# Bachelor Of Engineering
# In
# Information Technology

# Semester Six, Third Year(Even semester)
# 20th April 2022
# Offline - 13th Lecture

Padre Conceicao College of Engineering

Verna Goa 403722 India

# Web Technology

**RC 2019-20**

**Unit 2**

# UNIT 2

| Topic | Subtopics |
|---|---|
| jQuery: | Introduction, Syntax, jQuery Selector, jQuery Events, jQuery effects, jQuery and HTML |

# UNIT 2 :

| Sr.No | Title |
|---|---|
| 1 | Introduction |
| 2 | Syntax |
| 3 | jQuery Selector |
| 4 | jQuery Events |
| 5 | jQuery effects |
| 6 | jQuery and HTML |

# Introduction

- jQuery is a lightweight, "write less, do more", JavaScript library.
- jQuery is easy to learn.
- jQuery is the most popular and most extendable JavaScript library
- jQuery greatly simplifies JavaScript programming.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.
- Before you start studying jQuery, you should know HTML , CSS , Javascript

# jQuery features

The jQuery library contains the following features:

1. HTML/DOM manipulation
2. CSS manipulation
3. HTML event methods
4. Effects and animations
5. AJAX
6. Utilities

# Companies that use jQuery

The biggest companies on the Web use jQuery, such as:

1. IBM
2. Netflix
3. Google
4. Microsoft

# How to use jQuery on your web site?

1. Download the jQuery library from jQuery.com
2. Include jQuery from a CDN(content delivery network ), like Google

# Downloading jQuery

There are two versions of jQuery available for downloading:

1. Production version - this is for your live website because it has been minified and compressed
2. Development version - this is for testing and development (uncompressed and readable code)

Both versions can be downloaded from jQuery.com

The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section):

# How to reference jQuery Library in HTML file

- The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section)

```
<html>

<head>
<script src="jquery-3.6.0.min.js"></script>
</head>
```

- Place the downloaded file in the same directory as the pages where you wish to use it.

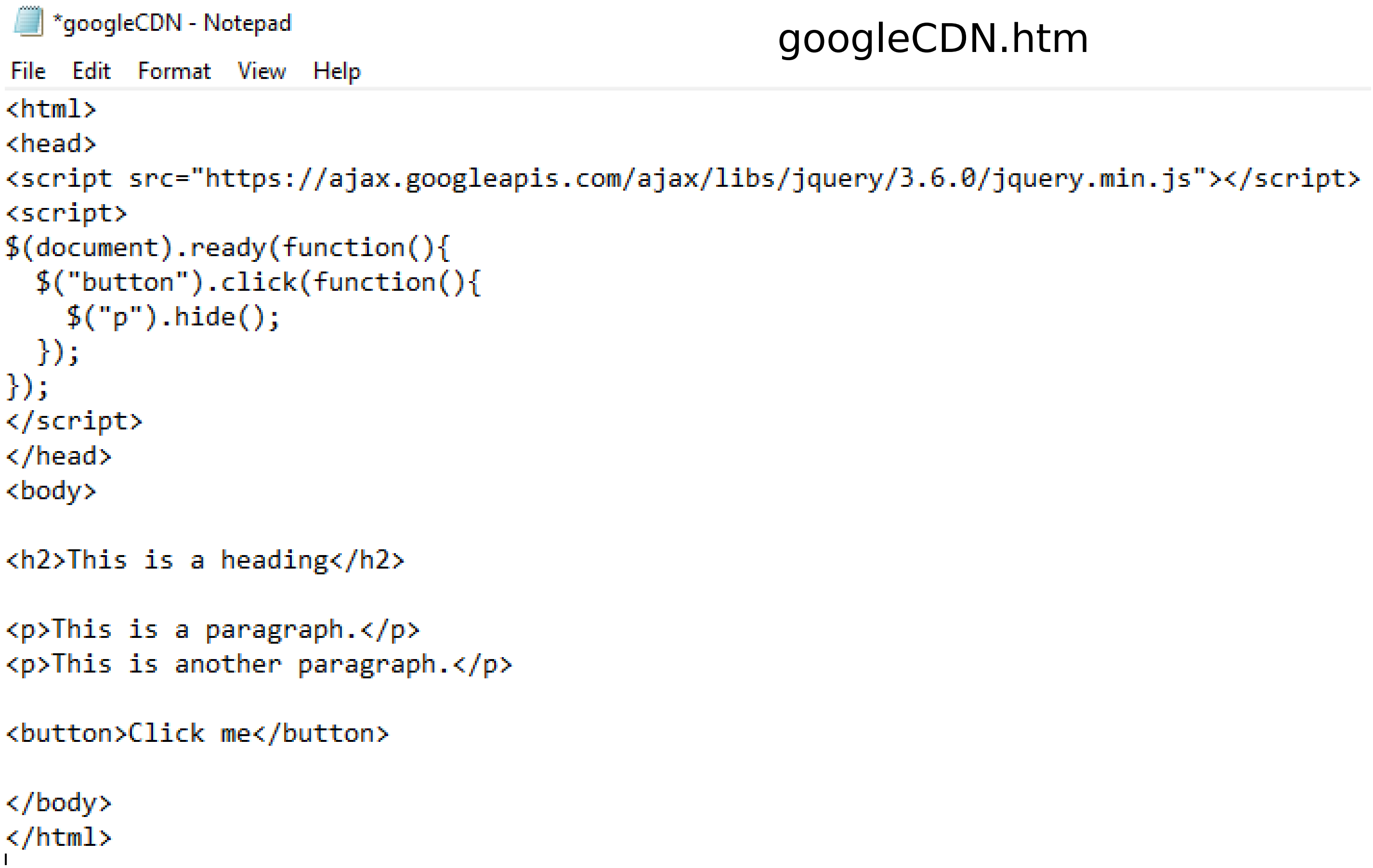
# jQuery CDN

- If you don't want to download and host jQuery yourself, you can include it from a CDN.
- An example of a Company who host jQuery is Google
- Google CDN:

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/
jquery/3.6.0/jquery.min.js"></script>
</head>
```

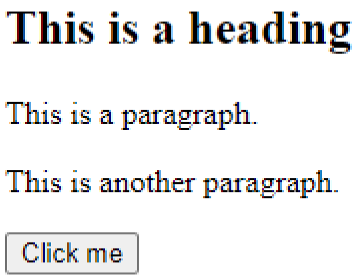
# Google CDN

googleCDN.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Click me</button>

</body>
</html>
```

output



**This is a heading**

This is a paragraph.

This is another paragraph.

# One advantage of using the hosted jQuery from Google

- Many users already have downloaded jQuery from Google when visiting another site.
- As a result, it will be loaded from cache when they visit your site, which leads to faster loading time.
- Most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

# jQuery Syntax

- The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).
- Basic syntax is: **$(*selector*).*action*()**

1. A **$** sign to define/access jQuery
2. A **(*selector*)** to "query (or find)" HTML elements
3. A jQuery ***action*()** to be performed on the element(s)

# Examples

**$(this).hide()** - hides the current element.
**$("p").hide()** - hides all <p> elements.
**$(".test").hide()** - hides all elements with class="test".
**$("#test").hide()** - hides the element with id="test".

# jQuery Selectors

- jQuery selectors are one of the most important parts of the jQuery library.
- jQuery selectors allow you to select and manipulate HTML element(s).
- jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and, it has some own custom selectors.
- All selectors in jQuery start with the dollar sign and parentheses: $().

