Flavours of Arrow Functions

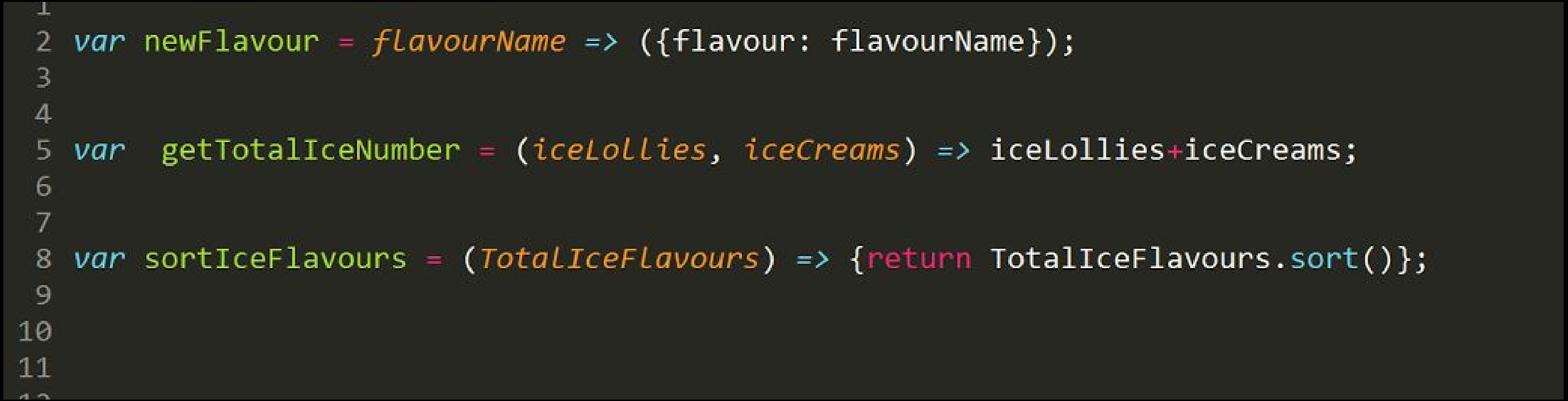
ES5 vs ES6

By using arrow function we avoid having to type the **function** keyword, **return** keyword (it’s implicit in arrow functions), and curly brackets.

```
//ES5
add = function(x, y) {
    return x + y;
}
console.log(add(1,2)); // prints 3


//ES6
add = (x,y) =>  x + y
console.log(add(1,2));  // prints 3
```

Some Tasty Examples



```

var newFlavour = flavourName => ({flavour: flavourName});


var  getTotalIceNumber = (iceLollies, iceCreams) => iceLollies+iceCreams;


var sortIceFlavours = (TotalIceFlavours) => {return TotalIceFlavours.sort()};



```

Basic Syntax

```
(param1, param2, …, paramN) => { statements }
(param1, param2, …, paramN) => expression
// equivalent to: (param1, param2, …, paramN) => { return expression; }

// Parentheses are optional when there's only one parameter name:
(singleParam) => { statements }
singleParam => { statements }
singleParam => expression


// The parameter list for a function with no parameters should be written with a pair of parentheses.
() => { statements }
```

