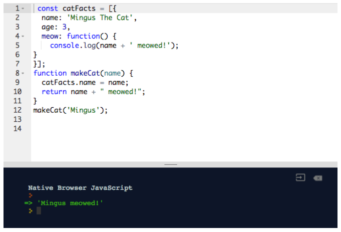
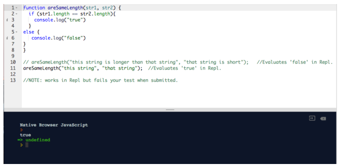
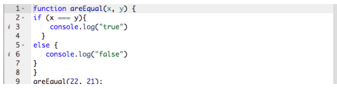
All Issues Code

Monday, October 2, 2017 4:45 PM

Problem Name	Status	Issue	Alerts	Code
makeCat	askHelp	Works in the repl but does not pass. 	failed should_create_a_new_cat_with_a_name_property should_add_a_method_called_meow_to_the_new_cat_object	const c name age: 3 meow cons } }]; functio catFa retur } makeC

areSameLength	askHelp	//Evaluates 'true' in Repl.
		
		//NOTE: works in Repl but fails your test when submitted.
		
			failed
			should_return_true_if_the_arguments_have_the_same_length
				functio
				if (str
				cons
				}
				else {
				cons
				}
				}
				
				// areS
				//Evalu
				areSam

areEqual	askHelp	//NOTE: works in Repl but fails your test when submitted.
		
			failed
			should_return_true_if_the_arguments_are_equal
				functio
				if (x ==
				cons
				}
				else {
				cons

```
const catFacts = [{
  name: 'Mingus The Cat',
  age: 3,
  meow: function() {
    console.log(name + ' meowed!');
}
}];
function makeCat(name) {
  catFacts.name = name;
  return name + " meowed!";
}
makeCat('Mingus');
```

```
Native Browser JavaScript
>
=> 'Mingus meowed!'
>
```

```
function areSameLength(str1, str2) {
  if (str1.length == str2.length){
    console.log("true")
  }
else {
  console.log("false")
}
}

// areSameLength("this string is longer than that string", "that string is short");   //Evaluates 'false' in Repl.
areSameLength("this string", "that string");  //Evaluates 'true' in Repl.

//NOTE: works in Repl but fails your test when submitted.
```

```
Native Browser JavaScript
>
true
=> undefined
>
```

```
function areEqual(x, y) {
if (x === y){
    console.log("true")
  }
else {
  console.log("false")
}
}
areEqual(22, 21);
```

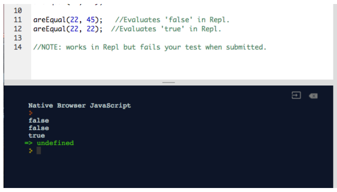
	Comments	Solution	A

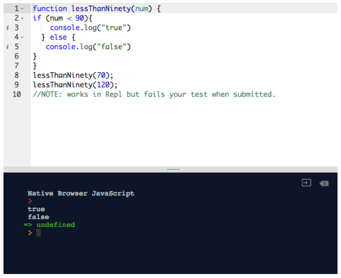
catFacts = [{
e: 'Mingus The Cat',
3,
w: function() {
sole.log(name + ' meowed!');
 	
on makeCat(name) {
cts.name = name;
n name + " meowed!";
 	
Cat('Mingus');			

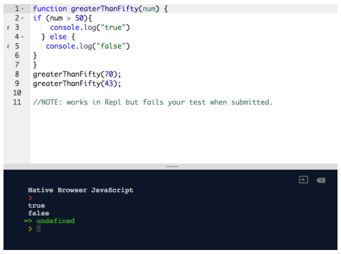
on areSameLength(str1, str2) {
r1.length == str2.length){
sole.log("true")
 	
ole.log("false")
 	
SameLength("this string is longer than that string", "that string is short");
uates 'false' in Repl.
meLength("this string", "that string");			

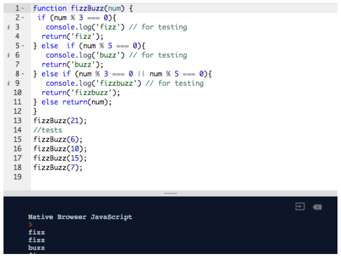
on areEqual(x, y) {
= y){
sole.log("true")
 	
ole.log("false")			

Analysis

		
		`areEqual(22, 45);   //Evaluates 'false' in Repl.`
		`areEqual(22, 22);  //Evaluates 'true' in Repl.`
		
		`//NOTE: works in Repl but fails your test when submitted.`
		
		`Native Browser JavaScript`
		`>`
		`false`
		`false`
		`true`
		`=> undefined`
		`>`
				cons
				}
				}
				areEqu
				
				areEqu
				areEqu

lessThanNinety	askHelp	//NOTE: works in Repl but fails your test when submitted.
		