# The element Selector

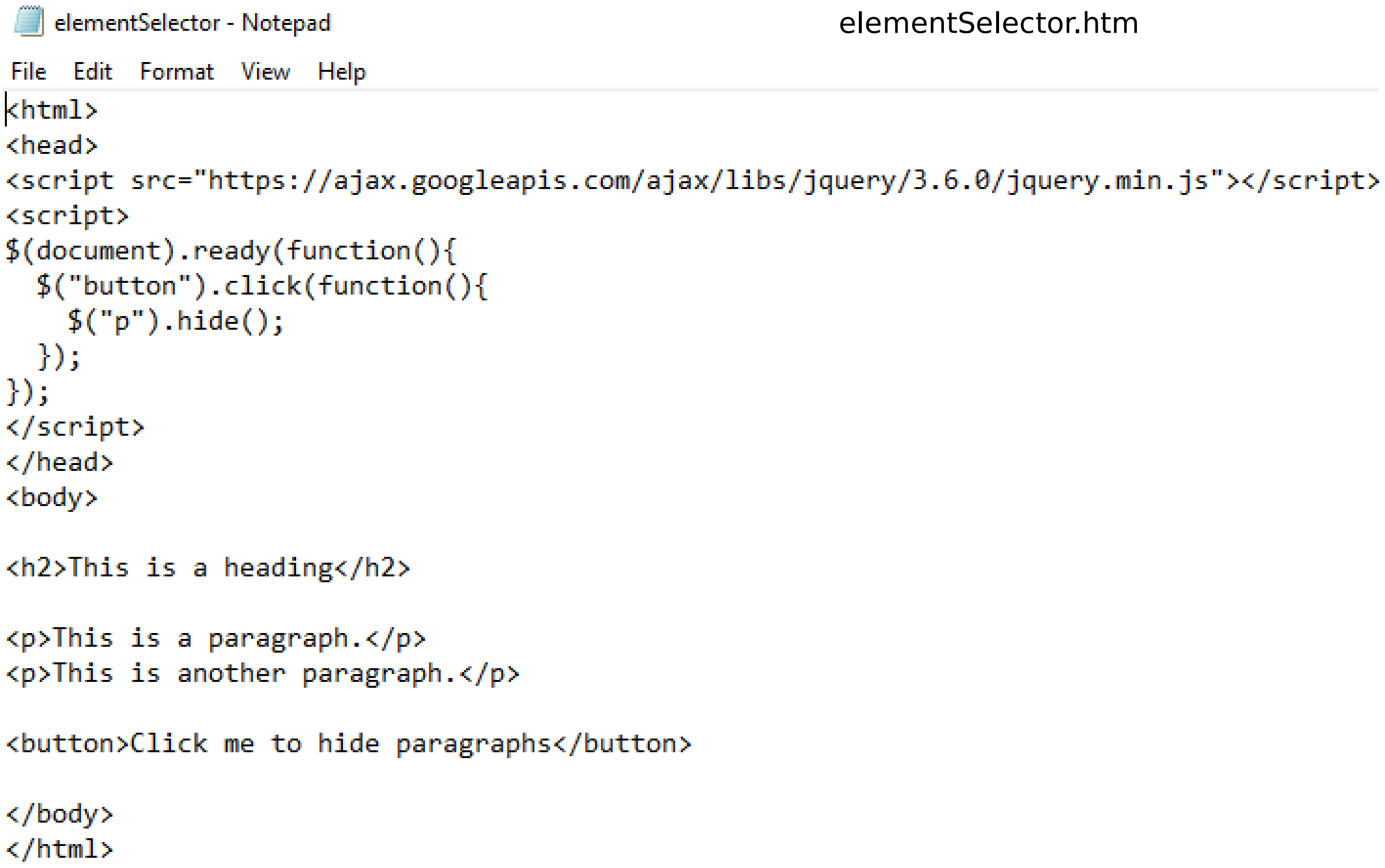
- The jQuery element selector selects elements based on the element name.
- You can select all <p> elements on a page like this:

```
$("p")
```

- Example
- When a user clicks on a button, all <p> elements will be hidden:

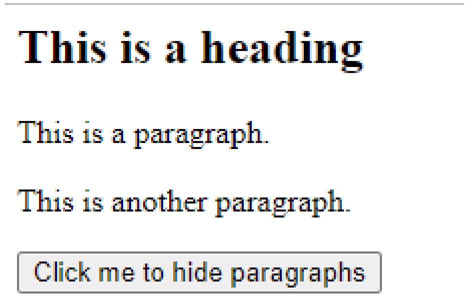
# The element Selector

elementSelector.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Click me to hide paragraphs</button>

</body>
</html>
```

output



**This is a heading**

This is a paragraph.

This is another paragraph.

# The #id Selector

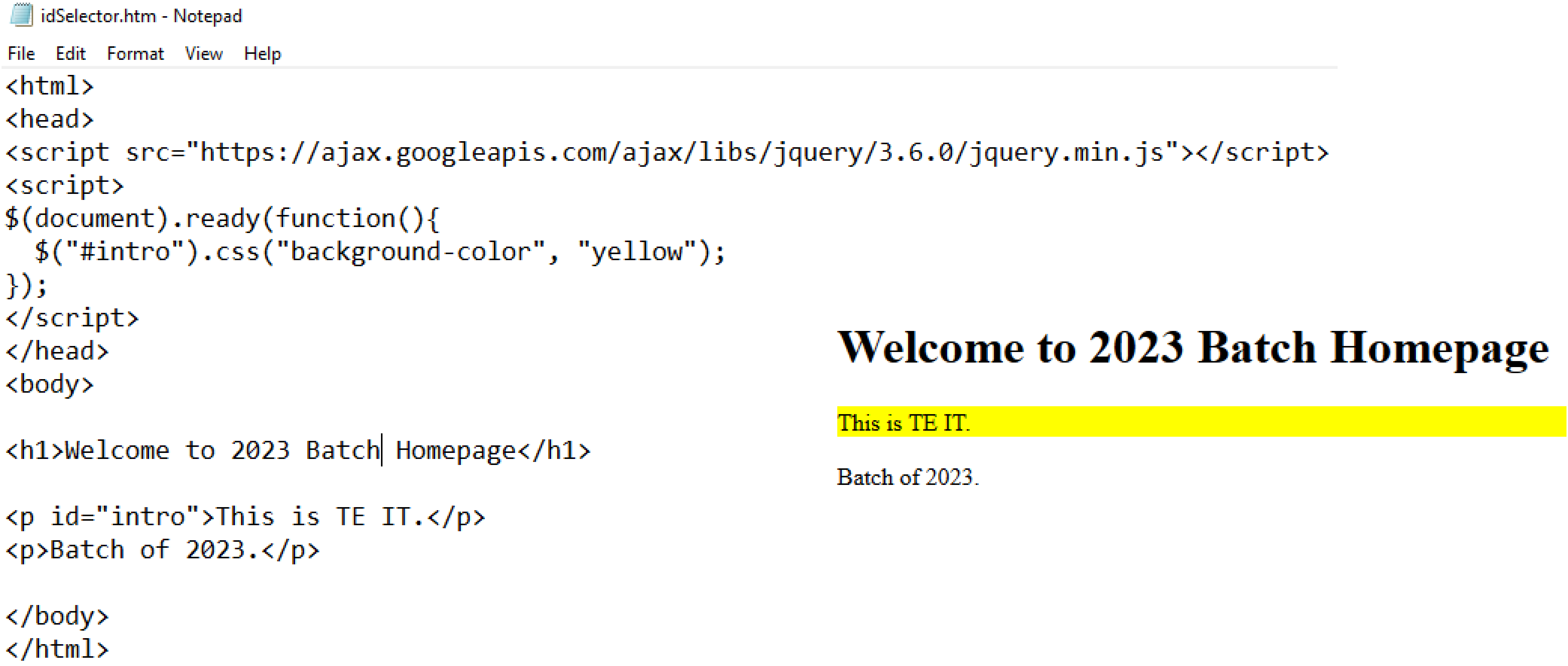
- The jQuery #id selector selects the element with the specific id.
- The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.
- An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.
- To find an element with a specific id, write a hash character, followed by the id of the HTML element
- Do not start an id attribute with a number. It may cause problems in some browsers.

# The #id Selector syntax

```
$("#id“)
```

| Parameter | Description |
|---|---|
| *id* | Required. Specifies the id of the element to select |

# The #id Selector



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#intro").css("background-color", "yellow");
});
</script>
</head>
<body>

<h1>Welcome to 2023 Batch Homepage</h1>

<p id="intro">This is TE IT.</p>
<p>Batch of 2023.</p>

</body>
</html>
```

**Welcome to 2023 Batch Homepage**

This is TE IT.

Batch of 2023.

# jQuery Event Methods

- jQuery is tailor-made to respond to events in an HTML page.

What are Events?