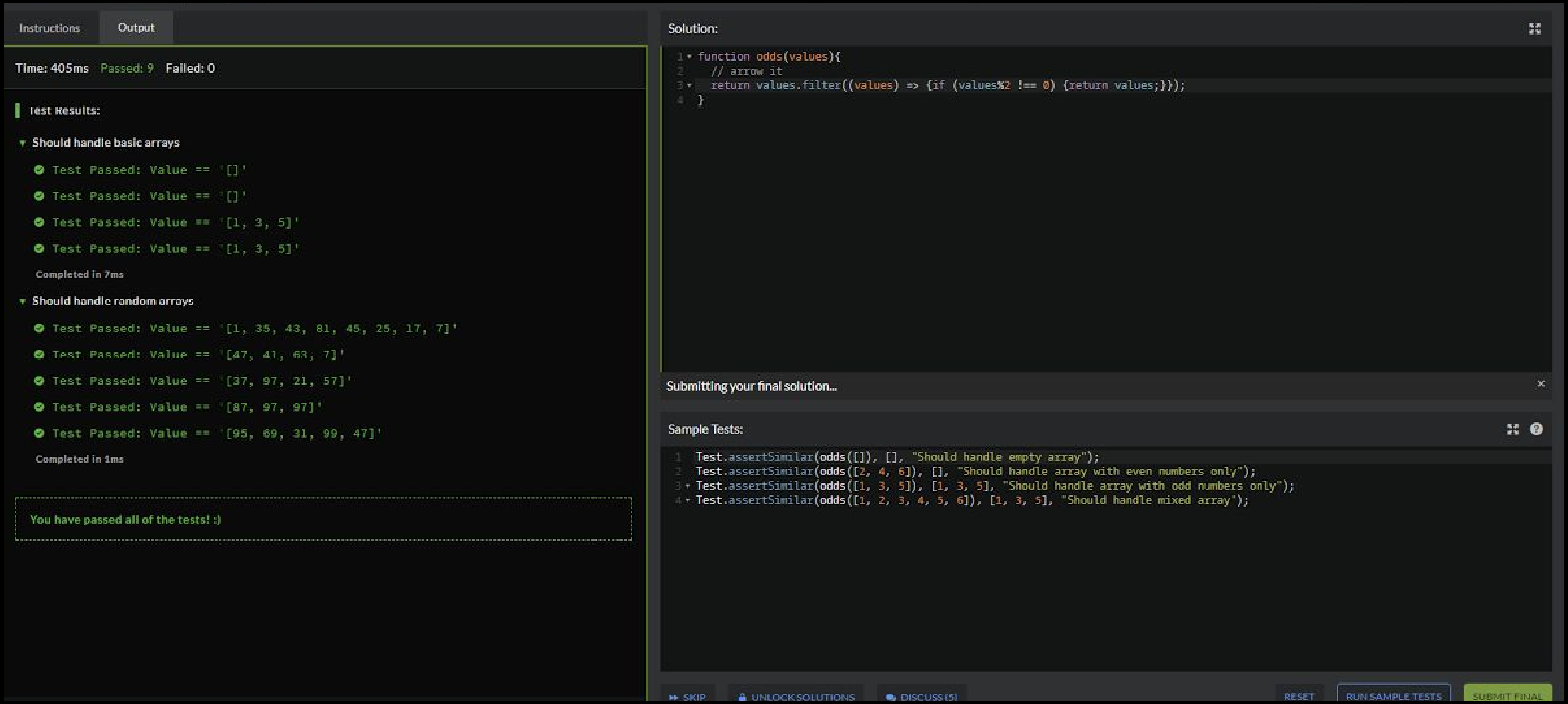
Advanced Syntax

```
// Parenthesize the body of function to return an object literal expression:
params => ({foo: bar})

// Rest parameters and default parameters are supported
(param1, param2, ...rest) => { statements }
(param1 = defaultValue1, param2, …, paramN = defaultValueN) => { statements }

// Destructuring within the parameter list is also supported
let f = ([a, b] = [1, 2], {x: c} = {x: a + b}) => a + b + c;
f();
// 6
```

Raw Horse Flesh Flavour



```
Time: 405ms   Passed: 9   Failed: 0

Test Results:
▾ Should handle basic arrays
  ✓ Test Passed: Value == '[]'
  ✓ Test Passed: Value == '[]'
  ✓ Test Passed: Value == '[1, 3, 5]'
  ✓ Test Passed: Value == '[1, 3, 5]'
  Completed in 7ms
▾ Should handle random arrays
  ✓ Test Passed: Value == '[1, 35, 43, 81, 45, 25, 17, 7]'
  ✓ Test Passed: Value == '[47, 41, 63, 7]'
  ✓ Test Passed: Value == '[37, 97, 21, 57]'
  ✓ Test Passed: Value == '[87, 97, 97]'
  ✓ Test Passed: Value == '[95, 69, 31, 99, 47]'
  Completed in 1ms

You have passed all of the tests! :)
```

Solution:

```
function odds(values){
  // arrow it
  return values.filter((values) => {if (values%2 !== 0) {return values;}});
}
```

Submitting your final solution...

Sample Tests:

```
Test.assertSimilar(odds([]), [], "Should handle empty array");
Test.assertSimilar(odds([2, 4, 6]), [], "Should handle array with even numbers only");
Test.assertSimilar(odds([1, 3, 5]), [1, 3, 5], "Should handle array with odd numbers only");
Test.assertSimilar(odds([1, 2, 3, 4, 5, 6]), [1, 3, 5], "Should handle mixed array");
```