Dart has two types of optional parameters: *named* and *positional*.

**Positional optional parameters**

A parameter wrapped by [ ] is a positional optional parameter. Here is an example:

getHttpUrl(String server, String path, [int port=80]) {  
 // ...  
}

In the above code, port is optional and has a default value of 80.

You can call getHttpUrl with or without the third parameter.

getHttpUrl('example.com', '/index.html', 8080); // port == 8080  
getHttpUrl('example.com', '/index.html'); // port == 80

You can specify multiple positional parameters for a function:

getHttpUrl(String server, String path, [int port=80, int numRetries=3]) {  
 // ...  
}

The optional parameters are *positional* in that you can't omit port if you want to specify numRetries.

getHttpUrl('example.com', '/index.html');  
getHttpUrl('example.com', '/index.html', 8080);  
getHttpUrl('example.com', '/index.html', 8080, 5);

Of course, unless you know what 8080 and 5 are, it's hard to tell what those apparently magic numbers are. You can use *named optional parameters* to create more readable APIs.

**Named optional parameters**

A parameter wrapped by { } is a named optional parameter. Here is an example:

getHttpUrl(String server, String path, {int port = 80}) {  
 // ...  
}

You can call getHttpUrl with or without the third parameter. You **must** use the parameter name when calling the function.

getHttpUrl('example.com', '/index.html', port: 8080); // port == 8080  
getHttpUrl('example.com', '/index.html'); // port == 80

You can specify multiple named parameters for a function:

getHttpUrl(String server, String path, {int port = 80, int numRetries = 3}) {  
 // ...  
}

Because named parameters are referenced by name, they can be used in an order different from their declaration.