- All the different visitors' actions that a web page can respond to are called events.
- An event represents the precise moment when something happens.
- Examples:

1. moving a mouse over an element
2. selecting a radio button
3. clicking on an element

- The term **"fires/fired"** is often used with events. Example: "The keypress event is fired, the moment you press a key".

# Common DOM events

| Mouse Events | Keyboard Events | Form Events | Document/Window Events |
|---|---|---|---|
| click | keypress | submit | load |
| dblclick | keydown | change | resize |
| mouseenter | keyup | focus | scroll |
| mouseleave |  | blur | unload |
| Mouse Events | Keyboard Events | Form Events | Document/Window Events |

# jQuery Syntax For Event Methods

- In jQuery, most DOM events have an equivalent jQuery method.
- To assign a click event to all paragraphs on a page, you can do this:

```
$("p") .click();
```

The next step is to define what should happen when the event fires. You must pass a function to the event

```
$("p").click(function(){
  // action goes here!!
});
```

# Commonly Used jQuery Event Methods

1. **$(document).ready()**

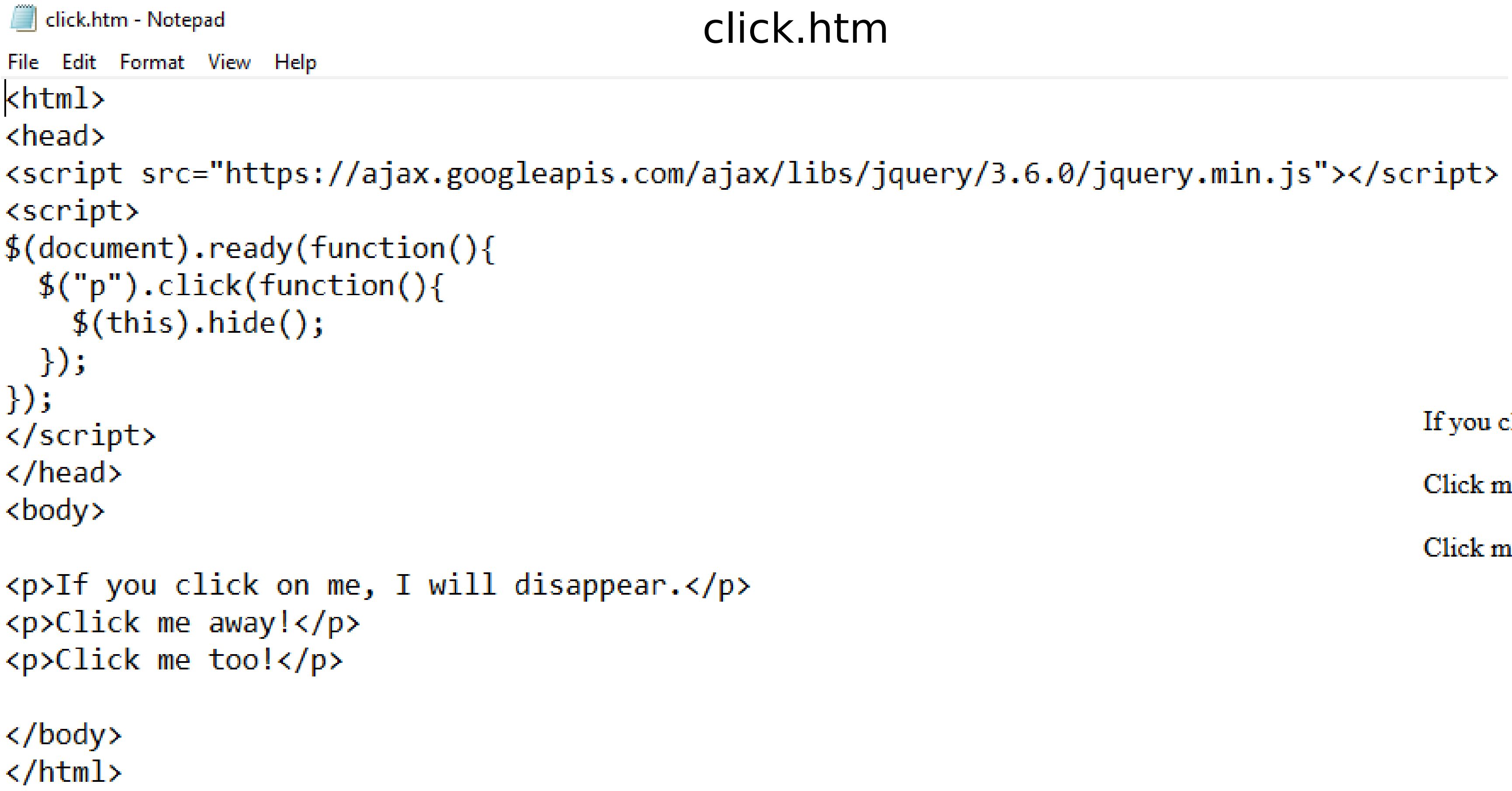
- The **$(document).ready()** method allows us to execute a function when the document is fully loaded.

2 . **click()**

- The **click()** method attaches an event handler function to an HTML element.
- The function is executed when the user clicks on the HTML element.
- Example : When a click event fires on a <p> element; hide the current <p> element:

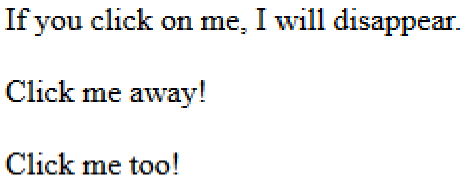
# click()

click.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

output



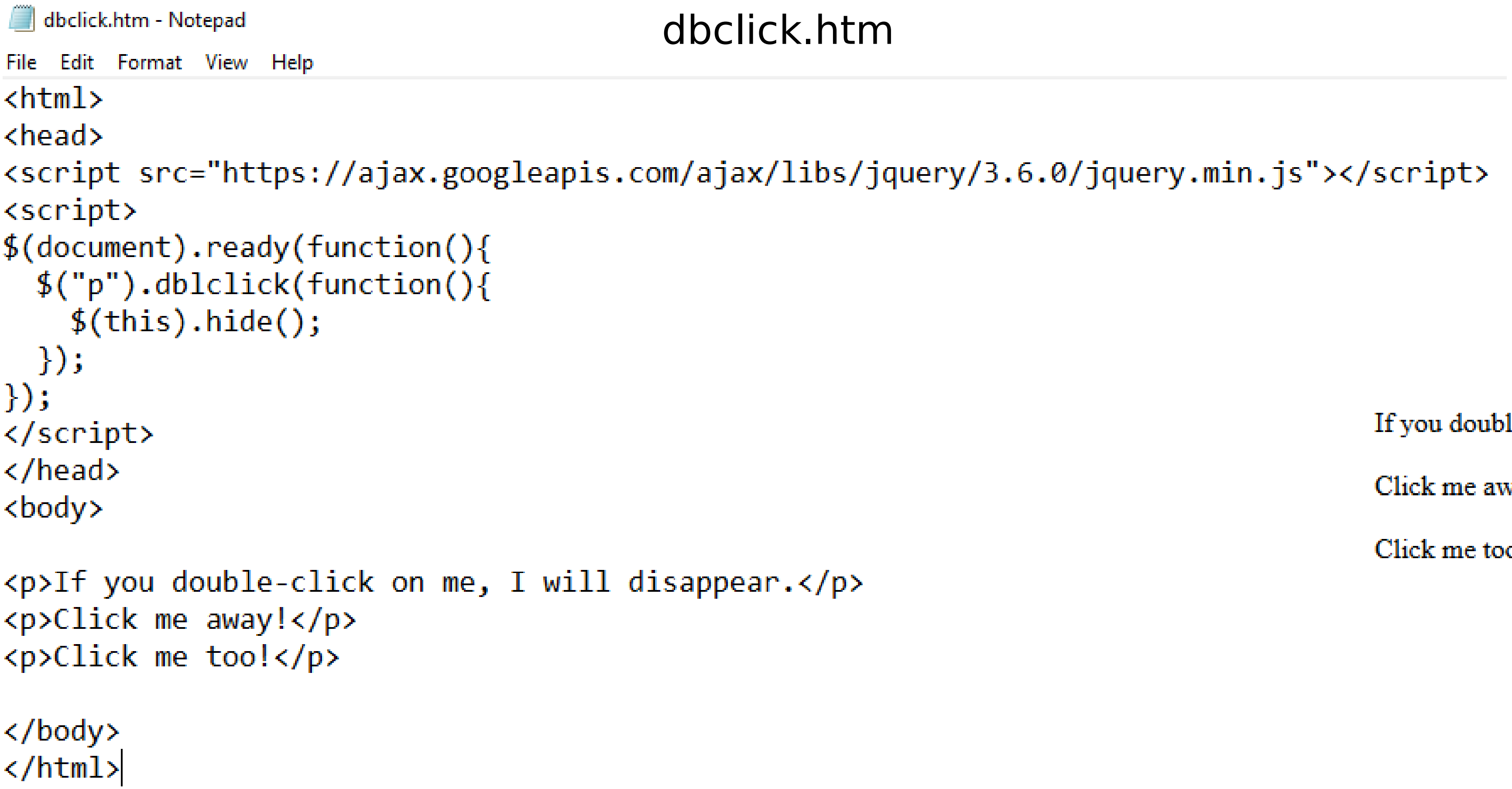
```
If you click on me, I will disappear.

Click me away!

Click me too!
```

