



A snippet is a small template for code that performs a specific task in a specific language and language version.



Each snippet belongs to a specific topic. Examples may include: Promises, jQuery AJAX request, SQL Select statement, etc.



Each snippet is written for a specific programming language



Each snippet is written for a specific version of said programming language



Snippets may be accompanied by fully written, working code samples that demonstrate the topic. A snippet may have more than one related code sample.



Snippets may be accompanied by an explanation of the topic.

There is only one explanation allowed for each snippet.



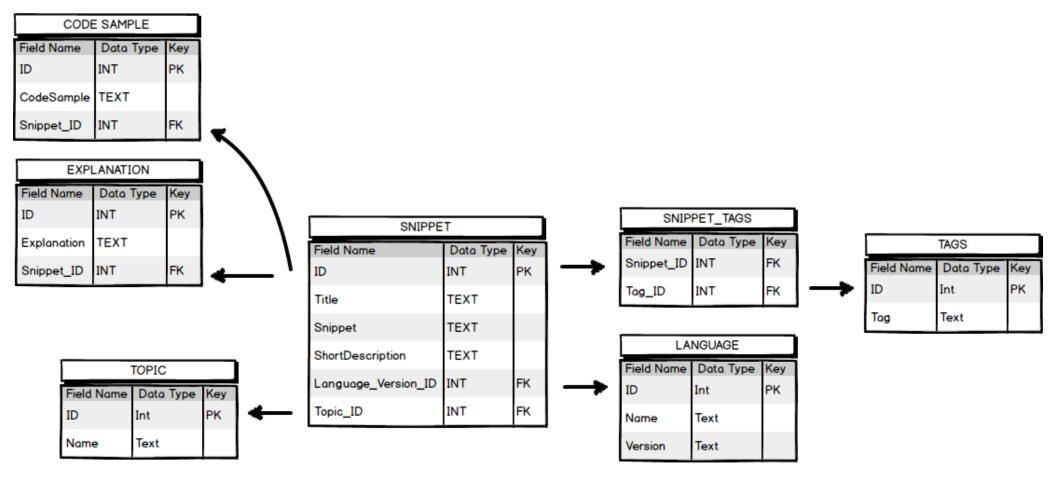
Snippet authors and consumers can tag the snippet with additional keywords to enhance search



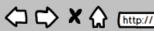
Sometimes there is more than one way to perform a task. In this case, a snippet can have more than one variation. All related snippets would be stored in the Snippet table, but the relationship would be tracked in another table.



A snippet has one author. A code sample has one author. An explanation can have many authors (like a wiki).



Users				
Field Name	Data Type	Key		
ID	INT	PK		
Username	TEXT			
Password	TEXT			





# >\_ A Big Title Home | About | Search | Add a Snippet

Q search

### Creating a Promise in JavaScript (ES6)

### Snippet

```
//Declaration
let myFirstPromise = new Promise((resolve, reject) => {
});
//Use:
myFirstPromise.then((successMessage) => {
```

### Examples

∢ Add an Example

```
One Two Three Four
let myFirstPromise = new Promise((resolve, reject) => {
 var x = 5
 var y = 10;
 resolve(x * y);
});
myFirstPromise.then((success) => {
 console.log(success); // console logs 50
});
```

### Rate this Code Sample

### Explanation

**∢** Edit this Example

A promise can be:

fulfilled - The action relating to the promise succeeded rejected - The action relating to the promise failed pending - Hasn't fulfilled or rejected yet settled - Has fulfilled or rejected