		`function lessThanNinety(num) {`
		`if (num < 90){`
		`    console.log("true")`
		`  } else {`
		`   console.log("false")`
		`}`
		`}`
		`lessThanNinety(70);`
		`lessThanNinety(120);`
		`//NOTE: works in Repl but fails your test when submitted.`
		
		`Native Browser JavaScript`
		`>`
		`true`
		`false`
		`=> undefined`
		`>`
			failed should_return_true_if_the_argument_is_less_than_ninety	functio
				if (nu
				cons
				} else
				cons
				}
				}
				lessTha
				lessTha

greaterThanFifty	askHelp	//NOTE: works in Repl but fails your test when submitted.
		
		`function greaterThanFifty(num) {`
		`if (num > 50){`
		`    console.log("true")`
		`  } else {`
		`   console.log("false")`
		`}`
		`}`
		`greaterThanFifty(70);`
		`greaterThanFifty(43);`
		
		`//NOTE: works in Repl but fails your test when submitted.`
		
		`Native Browser JavaScript`
		`>`
		`true`
		`false`
		`=> undefined`
		`>`
			should_return_true_if_the_argument_is_greater_than_fifty	functio
				if (nu
				cons
				} else
				cons
				}
				}
				greate
				greate

fizzBuzz	askHelp	works in repl fails test
		
		`function fizzBuzz(num) {`
		` if (num % 3 === 0){`
		`   console.log('fizz') // for testing`
		`  return('fizz');`
		`} else  if (num % 5 === 0){`
		`   console.log('buzz') // for testing`
		`  return('buzz');`
		`} else if (num % 3 === 0 || num % 5 === 0){`
		`   console.log('fizzbuzz') // for testing`
		`  return('fizzbuzz');`
		`} else return(num);`
		`}`
		`fizzBuzz(21);`
		`//tests`
		`fizzBuzz(6);`
		`fizzBuzz(10);`
		`fizzBuzz(15);`
		`fizzBuzz(7);`
		
		`Native Browser JavaScript`
		`>`
		`fizz`
		`fizz`
		`buzz`
			should_return_fizz_if_divisible_by_3
			should_return_buzz_if_divisible_by_5
			should_return_fizzbuzz_if_divisible_by_3_and_5
				functio
				if (nu
				cons
				retur
				} else
				cons
				retur

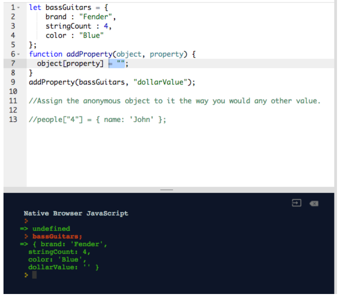
```
ole.log("false")


ual(22, 21);

ual(22, 45);   //Evaluates 'false' in Repl.
ual(22, 22);  //Evaluates 'true' in Repl.
```

```
on lessThanNinety(num) {
n < 90){
sole.log("true")
   {
ole.log("false")



anNinety(70);
anNinety(120);
```

```
on greaterThanFifty(num) {
n > 50){
sole.log("true")
   {
ole.log("false")



rThanFifty(70);
rThanFifty(43);
```

```
on fizzBuzz(num) {
m % 3 === 0){
ole.log('fizz') // for testing
n('fizz');
if (num % 5 === 0){
ole.log('buzz') // for testing
n('buzz');
f (num % 3 === 0 && num % 5 === 0){
```

		fizz
		=> 7
		>
			should_return_num_if_not_divisible_by_3_or_5	} else i
				cons
				retur
				} else r
				}
				fizzBu
				//tests
				fizzBu
				fizzBu
				fizzBu
				fizzBu

addProperty	askHelp	works in repl but fails test
		
		let bassGuitars = {
		brand : "Fender",
		stringCount : 4,
		color : "Blue"
		};
		function addProperty(object, property) {
		object[property] = "";
		}
		addProperty(bassGuitars, "dollarValue");
		
		//Assign the anonymous object to it the way you would any other value.
		