# dblclick()

- The **dblclick()** method attaches an event handler function to an HTML element.
- The function is executed when the user double-clicks on the HTML element:

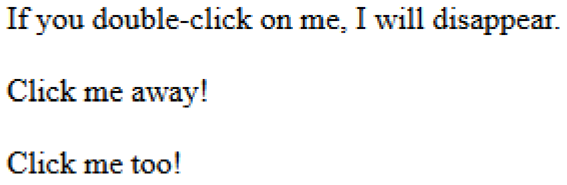
# dblclick()

dbclick.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").dblclick(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you double-click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

output



If you double-click on me, I will disappear.

Click me away!

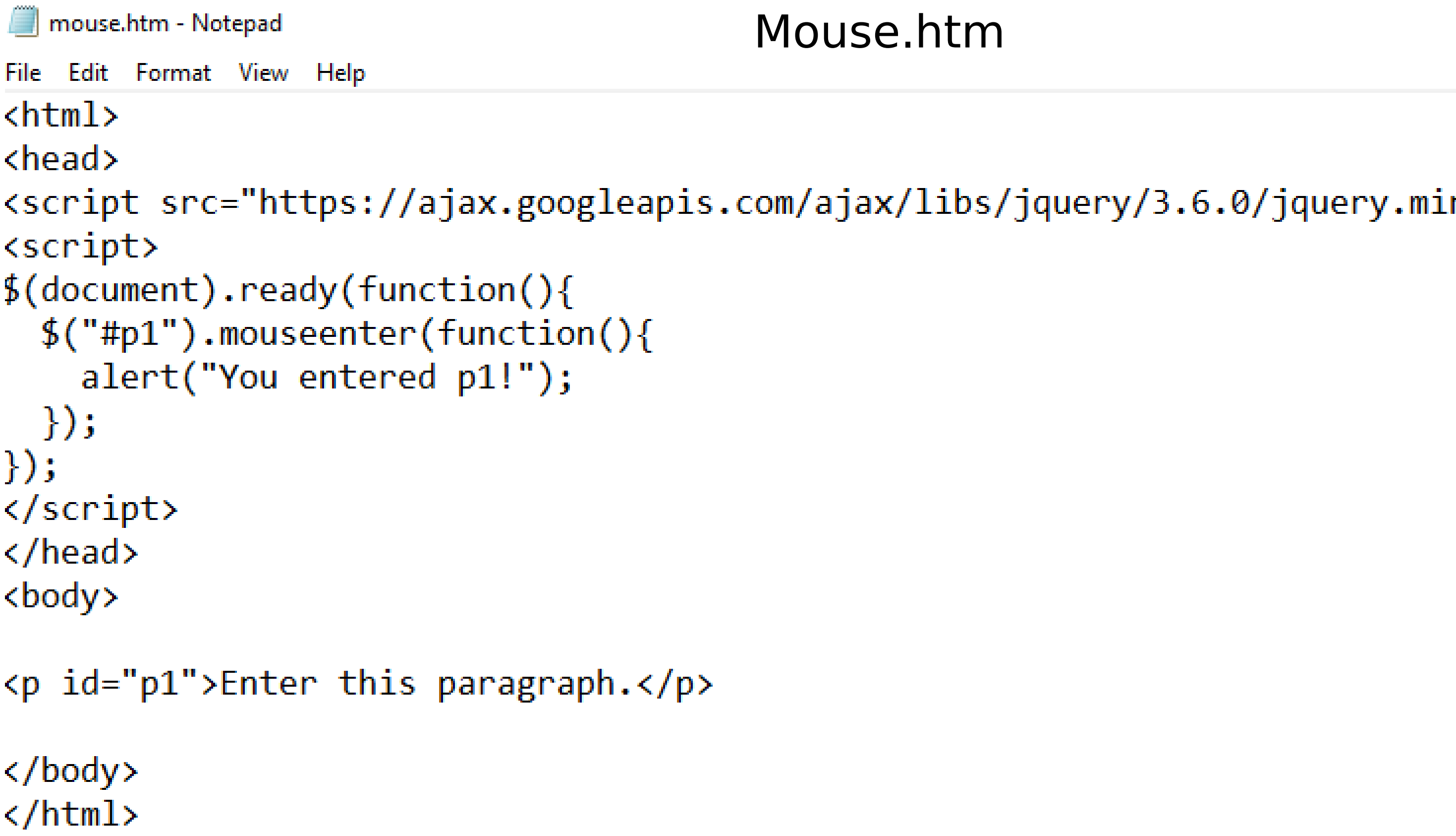
Click me too!

# mouseenter()

- The **mouseenter()** method attaches an event handler function to an HTML element.
- The function is executed when the mouse pointer enters the HTML element:

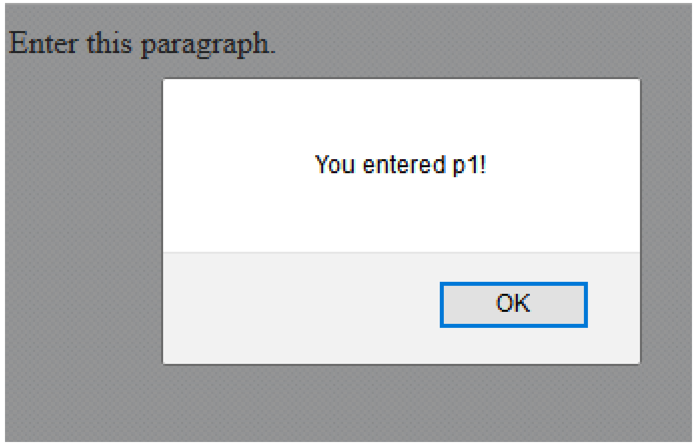
# mouseenter()

Mouse.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").mouseenter(function(){
    alert("You entered p1!");
  });
});
</script>
</head>
<body>

