



## Lesson 6.2 - Receipt Printer

Now that we know all about template literals, let's put them to use and create a function that will print out a receipt of all our items!

We are going to use the `strip` function from the last lesson to create this receipt, so let's go ahead and make a new file called `utils-strings.js` and paste in the `strip` function:

```
export const strip = (pieces, ...values) =>
  pieces
    .reduce((concatStr, piece, indx) => `${concatStr}${piece}${values[indx]} || ' '` , '')
    .replace(/^\s*/gm, '');
```

Ok great! Now it's as simple as importing the modules we need and creating a function that will print out a receipt! Lets create a file called `printer.js` and create our function:

```
import { strip } from './utils-strings';
import { calculate } from './calculator';

export function printReceipt(items) {
  let {total, totalTax} = calculate({ items });

  return strip`
    ${items.map(item => item.price).join('\n')}
    ${'-'}.repeat(30)}
    Sub-Total: ${total}
    ${'='}.repeat(30)}
    Tax: ${totalTax}
    ${'='}.repeat(30)}
    Total: ${total + totalTax}
  `;
}
```

That's it! So what's happening here? The first thing we did was calculate the total price and total tax using our `calculate` function, and assigned the values to `total` and `totalTax` variables using destructuring. Then we used our custom template literal called `strip` to print out our receipt! We also snuck in another new ES2015 feature: `String.prototype.repeat`. It does exactly what you think it does, repeat the string the number of times you specify in the argument! Let's try this out in the console:

```
$ babel-node
> var { items } = require('./calculator')
> var { printReceipt } = require('./printer')
> console.log(printReceipt(items))
->
10
22
13.45
-----
Sub-Total: 45.45
=====
Tax: 1.876
=====
Total: 47.326
```