

\$(this)

Consider the following HTML codes:

```





```

Let's say that you want to make your app work such that when the user clicks an image, that particular image disappears. How would you do this?

One way is to change your HTML so that we add a unique ID for each image and add a jQuery code like below:

```
<html>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
  <script>
    $(document).ready(function() {
      $('#image1').click(function() {
        $('#image1').hide();
      });
      $('#image2').click(function() {
        $('#image2').hide();
      });
      $('#image3').click(function() {
        $('#image3').hide();
      });
      $('#image4').click(function() {
        $('#image4').hide();
      });
      $('#image5').click(function() {
        $('#image5').hide();
      });
    });
  </script>
</head>
<body>
  
  
  
  
  
</body>
```

But the above code is redundant! What if there is a way we can choose the particular element within the selector that triggered the event? And there is! It's called `$(this)`. `$(this)` selects that particular element that triggered that event. Using this, we can clean up the above code like below:

```
<html>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
</head>
<body>
  <script>
    $(document).ready(function() {
      //attach a click event listener to all the img tags when the document is ready
      $('img').click(function() {
        $(this).hide();
      });
    });
  </script>
</body>
</html>
```

This is an example of how to streamline your jQuery code. You never have to write the same code twice. By leveraging the power of the `$(this)` selector, you can really cut down on the number of lines of code you have.