<p id="p1">Enter this paragraph.</p>

</body>
</html>
```

output


Enter this paragraph.

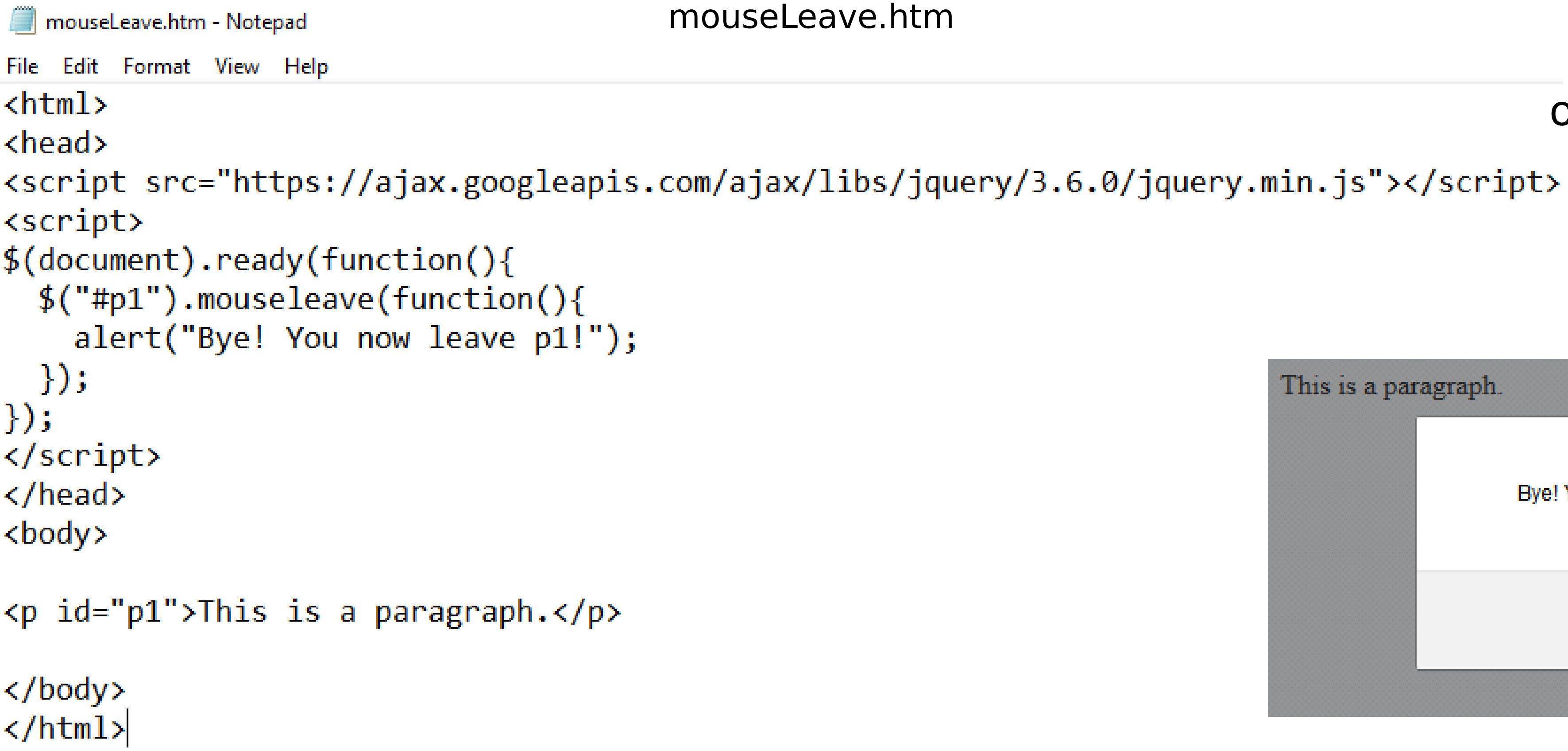
You entered p1!

# mouseleave()

- The **mouseleave()** method attaches an event handler function to an HTML element.
- The function is executed when the mouse pointer leaves the HTML element:

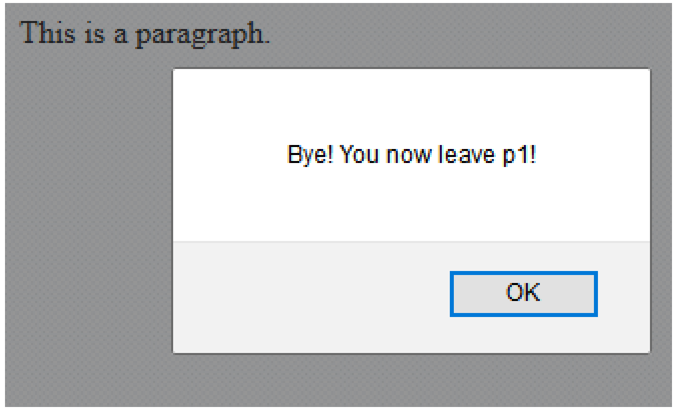
# mouseleave()

mouseLeave.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").mouseleave(function(){
    alert("Bye! You now leave p1!");
  });
});
</script>
</head>
<body>

<p id="p1">This is a paragraph.</p>

</body>
</html>
```

output



This is a paragraph.

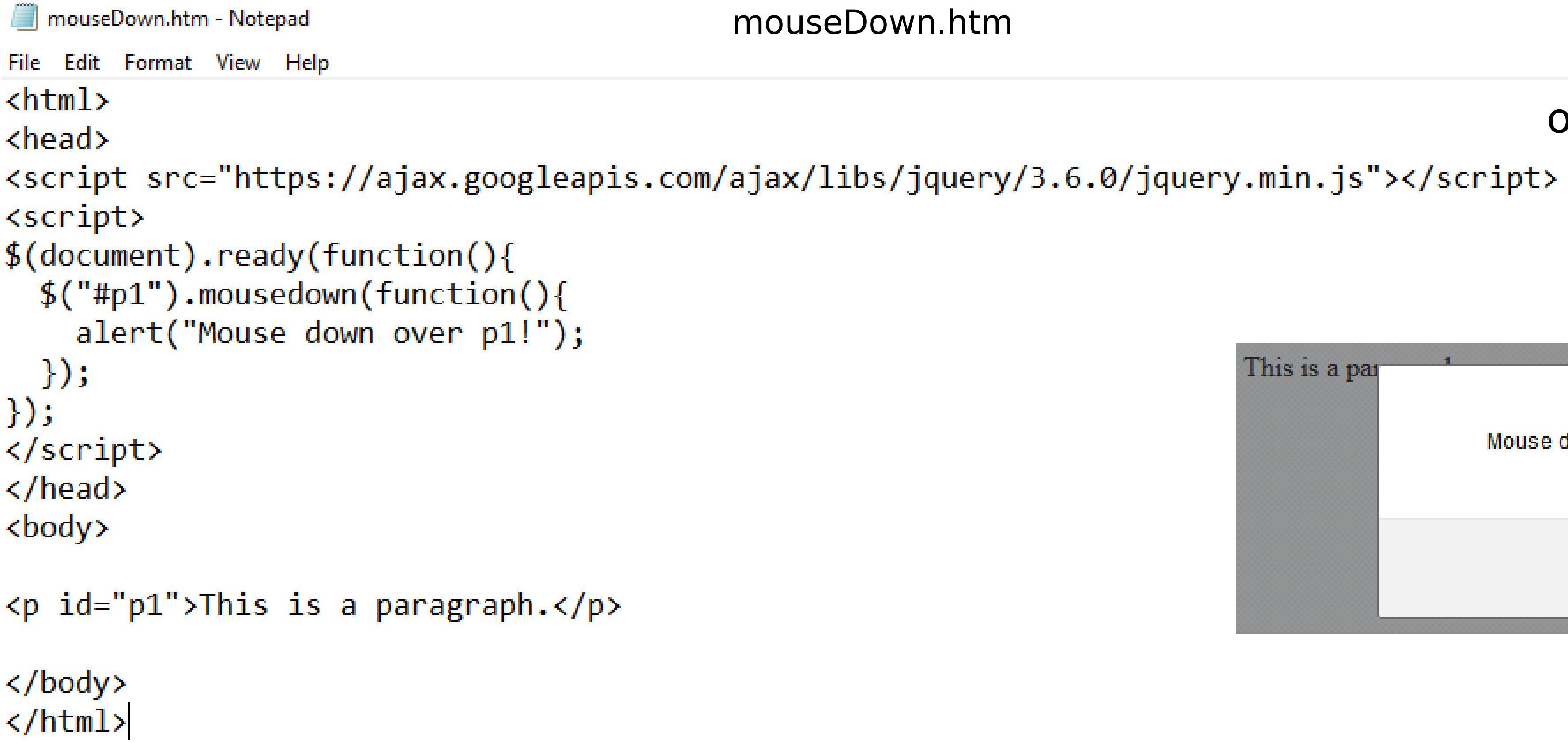
Bye! You now leave p1!