		//people["4"] = { name: 'John' };
		Native Browser JavaScript
		>
		=> undefined
		> bassGuitars;
		=> { brand: 'Fender',
		stringCount: 4,
		color: 'Blue',
		dollarValue: '' }
		>
			failed should_add_the_property_to_the_object_with_a_value_of_null	let bas
				bran
				strin
				colo
				};
				functio
				bassG
				}
				addPro
				
				//Assig
				
				//peop

invokeMethod	askHelp	lost on this	failed should_invoke_the_method_on_the_object What's Wrong? TypeError: Cannot read property 'playNote' of undefined at invokeMethod (eval at n.evaluate (https://replit.org/public/replbox_javascript.9411632bd490dec2b542.bundle.js:1:4058), <anonymous>:9:9) at eval (eval at n.evaluate (https://replit.org/public/replbox_javascript.9411632bd490dec2b542.bun	let not {note playN conso } }] functio notes } invoke

<table>
<tr><td>f (num % 3 === 0 || num % 5 === 0){
ole.log('fizzbuzz') // for testing
n('fizzbuzz');
return(num);

zz(21);
s
zz(6);
zz(10);
zz(15);
zz(7);</td><td></td><td></td><td></td></tr>
<tr><td>ssGuitars = {
nd : "Fender",
gCount : 4,
r : "Blue"

on addProperty(object, property) {
Guitars["value"] = "";

operty(bassGuitars, "value");

gn the anonymous object to it the way you would any other value.

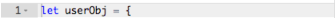
ple["4"] = { name: 'John' };</td><td></td><td></td><td></td></tr>
<tr><td>tes = [
esAvailable : ['A','Bb','B','C','C#','D','Eb','E','F'],
ote : function(note){
ole.log('I just played a ' + note);

on invokeMethod(notes, method) {
s.playNote(note);

eMethod(notes.playNote,'Bb');</td><td></td><td></td><td></td></tr>
</table>

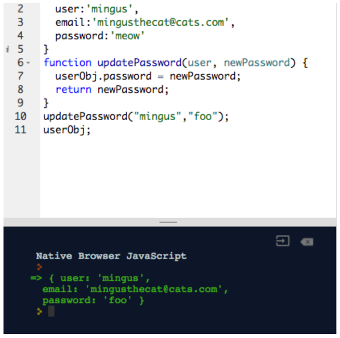
			ascript.9411632bd490dec2b542.bundle.js:1:4058), <anonymous>:11:1) at eval (<anonymous>) at n.evaluate (https://replit.org/public/replbox_javascript.9411632bd490dec2b542.bundle.js:1:4058) at n.<anonymous> (https://replit.org/public/replbox_javascript.9411632bd490dec2b542.bundle.js:1:2136) at n.emit (https://replit.org/public/replbox_javascript.9411632bd490dec2b542.bundle.js:1:9782) at r (https://replit.org/public/replbox_javascript.9411632bd490dec2b542.bundle.js:1:4474) at Object._custom_runSingleUnitTests (https://replit.org/public/replbox_javascript.9411632bd490dec2b542.bundle.js:1:5796) at https://replit.org/public/secure/runner.js:1:1323	

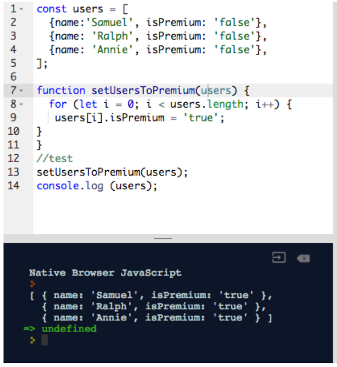
createObject	askHelp	??	Failed tests
			should_return_an_object_with_name_age_and_hobbies_properties
			should have the required properties with the required value types
			RangeError: Maximum call stack size exceeded
			More info
				functi
				 const
				  nam
				  age:
				  hob
				 };
				 consc
				 retur
				
