

**Stefano Schmidt**  
**Launch School**  
**Introduction to Programming With JavaScript**  
**Flow Control – Exercises**

**1:** This exercise asks us say what the following values evaluate to:

*false* || (*true* && *false*) : evaluates to false.  
*true* || (1+2) : evaluates to true.  
(1+2) || *true* : evaluates to 3.  
*true* && (1+2) : evaluates to 3.  
*false* && (1+2) : evaluates to false.  
(1+2) && *true* : evaluates to true.  
(32\*4) >= 129 : evaluates to false.  
*false* !== *!true* : evaluates to false.  
*true* === 4 : evaluates to false.  
*false* === (847 === '847') : evaluates to true.  
*false* === (847 == '847') : evaluates to false.  
(*!true* || (!!(100/5) === 20) || ((328/4) === 82) || *false*) : evaluates to true.

**2:** This exercise asks us to write a function that takes an integer and logs 'even' to the console if the number is even, and 'odd' if the number is odd. I did this as follows:

```
function evenOrOdd(num) {  
  if (num%2 === 0) {  
    console.log('even');  
  } else {  
    console.log('odd');  
  }  
}
```

**3:** This exercise asks us to modify the function from the previous exercises to include a validation check to ensure that the argument is an integer, and issue an error message and return if it isn't. I did this as follows:

```
function evenOrOdd(num) {
  if (! Number.isInteger(num)) {
    console.log('Function expects an integer as argument');
    return;
  } else if (num%2 === 0) {
    console.log('even');
  } else {
    console.log('odd');
  }
}
```

4: This exercise asks us to explain what the following code logs to the console:

```
function barCodeScanner(serial) {
  switch (serial) {
    case '123':
      console.log('Product1');
    case '113':
      console.log('Product2');
    case '142':
      console.log('Product3');
    default:
      console.log('Product not found!');
  }
}

barCodeScanner('113');
```

The answer is that this code will log 'Product2', 'Product3', and 'Product not found!' to the console. This is because none of the cases in a switch statement have a break keyword, and so once the first case matches, the code in the remaining cases will also be executed.

**5:** This exercise asks us to refactor the following code to use an if statement instead of a ternary operator:

```
return foo() ? 'bar' : qux();
```

I did this as follows:

```
if (foo()) {  
  return 'bar';  
} else {  
  return qux();  
}
```

**6:** This exercise asks us to explain what the following code outputs to the console:

```
function isArrayEmpty(arr) {  
  if (arr) {  
    console.log('Not Empty');  
  } else {  
    console.log('Empty');  
  }  
}  
  
isArrayEmpty([]); // [] is an empty Array
```

The answer is that this code outputs 'Not Empty'. This is because arrays, even if they do not have any elements in them, are never falsy. The only falsy values are: false, 0, "", NaN, undefined, and null.

**7:** This exercise asks us to write a function that takes a string as argument and returns an all-caps version of the string if the string has more than 10 characters, and returns the original string otherwise. I did this as follows:

```
const capsIfLong = string => {
  if (string.length > 10) {
    return string.toUpperCase();
  }
  return string;
}
```

8: This exercise asks us to write a function that logs whether a number is between 0 and 50 (inclusive), between 51 and 100 (inclusive), greater than 100, or less than 0. I did this as follows:

```
const whatRange = num => {
  if (0 <= num && num <= 50) {
    console.log(`${num} is between 0 and 50`);
  } else if (51 <= num && num <= 100) {
    console.log(`${num} is between 51 and 100`);
  } else if (100 < num) {
    console.log(`${num} is greater than 100`);
  } else {
    console.log(`${num} is less than 0`);
  }
}
```

However, a slightly better/cleaner way of doing this is what is proposed in the solutions to the chapter, as it uses less comparisons and overall results in slightly more elegant code:

```
function numberRange(number) {
  if (number < 0) {
    console.log(`${number} is less than 0`);
  } else if (number <= 50) {
    console.log(`${number} is between 0 and 50`);
  } else if (number <= 100) {
    console.log(`${number} is between 50 and 100`);
  } else {
    console.log(`${number} is greater than 100`);
  }
}
```