# mousedown()

- The **mousedown()** method attaches an event handler function to an HTML element.
- The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element:

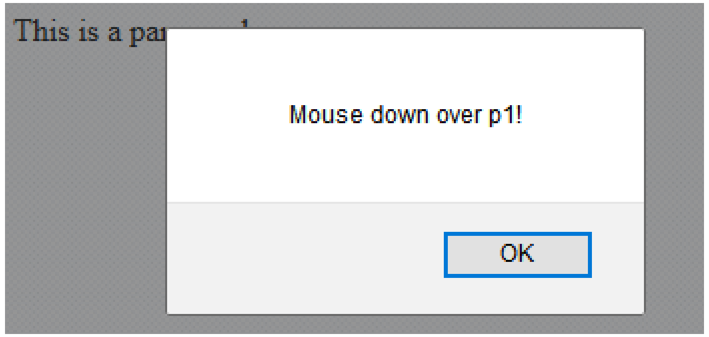
# mousedown()

mouseDown.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").mousedown(function(){
    alert("Mouse down over p1!");
  });
});
</script>
</head>
<body>

<p id="p1">This is a paragraph.</p>

</body>
</html>
```

output



This is a par

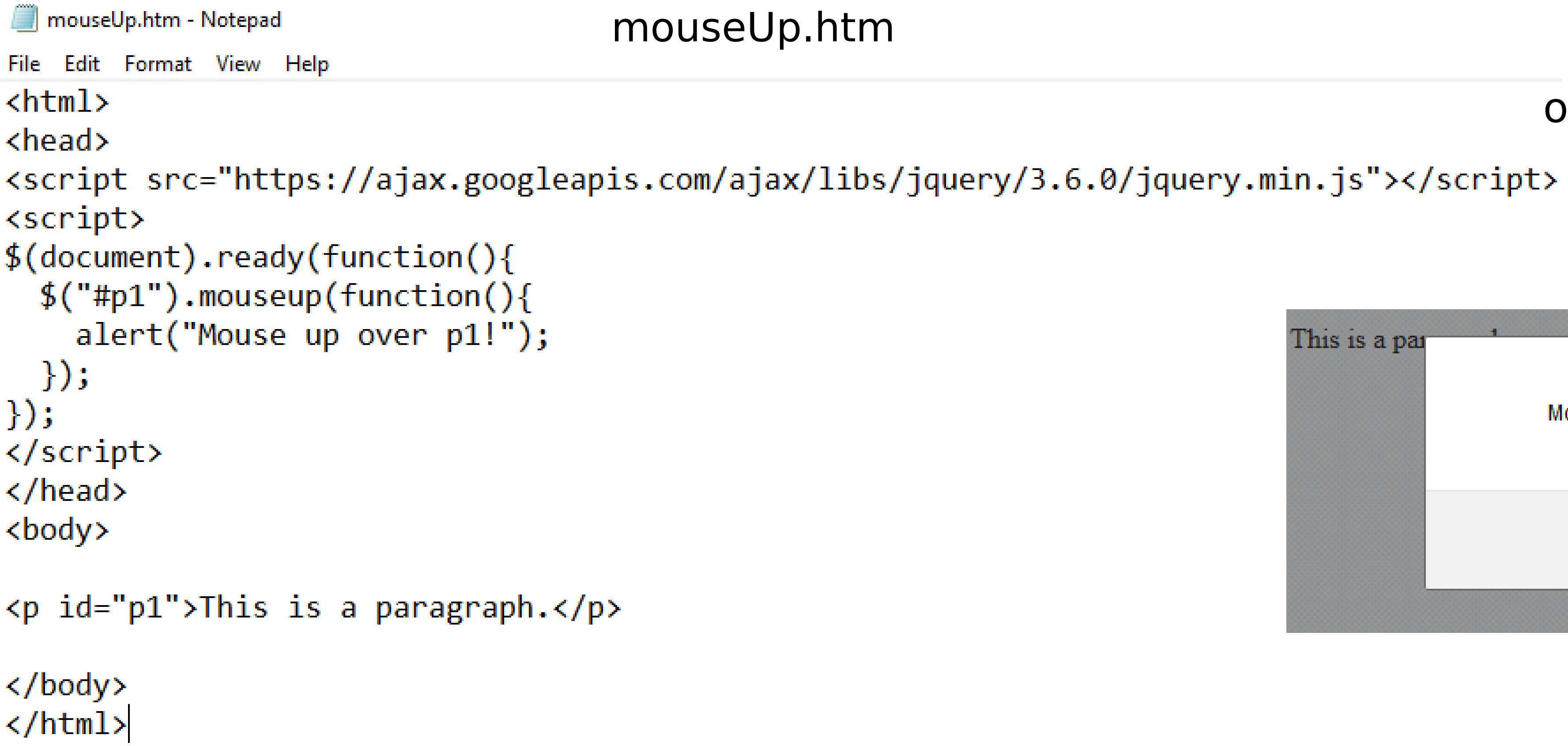
Mouse down over p1!

# mouseup()

- The **mouseup()** method attaches an event handler function to an HTML element.
- The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element:

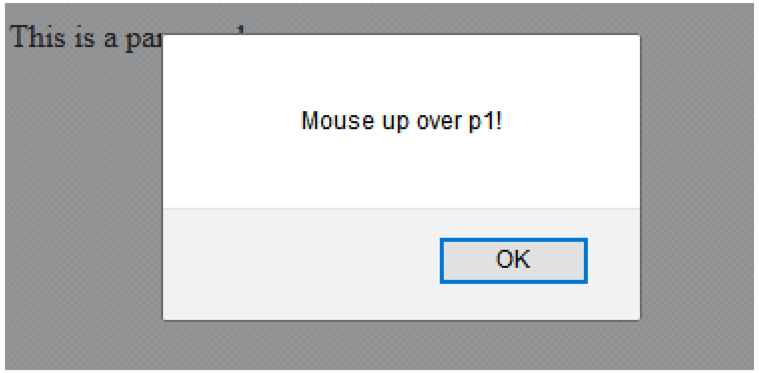
# mouseup()

mouseUp.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").mouseup(function(){
    alert("Mouse up over p1!");
  });
});
</script>
</head>
<body>

<p id="p1">This is a paragraph.</p>

</body>
</html>
```

output



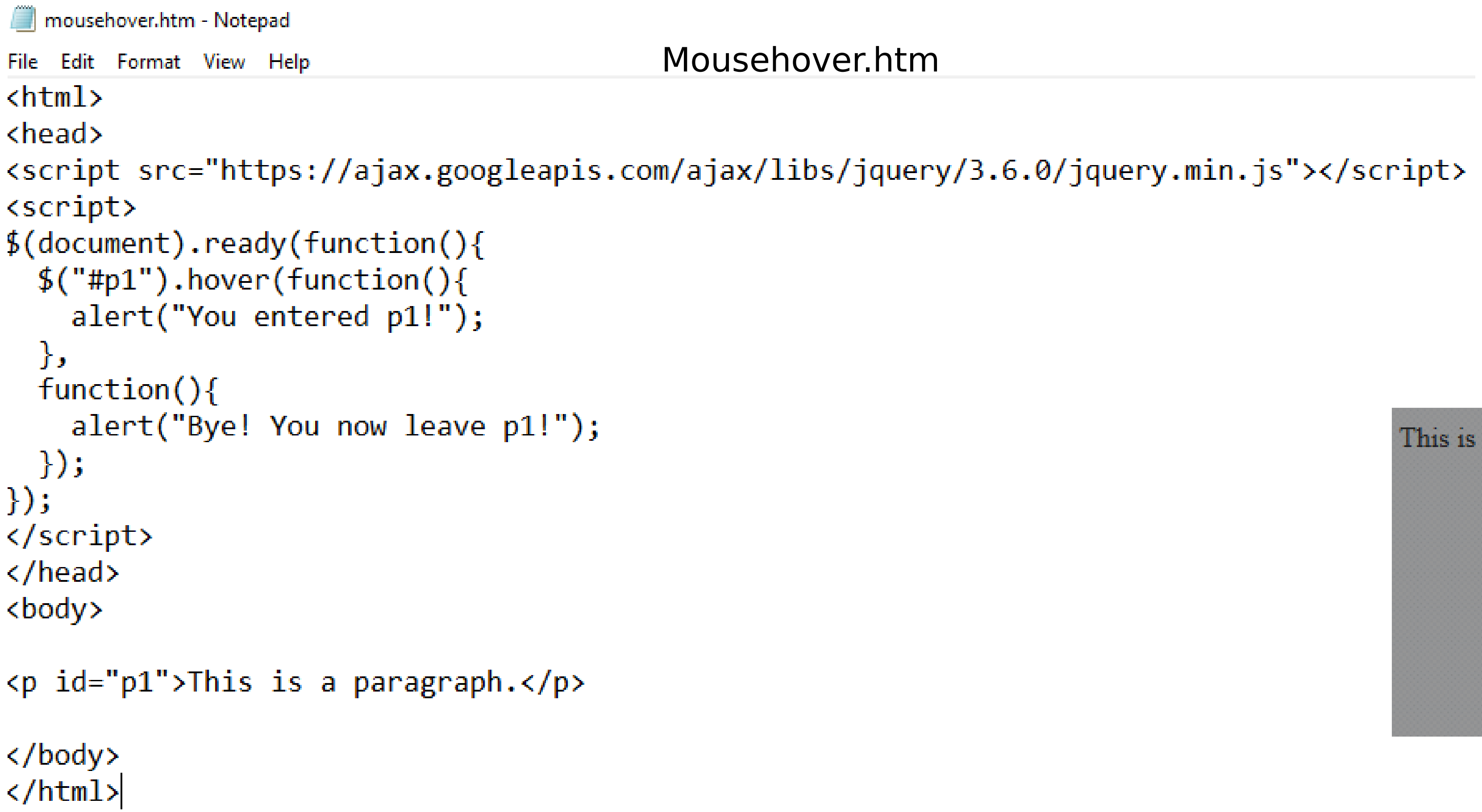
Mouse up over p1!

