Review

1 min

Throughout this lesson, you learned about recursion as you coded a factorial function. While every recursive problem is a little different, the following features will always be required:

- *Recursive case* – the conditions under which the function will perform an action and call itself.
- *Base case* – the conditions under which the function returns a value without making any additional calls to itself.

In this example, we started by considering the recursive case. With some problems it may be easier to start with the base case. Regardless, when you are dealing with a recursive problem, start by considering each of these cases.

Instructions

In **index.js**, we included both the iterative and recursive solutions to factorial. Both approaches work equally well for this problem.

As you learn more about recursion, you may find the syntax to be more readable and easier to understand when addressing certain problems. Consider it another tool in your toolbox as you approach increasingly challenging programming problems.

index.js

```
const recursiveFactorial = (n) => {
  if (n === 0) {
    return 1;
  } else if (n > 0) {
    return n * recursiveFactorial(n - 1);
  }
}


const iterativeFactorial = (n) => {
  result = 1;
  while(n > 0) {
    result *= n;
    n -= 1;
  }
  return result;
}
```

```
module.exports = {
  recursiveFactorial,
  iterativeFactorial
};
```