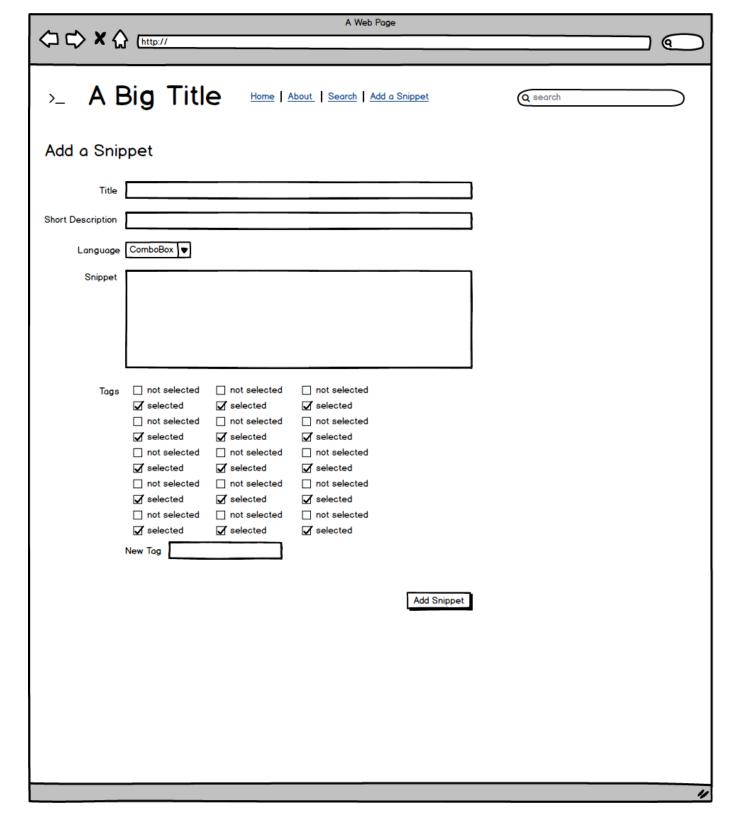
The promise constructor takes one argument, a callback with two parameters, resolve and reject. Do something within the callback, perhaps async, then call resolve if everything worked, otherwise call

Like throw in plain old JavaScript, it's customary, but not required, to reject with an Error object. The benefit of Error objects is they capture a stack trace, making debugging tools more helpful.

#### Tags

reject.

JavaScript Promises ES6 Asynchronous Programming



Ideas for Tag Submission: Autocomplete? Have Preset Tags Available, Choose using Drop-down list or Checkboxes? Only add Tag if it doesn't already exist?





	•	<b>—</b> :	
	^	Tit	$\sim$
>	$\boldsymbol{\leftarrow}$		_
_	, ,		

Home | About | Search | Add a Snippet

**Q** search

Add Explanation
-----------------

I			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı			
ı	1		
ı			
ı			
I			
ı			
I			
I			
I			
I			
I			
I			
١	I		

## **Snippet Title**

//Declaration

let myFirstPromise = new Promise((resolve, reject) => { });

//Use:

myFirstPromise.then((successMessage) => {
});

Add Explanation

_			$\overline{}$
(Q	search		
<i>(~</i>	ocuron.		

Add Code Sample
-----------------

l		
1		

## **Snippet Title**

//Declaration

let myFirstPromise = new Promise((resolve, reject) => { **})**;

//Use:

myFirstPromise.then((successMessage) => {

Add Example

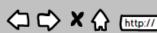
Α				





Q Promises

Snippet	Topic	Langauge	Tags
Creating Promises using ES6	Promises	JavaScript (ES6)	#Asynchronous #JavaScript #Promises
Basic JavaScript Promises	Promises	JavaScript	





# >\_ A Big Title

Home About Search Add a Snippet

**Q** search

### Recent Snippets

Snippet	Topic	Langauge
Creating Promises using ES6	Promises	JavaScript (ES6)
Basic JavaScript Promises	Promises	JavaScript
Creating Promises using ES6	Promises	JavaScript (ES6)
Basic JavaScript Promises	Promises	JavaScript
Creating Promises using ES6	Promises	JavaScript (ES6)
Basic JavaScript Promises	Promises	JavaScript
Creating Promises using ES6	Promises	JavaScript (ES6)
Basic JavaScript Promises	Promises	JavaScript

### Popular Snippets

Topic	Langauge
Promises	JavaScript (ES6)
Promises	JavaScript
	Promises

### Featured Snippet



#### Featured Snippet Title

A paragraph of text with an unassigned link. A second row of text with a web link

//Declaration let myFirstPromise = new Promise((resolve, reject) => { }); //Use: myFirstPromise.then((successMessage) => {

#### A promise can be:

fulfilled - The action relating to the promise succeeded rejected - The action relating to the promise failed pending - Hasn't fulfilled or rejected yet settled - Has fulfilled or rejected

The promise constructor takes one argument, a callback with two parameters, resolve and reject. Do something within the callback, perhaps async, then call resolve if everything worked, otherwise call reject.

Like throw in plain old JavaScript, it's customary, but not required, to reject with an Error object. The benefit of Error objects is they capture a stack trace, making debugging tools more helpful.