```javascript
// LECTURE 15: Template Literals
// The traditional method
const firstName = "Jonathan", job = "unemployed", birthYear = 2001;
const jonathan = "I'm " + firstName + ", " + job + ", age " + (2021 - birthYear); // see the operator precedence, we are forcing the compiler to do the subtraction before the concatenation.
// Also how does this work in JS since 2021 - birthYear is a number? This is something called "Type Coersion" - but JS will automatically convert this number into a string and THEN output it to the console.
console.log(jonathan);

// Using template literals for strings
// Can assemble multiple pieces into 1 final strings
// Template literals use backticks ``
const newJon = `I'm ${firstName}, ${job}, age ${2021-birthYear}`; // new ES6 feature!!
console.log(newJon);
// We can use backticks for any regular string
console.log(`Just a normal string!`);

// Creating multiline strings
console.log('String with \n\
multiple lines \n\
lines'); // JavaScript ES5
//With template strings:
console.log(`string with
Multiple
Lines`); // you don't need to put \n\
  anymore!!
```