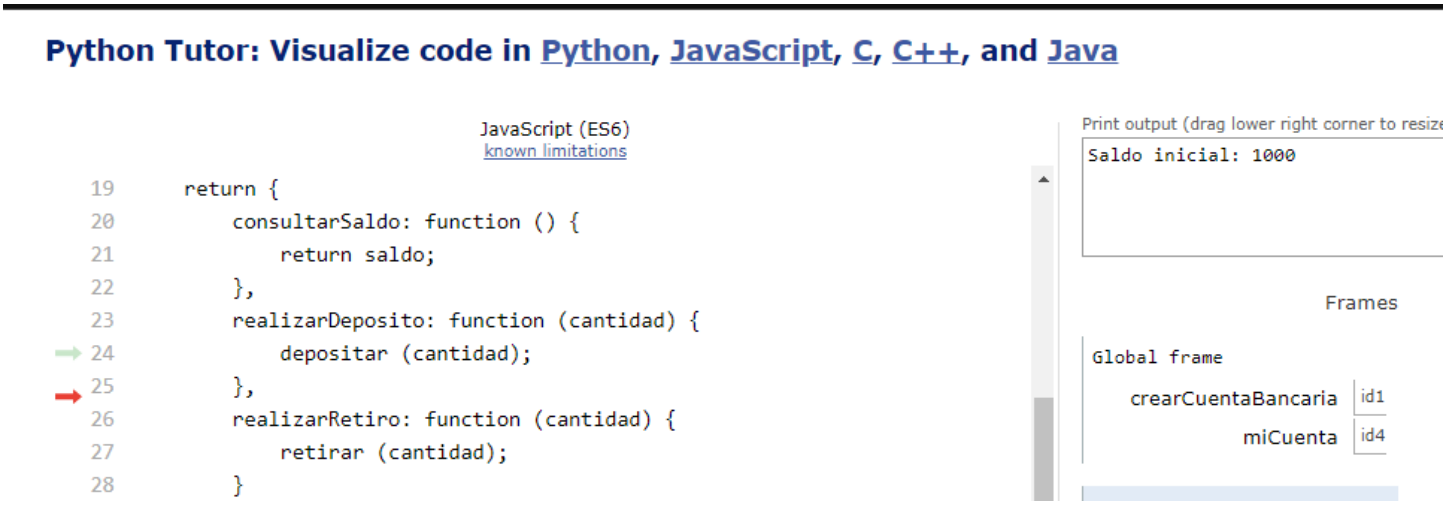


Paso 13: Si lo que se depositó es igual a cero entonces en consola nos va a aparecer el mensaje de abajo.



Paso 14:

Ahora vamos a ver la función de cuando se va a hacer un retiro que se llame como la misma acción que se va a realizar.



Paso 15: En la consola vamos a ver la cantidad después, al llamar a la función de consultar saldo

```
34 let miCuenta = crearCuentaBancaria (1000);
35 console.log("Saldo inicial: " + miCuenta.consultarSaldo());
36 miCuenta.realizarDeposito(500);
37
38 console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39 miCuenta.realizarRetiro(200);
40
41 console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
42
43
```

Paso 16: Y nos va a retornar la cantidad de 1500

```
20         consultarSaldo: function () {
21             return saldo;
22         },
23         realizarDeposito: function (cantidad) {
24             depositar (cantidad);
25         },
26         realizarRetiro: function (cantidad) {
27             retirar (cantidad);
28         }
```

this	id4
parent:saldo	1500
parent:depositar	id2
parent:retirar	id3

Paso 17: vemos la cantidad requerida anteriormente

```
18
19     return {
20         consultarSaldo: function () {
21             return saldo;
22         },
23         realizarDeposito: function (cantidad) {
24             depositar (cantidad);
25         },
26         realizarRetiro: function (cantidad) {
27             retirar (cantidad);
28         }
29     }
```

miCuenta	id4

this	id4
parent:saldo	1500
parent:depositar	id2
parent:retirar	id3
Return value	1500

