**Title:** Default parameters

**Answer:**

**Default function parameters** allow formal parameters to be initialized with default values if no value or `undefined` is passed.

```
function multiply(a, b = 1) {
  return a * b;
}

console.log(multiply(5, 2));
// expected output: 10

console.log(multiply(5));
// expected output: 5
```

In JavaScript, parameters of functions default to `undefined (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/undefined)` . However, in some situations it might be useful to set a different default value. This is where default parameters can help.

In the past, the general strategy for setting defaults was to test parameter values in the body of the function and assign a value if they are `undefined` . If in the following example, no value is provided for `b` in the call, its value would be `undefined` when evaluating `a*b` and the call to `multiply` would have returned `NaN` . However, this is caught with the second line in this example:

```
function multiply(a, b) {
  b = (typeof b == 'undefined') ?  1 : b;
  return a * b;
}

multiply(5, 2); // 10
multiply(5, 1); // 5
multiply(5);    // 5
```

With default parameters in ES2015 ( ES6 ), the check in the function body is no longer necessary. Now, you can simply put `1` as the default value for `b` in the function head:

```
function multiply(a, b = 1) {
  return a * b;
}

multiply(5, 2); // 10
multiply(5, 1); // 5
multiply(5);    // 5
```

**Tags:** es6, functions / methods, javascript