Take a look at this code:

**function** **greet**(name, greeting) {

name = (**typeof** name !== 'undefined') ? name : 'Student';

greeting = (**typeof** greeting !== 'undefined') ? greeting : 'Welcome';

**return** `${greeting} ${name}!`;

}

greet(); *// Welcome Student!*

greet('James'); *// Welcome James!*

greet('Richard', 'Howdy'); *// Howdy Richard!*

***Returns:*** *Welcome Student!  
Welcome James!  
Howdy Richard!*

What is all that horrible mess in the first two lines of the greet() function? All of that is there to provide default values for the function if the required arguments aren't provided. It's pretty ugly, though...

Fortunately, ES6 has introduced a new way to create defaults. It's called *default function parameters*.

**Default function parameters**

**Default function parameters** are quite easy to read since they're placed in the function's parameter list:

**function** **greet**(name = 'Student', greeting = 'Welcome') {

**return** `${greeting} ${name}!`;

}

greet(); *// Welcome Student!*

greet('James'); *// Welcome James!*

greet('Richard', 'Howdy'); *// Howdy Richard!*

***Returns:*** *Welcome Student!  
Welcome James!  
Howdy Richard!*

Wow, that's a lot less code, so much cleaner, and significantly easier to read!

To create a default parameter, you add an equal sign ( = ) and then whatever you want the parameter to default to if an argument is not provided. In the code above, both parameters have default values of strings, but they can be any JavaScript type!

**QUIZ QUESTION**

Take a look at the following code:

**function** **shippingLabel**(name, address) {

name = (**typeof** name !== 'undefined') ? name : 'Richard';

address = (**typeof** address !== 'undefined') ? address : 'Mountain View';

**return** `To: ${name} In: ${address}`;

}

Which of the following choices is the correct way to write the shippingLabel() function using default function parameters?

* 

function shippingLabel(name = '', address = '') {  
    return `To: ${name} In: ${address}`;  
}

* 

function shippingLabel(name, address) {  
    name = name || 'Richard';  
    address = address || 'Mountain View';  
    return `To: ${name} In: ${address}`;  
}

* 

function shippingLabel(name, address) {  
    return `To: ${name} In: ${address}`;  
}

* function shippingLabel(name = 'Richard', address = 'Mountain View') {  
      return `To: ${name} In: ${address}`;  
  }