Paso 18: En la consola nos va a aparecer el saldo después del depósito (como está escrito) y la variable de mi cuenta para tener la información y la función consultar saldo para que nos aparezca el monto que queda.

```
31
32
33
34 let miCuenta = crearCuentaBancaria (1000);
35 console.log("Saldo inicial: " + miCuenta.consultarSaldo());
36 miCuenta.realizarDeposito(500);
37
38 console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39 miCuenta.realizarRetiro(200);
40
41 console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
42
```

Frames

Global frame

crearCuentaBancaria	id1
miCuenta	id4

Paso 19: nos va a aparecer la cantidad de 1500, después del depósito

---

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

JavaScript (ES6)
known limitations

```
30 }
31
32
33
34 let miCuenta = crearCuentaBancaria (1000);
35 console.log("Saldo inicial: " + miCuenta.consultarSaldo());
36 miCuenta.realizarDeposito(500);
37
38 console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39 miCuenta.realizarRetiro(200);
40
41 console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
42
```

Print output (drag lower right corner to resize)

```
Saldo inicial: 1000
Saldo después del depósito:1500
```

Frames

Global frame

crearCuentaBancaria	id1
miCuenta	id4

Paso 20: por último, vamos a realizar un retiro y para eso creamos otra función que va a llamar a la string cantidad.

```
26         realizarRetiro: function (cantidad) {
27             retirar (cantidad);
28         }
29     }
30 }
31
```

Paso 21: si la cantidad es mayor a cero y la cantidad es menor o igual al saldo

```
            saldo += cantidad;
        } else {
            console.log ("La cantidad depositada debe ser mayor a cero.");
        }
    }

function retirar (cantidad) {
    if (cantidad > 0 && cantidad <= saldo) {
        saldo -= cantidad;
        } else {
        console.log ("La cantidad a retirar debe ser mayor a cero y no exce
        }
    }

    return {
        consultarSaldo: function () {
```

Edit this code

Frames

Global frame

crearCuentaBancaria	id1
miCuenta	id4

this	id4
parent:saldo	1500
parent:depositar	id2
parent:retirar	id3
cantidad	200

retirar

parent:saldo	1500
parent:depositar	id2
parent:retirar	id3

Paso 22: va a restar la cantidad que se tiene actualmente.

JavaScript (ES6)
known limitations

```
function crearCuentaBancaria (saldoInicial) {
    let saldo = saldoInicial;
    function depositar(cantidad) {
        if (cantidad > 0) {
            saldo += cantidad;
        } else {
            console.log ("La cantidad depositada debe ser mayor a cero.");
        }
    }

function retirar (cantidad) {
    if (cantidad > 0 && cantidad <= saldo) {
        saldo -= cantidad;
        } else {
        console.log ("La cantidad a retirar debe ser mayor a cero y no exce
        }
    }
```

Print output (drag lower right corner to resize)

```
Saldo inicial: 1000
Saldo después del depósito:1500
```

