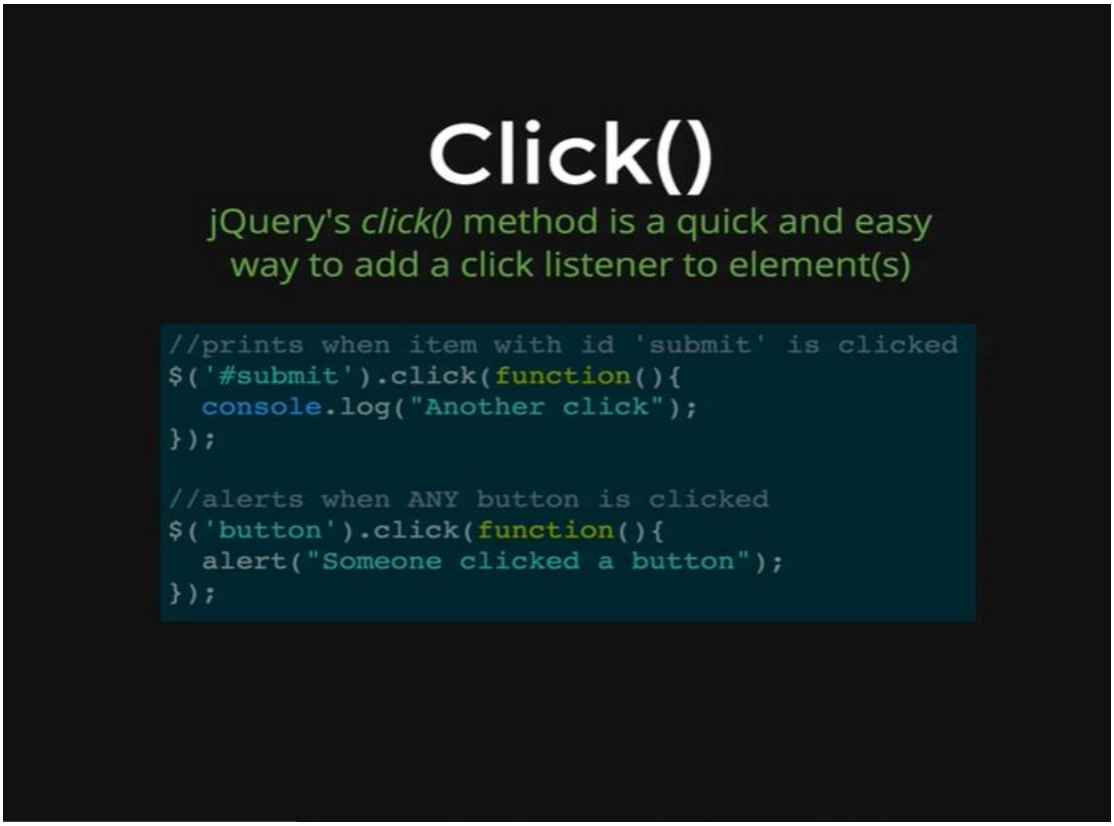
Objectives:

- *click()*
- *keypress()*
- *on()*

In this lesson we will introduce Events in jQuery. Think back to vanilla JS, when we added event-listeners, we used a method called *addEventListener.* jQuery has a ton of methods that had to do with events. We can check all those events by visiting the link below.

api.jquery.com/category/events

We will highlight the three most important ones, the three that we can use 99% of the times. Those methods are *click()*, *keypress()* and *on()*. These three methods we can use the 99% of the times. It is actually the on() method that we can use 99% of the times.



jQuery's *click* method is how we can very quickly and easily add a click listener to a single element or a collection of elements. We can see on the picture that we have two examples. The first one as the following.

```
// prints when item with id 'submit' is clicked
$('#submit').click(function() {
        console.log("Another click");
});
```

Here, we are selecting an id and then adding a click listener using the *.click()* method. We are passing a callback function inside the *.click()* method, which means when something with the id of 'submit' will be clicked the callback function will run. As we are selecting an id, it is only one on the page and then when that one item is clicked the console.log in the callback function is run.

Now, lets demonstrate the second example on the picture. It is as below.

```
// alerts when ANY button is clicked
$('button').click(function() {
        alert("Someone clicked a button");
});
```

Here we can see that we are selecting all buttons on the page and we are adding a click listener to all of them. That is why anytime a button is clicked, we will alert *"Someone clicked a button"*.