# hover()

- The **hover()** method takes two functions and is a combination of the
- **mouseenter()**
- **mouseleave()**
- The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element:

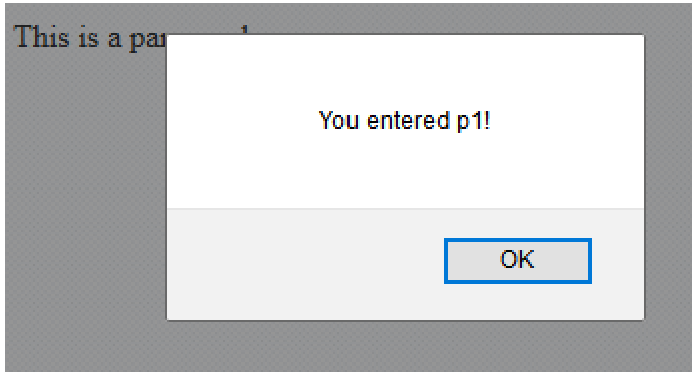
# hover()

Mousehover.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").hover(function(){
    alert("You entered p1!");
  },
  function(){
    alert("Bye! You now leave p1!");
  });
});
</script>
</head>
<body>

<p id="p1">This is a paragraph.</p>

</body>
</html>
```

output


This is a par

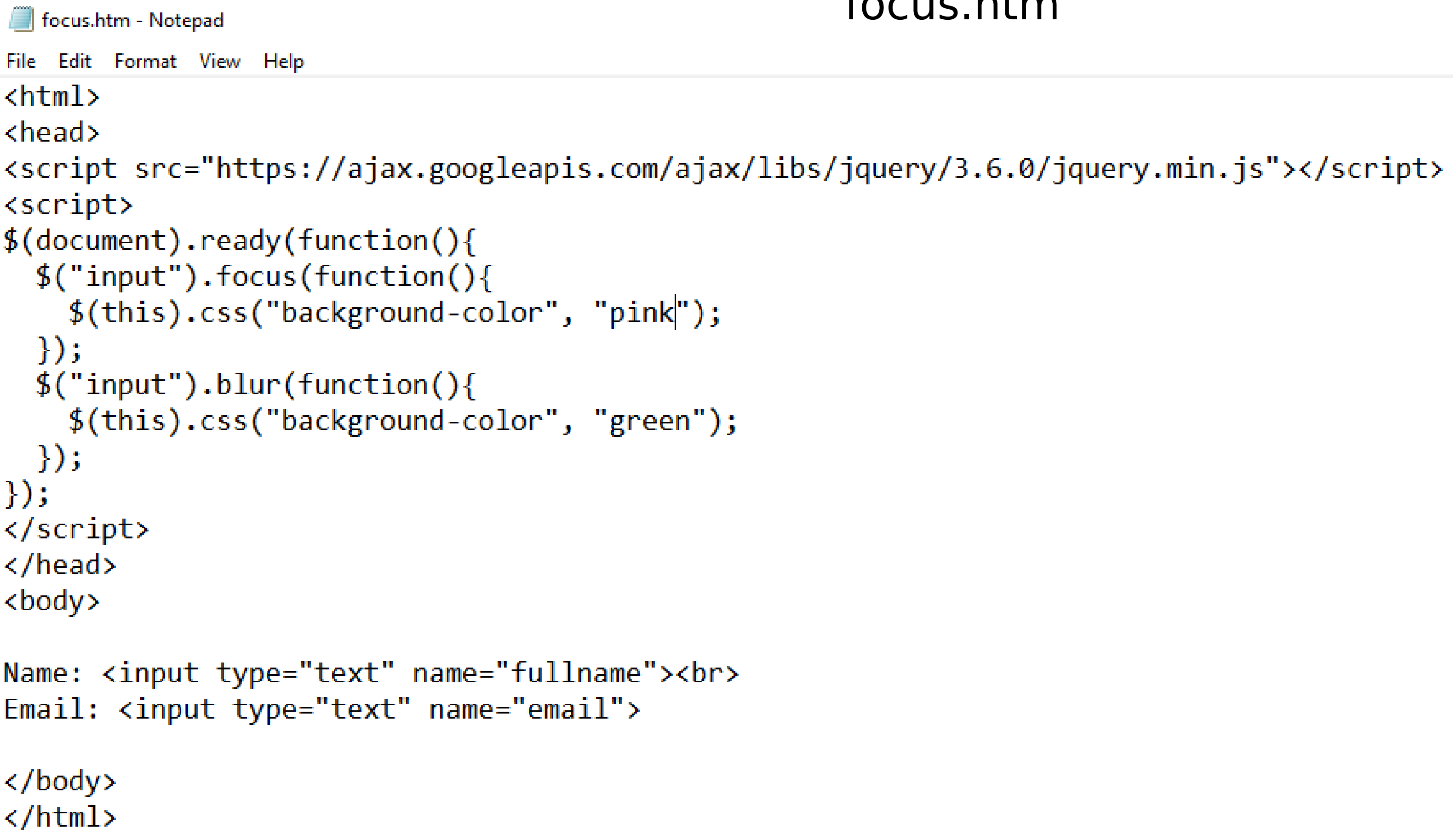
You entered p1!

# focus()

- The **focus()** method attaches an event handler function to an HTML form field.
- The function is executed when the form field gets focus:

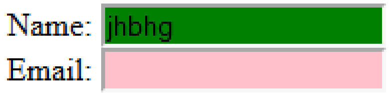
# focus()

focus.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("input").focus(function(){
    $(this).css("background-color", "pink");
  });
  $("input").blur(function(){
    $(this).css("background-color", "green");
  });
});
</script>
</head>
<body>

Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">

</body>
</html>
```