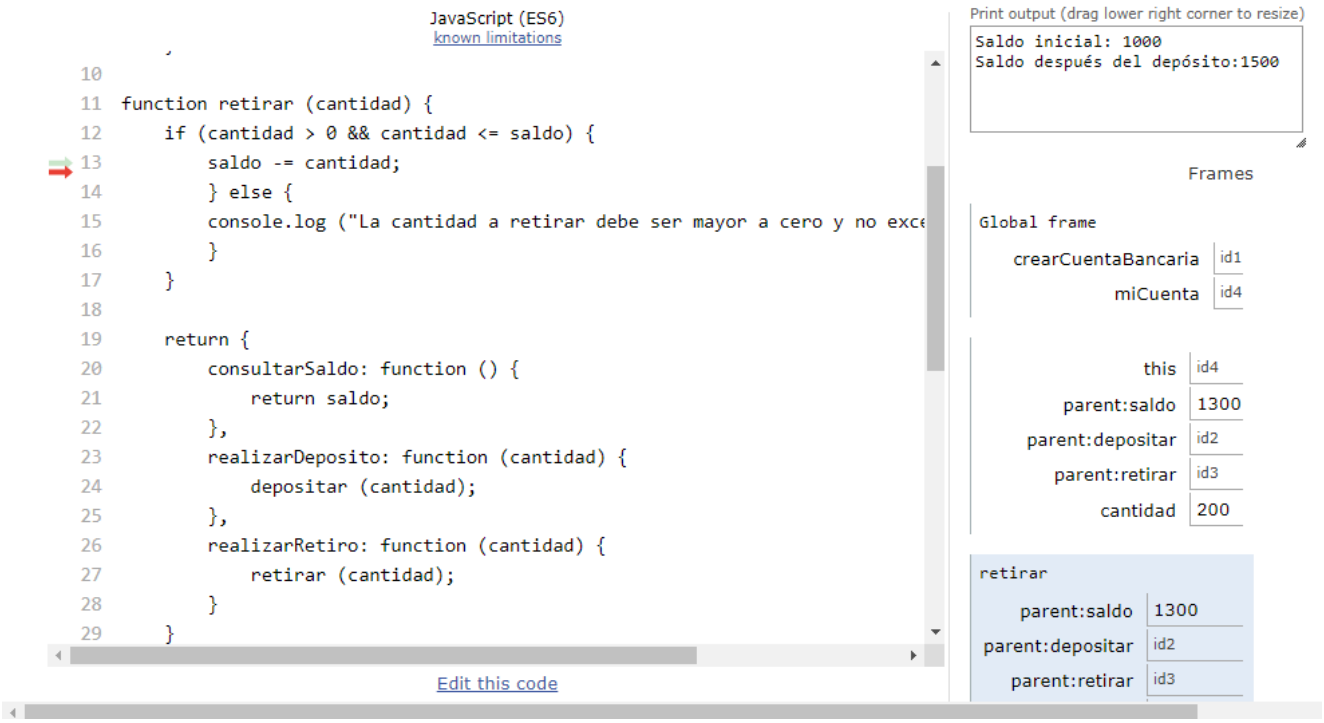
Frames

Global frame

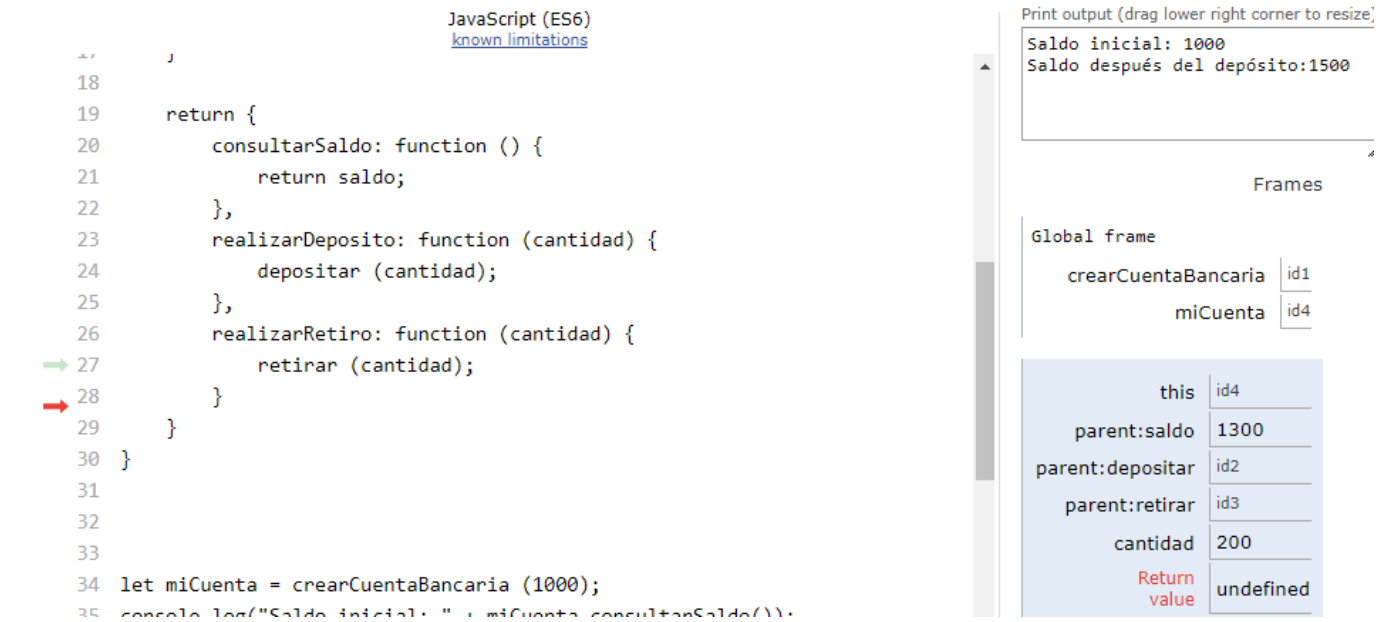
crearCuentaBancaria	id1
miCuenta	id4

this	id4
parent:saldo	1500
parent:depositar	id2
parent:retirar	id3
cantidad	200

Paso 23: de lo contrario nos indica el comando de else



Paso 24: se procede a verificar la función de retirar y el string de la cantidad para ver cuánto



Paso 25: En consola nos va a aparecer el saldo después del retiro y para ver cuánto es tenemos que tomar en cuenta la variable de "mi cuenta" y la función que es consultar saldo

```
33
34 let miCuenta = crearCuentaBancaria (1000);
35 console.log("Saldo inicial: " + miCuenta.consultarSaldo());
36 miCuenta.realizarDeposito(500);
37
38 console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39 miCuenta.realizarRetiro(200);
40
41 console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
```

crearCuentaBancaria id1
miCuenta id4

Paso 26: Nos va a retornar la información del saldo después de llamar la función para consultar saldo.

JavaScript (ES6)
known limitations

```
11 function retirar (cantidad) {
12     if (cantidad > 0 && cantidad <= saldo) {
13         saldo -= cantidad;
14         } else {
15         console.log ("La cantidad a retirar debe ser mayor a cero y no exce
16         }
17     }
18
19     return {
20         consultarSaldo: function () {
21             return saldo;
22         },
23         realizarDeposito: function (cantidad) {
24             depositar (cantidad);
25         },
```

Print output (drag lower right corner to resize)

```
