

## Computer Property Names

ES6's "Computed Property Names" feature allows you to have an expression (a piece of code that results in a single value like a variable or function invocation) be computed as a property name on an object.

For example, say you wanted to create a function that took in two arguments (key, value) and returned an object using those arguments. Before Computed Property Names, because the property name on the object was a variable (key), you'd have to create the object first, then use bracket notation to assign that property to the value.

```
function objectify(key, value) {  
  let obj = {};  
  obj[key] = value;  
  return obj;  
}  
  
objectify("name", "Tyler"); // { name: 'Tyler' }
```

However, now with Computed Property Names, you can use object literal notation to assign the expression as a property on the object without having to create it first. So the code above can now be rewritten like this.

```
function objectify(key, value) {  
  return {  
    [key]: value,  
  };  
}  
  
objectify("name", "Tyler"); // { name: 'Tyler' }
```

Where key can be any expression as long as it's wrapped in brackets, [].