Let's demonstrate this on our HTML page.

```
<!DOCTYPE html>
<html>
    <head>
        <title>jQuery Events</title>
        <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
    </head>
    <body>
        <h1>jQuery Events</h1>

        <button>Don't Click Me</button>
        <button>Seriously Don't Click Me</button>
        <button>This is your last warning!</button>
    </body>
</html>
```

Output:

Console coding:

jQuery

➔ *function (a,b) {return new n.fn.init(a,b)}*

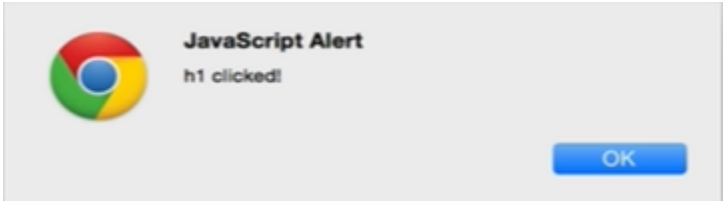
$("h1")

➔ *[<h1>jQuery Events</h1>]*

```
$("h1").click(function() {
    alert("h1 clicked!");
});
```

➔ *[<h1>jQuery Events</h1>]*

We have added a click listener to our *"h1"* element and now when we click on *"h1"* we get the following alert.



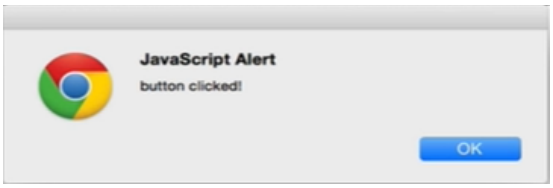
Now we will see how we can add a click listener to a collection of elements. We are going to select all the buttons on our page.

Console coding:

```
$("button").click(function() {
    alert("button clicked!");
});
```

➔ *[<button>Don't Click Me</button>, <button>Seriously Don't Click Me</button>, <button>This is your last warning</button>]*

Now when we click on any of those buttons we get the following alert.



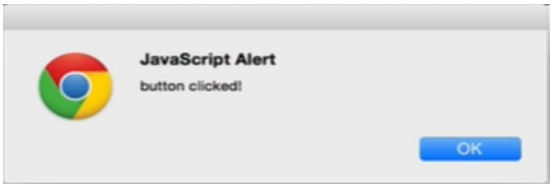
Think back to vanilla JavaScript, to add an event listener to collection of buttons we would need to select them all with *querySelectorAll* or *getElementsByTagName* and then we need to use a for loop to loop through all of the selected elements and then we need to individually add an event listener. This what is happening behind the scene here in jQuery too but all we need to do is write the short jQuery code like the click listener and then that would add the event listener and loop through for the us without us defining those extra methods. This makes our lives a lot easier.

Rather than just alerting the user how about we want to change the background color of a button. We can do that as below.

Console coding:

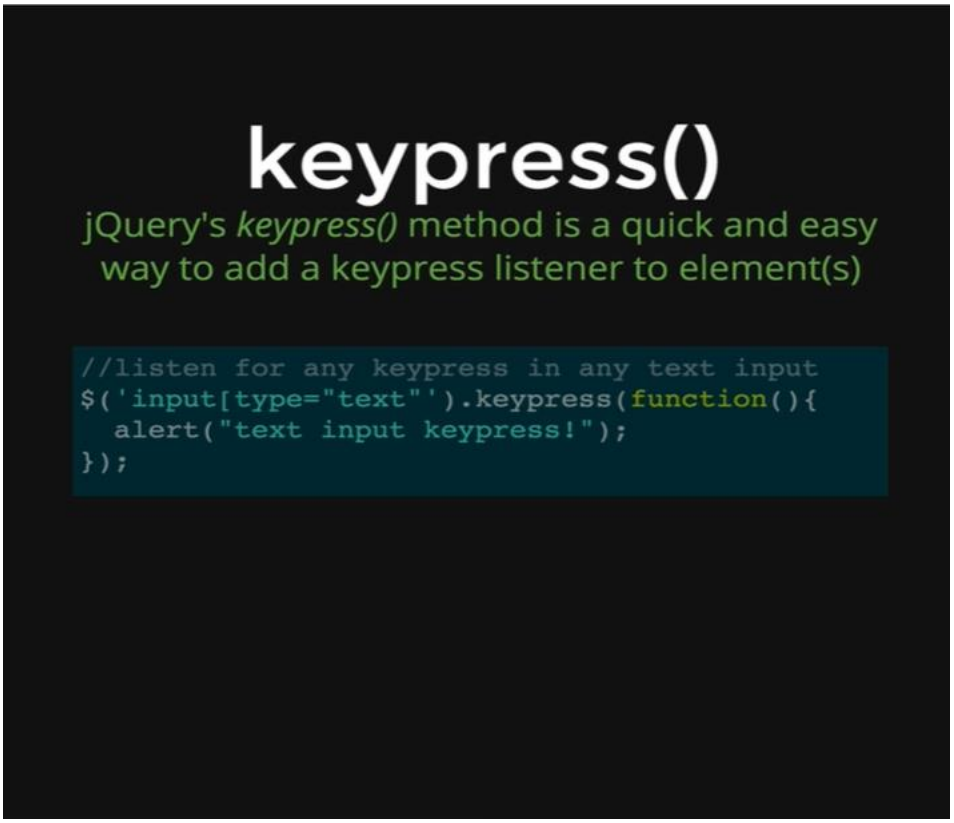
```
$("button").click(function() {
        $(this).css("background", "pink");
});
```

➔ *[<button>Don't Click Me</button>, <button>Seriously Don't Click Me</button>, <button>This is your last warning</button>]*



First, we get the alert because, we our on click function before this is still registered, and then the button we clicked turns pink.

The next click event we will see is called *keypress().*



keypress() is one of the ways in jQuery that we can add a keypress listener to elements. There three events in the jQuery framework called *.keydown(), .keypress()* and *.keyup()*; these are three different events those are triggered when a user type something on the keyboard.

The event *.keydown()* is fired after we press a key down.

The event *.keyup()* is fired when we release a key.

The event *.keypress()* is fired between *.keyup* and *.keydown.* The keydown and keyup provide a code indicating which key is pressed, while keypress indicates which character was entered. For example, a lowercase “a” will be reported as 65 by keydown and keyup, but as 97 by keypress. An uppercase “A” is reported as 65 by all events. Because of this distinction, when catching special keystrokes such as arrow keys, *.keydown()* or *.keyup()* is a better choice.

.keypress() works just like *.click()*, where we select something with dollar sign ($). For example we have a certain class on our HTML page and we can add a *.keypress()* event on that class passing a function with that event. Now if we press any key on that class at our webpage, the *.keypress()* event will execute the callback function that was passed in it.

```
$(“#target”).keypress(function() {
        console.log( “Handler for .keypress() called.”);
});
```

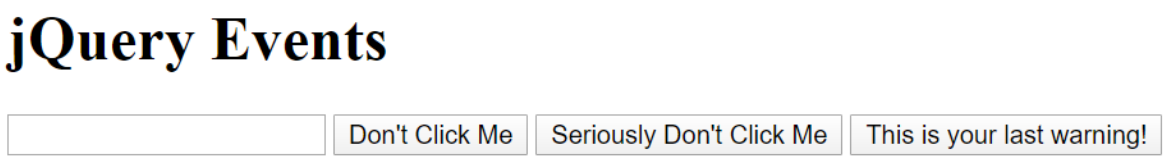
Let’s demonstrate this method on our HTML page.

```
<!DOCTYPE html>
<html>
  <head>
    <title>jQuery Events</title>
    <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
  </head>
  <body>
    <h1>jQuery Events</h1>

      <input type= “text”>

    <button>Don't Click Me</button>
    <button>Seriously Don't Click Me</button>
    <button>This is your last warning!</button>
  </body>
</html>
```

Output:



Now anytime we press in a key on our input field it will get console.logged by the code we are going to write now.

Console coding:

$("input")

- ➔ *[<input type= "text">]*

$("input").keypress(function () {

console.log("YOU PRESSED A KEY!");

});

Now whenever we press a key on our input field, we get the output as following.

YOU PRESSED A KEY!