Saldo inicial: 1000
Saldo después del depósito:1500
```

Frames

Global frame
crearCuentaBancaria id1
miCuenta id4

this id4
parent:saldo 1300
parent:depositar id2
parent:retirar id3

Paso 27: Vamos a ver la cantidad.

JavaScript (ES6)
known limitations

```
15         console.log ("La cantidad a retirar debe ser mayor a cero y no exce
16         }
17     }
18
19     return {
20         consultarSaldo: function () {
21             return saldo;
22         },
23         realizarDeposito: function (cantidad) {
24             depositar (cantidad);
25         },
26         realizarRetiro: function (cantidad) {
27             retirar (cantidad);
28         }
29     }
30 }
31
```

Print output (drag lower right corner to resize)

```
Saldo inicial: 1000
Saldo después del depósito:1500
```

Frames

Global frame
crearCuentaBancaria id1
miCuenta id4

this id4
parent:saldo 1300
parent:depositar id2
parent:retirar id3
Return value 1300

Paso 28: vamos a tener en consola el mensaje siempre con la variable de mi cuenta y la función de consultar saldo para ver cuánto nos queda.

```
29      }
30  }
31
32
33
34  let miCuenta = crearCuentaBancaria (1000);
35  console.log("Saldo inicial: " + miCuenta.consultarSaldo());
36  miCuenta.realizarDeposito(500);
37
38  console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39  miCuenta.realizarRetiro(200);
40
41  console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
42
```

Global frame
crearCuentaBancaria id1
miCuenta id4

Paso 29: Vamos a verificar que son 1300 de la variable que es mi cuenta.