				}
				//tests

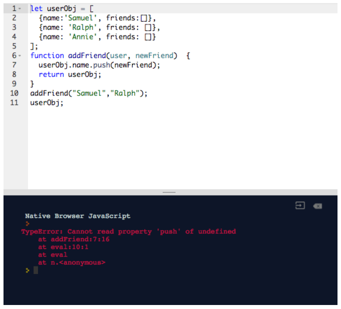
updatePassword	askHelp	Password updates in repl but fails test  let userObj = {	failed should_return_the_user_object_with_the_updated_password	let use user: email

on createObject() {
 myObject = {
ne: 'David',
 46,
bies: ['music', 'camping','boating']
 	
ole.log(createObject());
n myObject;
 	
s			

erObj = { 'mingus', l:'mingusthecat@cats.com', word:'meow'			

		
		user:'mingus',
		email:'mingusthecat@cats.com',
		password:'meow'
		}
		function updatePassword(user, newPassword) {
		userObj.password = newPassword;
		return newPassword;
		}
		updatePassword("mingus","foo");
		userObj;
		Native Browser JavaScript
		>
		=> { user: 'mingus',
		email: 'mingusthecat@cats.com',
		password: 'foo' }
		>
				passw
				}
				functio
				userO
				retur
				}
				update
				userOl

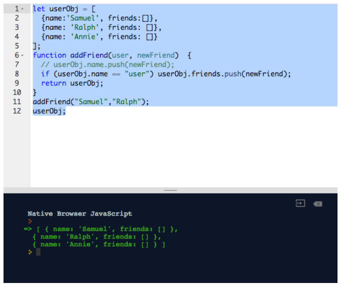
setUsersToPremium	askHelp	Code works but fails test
		
		const users = [
		{name:'Samuel', isPremium: 'false'},
		{name: 'Ralph', isPremium: 'false'},
		{name: 'Annie', isPremium: 'false'},
		];
		
		function setUsersToPremium(users) {
		for (let i = 0; i < users.length; i++) {
		users[i].isPremium = 'true';
		}
		}
		//test
		setUsersToPremium(users);
		console.log (users);
		Native Browser JavaScript
		>
		[{ name: 'Samuel', isPremium: 'true' },
		{ name: 'Ralph', isPremium: 'true' },
		{ name: 'Annie', isPremium: 'true' }]
		=> undefined
		>
			failed should_return_the_users_array_with_each_users_isPremium_property_set_to_true	const u
				{nam
				{nam
				{nam
				];
				
				functio
				for (le
				users
				}
				}
				//test
				setUse
				consol

addFriend	tryAgain	Cannot read property 'push' of undefined
		
		let userObj = [
		{name:'Samuel', friends:[]},
		{name: 'Ralph', friends: []},
		{name: 'Annie', friends: []}
		];
		function addFriend(user, newFriend) {
		userObj.name.push(newFriend);
		return userObj;
		}
		addFriend("Samuel","Ralph");
		userObj;
		Native Browser JavaScript
		>
		TypeError: Cannot read property 'push' of undefined
		at addFriend:7:16
		at eval:10:1
		at eval
		at n.<anonymous>
		>
		
		failed should_add_a_new_friend_to_the_end_of_the_friends_array_propert
				let use
				{nam
				{nam
				{nam
				];
				functio
				// use
				if (us
				retur
				}
				addFri
				userOl

```
word: meow

on updatePassword(user, newPassword) {
Obj.password = newPassword;
n newPassword;

ePassword("mingus","foo");
bj;
```

```
users = [
e:'Samuel', isPremium: 'false'},
e: 'Ralph', isPremium: 'false'},
e: 'Annie', isPremium: 'false'},


on setUsersToPremium(users) {
et i = 0; i < users.length; i++) {
s[i].isPremium = 'true';



ersToPremium(users);
le.log (users);
```

```
erObj = [
e:'Samuel', friends:[]},
e: 'Ralph', friends: []},
e: 'Annie', friends: []}

on addFriend(user, newFriend)  {
erObj.name.push(newFriend);
erObj.name == "user") userObj.friends.push(newFriend);
n userObj;

iend("Samuel","Ralph");
bj;
```

<table>
<tr><td></td><td></td><td>y

<pre>
let userObj = [
  {name:'Samuel', friends:[]},
  {name: 'Ralph', friends: []},
  {name: 'Annie', friends: []}
];
function addFriend(user, newFriend) {
  // userObj.name.push(newFriend);
  if (userObj.name == "user") userObj.friends.push(newFriend);
  return userObj;
}
addFriend("Samuel","Ralph");
userObj;
</pre>
<pre>
Native Browser JavaScript
 >
=> [{ name: 'Samuel', friends: [] },
  { name: 'Ralph', friends: [] },
  { name: 'Annie', friends: [] }]
 >
</pre></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>