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

JavaScript (ES6)
known limitations

```
29      }
30  }
31
32
33
34  let miCuenta = crearCuentaBancaria (1000);
35  console.log("Saldo inicial: " + miCuenta.consultarSaldo());
36  miCuenta.realizarDeposito(500);
37
38  console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39  miCuenta.realizarRetiro(200);
40
41  console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
42
43
44  try {
45      miCuenta.depositar(100);
46  } catch (e) {
47      console.log(e.message);
48  }
```

Print output (drag lower right corner to resize)

```
Saldo inicial: 1000
Saldo después del depósito:1500
Saldo después del retiro: 1300
```

Frames

Global frame
crearCuentaBancaria id1
miCuenta id4

Paso 30: Vamos a depositar el string en el objeto que es "mi cuenta".

```
38 console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39 miCuenta.realizarRetiro(200);
40
41 console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
42
43
44 try {
45     miCuenta.depositar(100);
46 } catch (e) {
47     console.log(e.message);
48 }
49 try {
50     miCuenta.retirar(100);
51 } catch (e) {
52     console.log(e.message)
```

Global frame
crearCuentaBancaria id1
miCuenta id4

Edit this code

line that just executed
next line to execute

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

Step 30 of 34

TypeError: miCuenta.depositar is not a function

see unsupported features or click "Get AI Help" to debug

Paso 31: nos va a aparecer un mensaje.

JavaScript (ES6)
known limitations

```
33
34 let miCuenta = crearCuentaBancaria (1000);
35 console.log("Saldo inicial: " + miCuenta.consultarSaldo());
36 miCuenta.realizarDeposito(500);
37
38 console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39 miCuenta.realizarRetiro(200);
40
41 console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
42
43
44 try {
45     miCuenta.depositar(100);
46 } catch (e) {
47     console.log(e.message);
48 }
```

Print output (drag lower right corner to resize)

```
Saldo inicial: 1000
Saldo después del depósito:1500
Saldo después del retiro: 1300
```

Frames

Global frame
crearCuentaBancaria id1
miCuenta id4
e id8

Paso 32: Vamos a indicar de dónde vamos a ejecutar la función retirar

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

JavaScript (ES6)
known limitations

```
34 let miCuenta = crearCuentaBancaria (1000);
35 console.log("Saldo inicial: " + miCuenta.consultarSaldo());
36 miCuenta.realizarDeposito(500);
37
38 console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39 miCuenta.realizarRetiro(200);
40
41 console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
42
43
44 try {
45     miCuenta.depositar(100);
46 } catch (e) {
47     console.log(e.message);
48 }
49 try {
50     miCuenta.retirar(100);
51 } catch (e) {
52     console.log(e.message)
53 }
```

Print output (drag lower right corner to resize)

```
Saldo inicial: 1000
Saldo después del depósito:1500
Saldo después del retiro: 1300
miCuenta.depositar is not a function
```

Frames

Global frame

crearCuentaBancaria	id1
miCuenta	id4

Paso 33: No nos aparece porque la información es privada.

```
46 } catch (e) {
47     console.log(e.message);
48 }
49 try {
50     miCuenta.retirar(100);
51 } catch (e) {
52     console.log(e.message)
53 }
```

Edit this code

line that just executed
next line to execute

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

Step 33 of 34

TypeError: miCuenta.retirar is not a function

see unsupported features or click "Get AI Help" to debug

Paso 34: nos tendría que aparecer el mensaje después del retiro pero no lo vemos porque es información privada.

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

JavaScript (ES6)
known limitations

```
34 let miCuenta = crearCuentaBancaria (1000);
35 console.log("Saldo inicial: " + miCuenta.consultarSaldo());
36 miCuenta.realizarDeposito(500);
37
38 console.log("Saldo después del depósito:" + miCuenta.consultarSaldo());
39 miCuenta.realizarRetiro(200);
40
41 console.log("Saldo después del retiro: " + miCuenta.consultarSaldo());
42
43
44 try {
45     miCuenta.depositar(100);
46 } catch (e) {
47     console.log(e.message);
48 }
49 try {
50     miCuenta.retirar(100);
51 } catch (e) {
52     console.log(e.message)
53 }
```

Edit this code

Print output (drag lower right corner to resize)

```
Saldo inicial: 1000
Saldo después del depósito:1500
Saldo después del retiro: 1300
miCuenta.depositar is not a function
```

Frames

Global frame

crearCuentaBancaria	id1
miCuenta	id4
e	id9