output



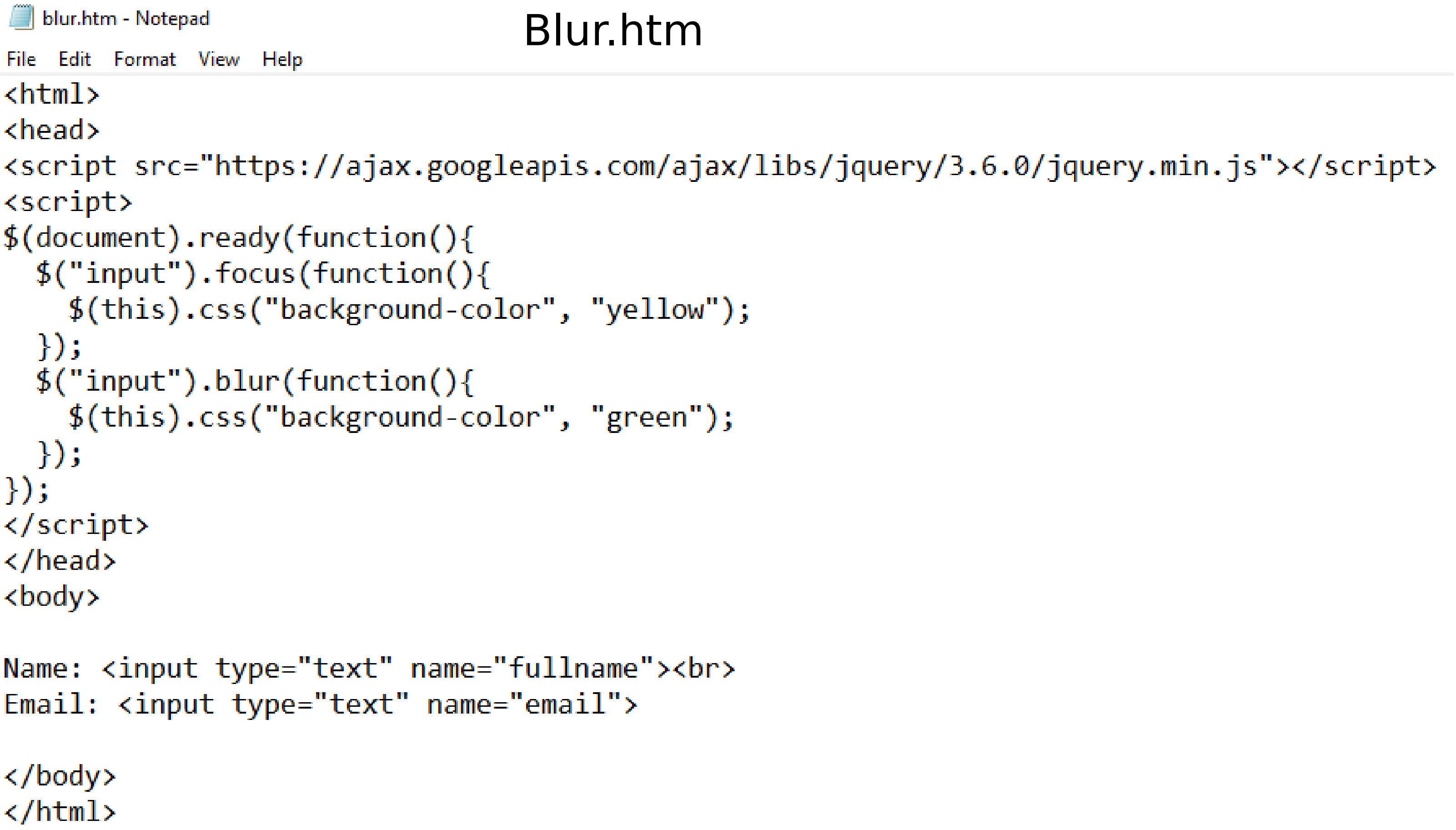
Name: jhbhg
Email:

# blur()

- The **blur()** method attaches an event handler function to an HTML form field.
- The function is executed when the form field loses focus:

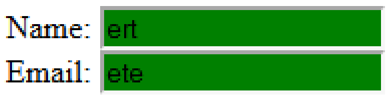
# blur()

Blur.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("input").focus(function(){
    $(this).css("background-color", "yellow");
  });
  $("input").blur(function(){
    $(this).css("background-color", "green");
  });
});
</script>
</head>
<body>

Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">

</body>
</html>
```

output

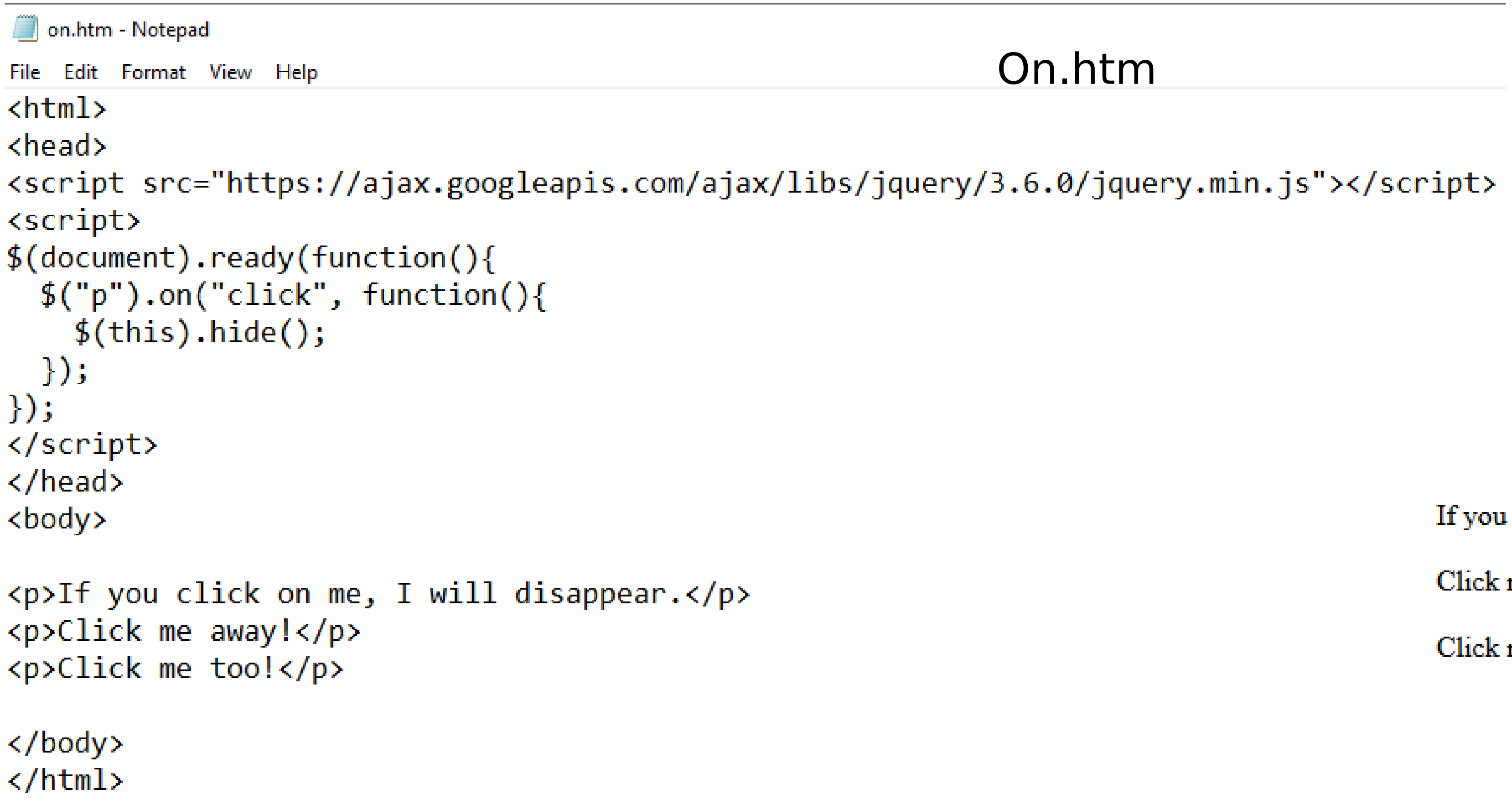


Name: ert
Email: ete

# The on() Method

- The on() method attaches one or more event handlers for the selected elements. Attach a click event to a <p> element:

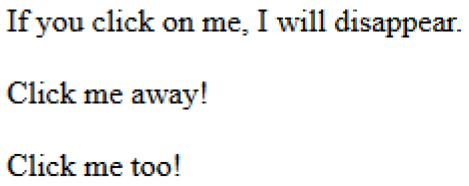
# The on() Method



On.htm

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").on("click", function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

output



If you click on me, I will disappear.

Click me away!

Click me too!

# Bachelor Of Engineering
# In
# Information Technology

# Semester Six, Third Year(Even semester)
# 25th April 2022
# Offline - 14th Lecture

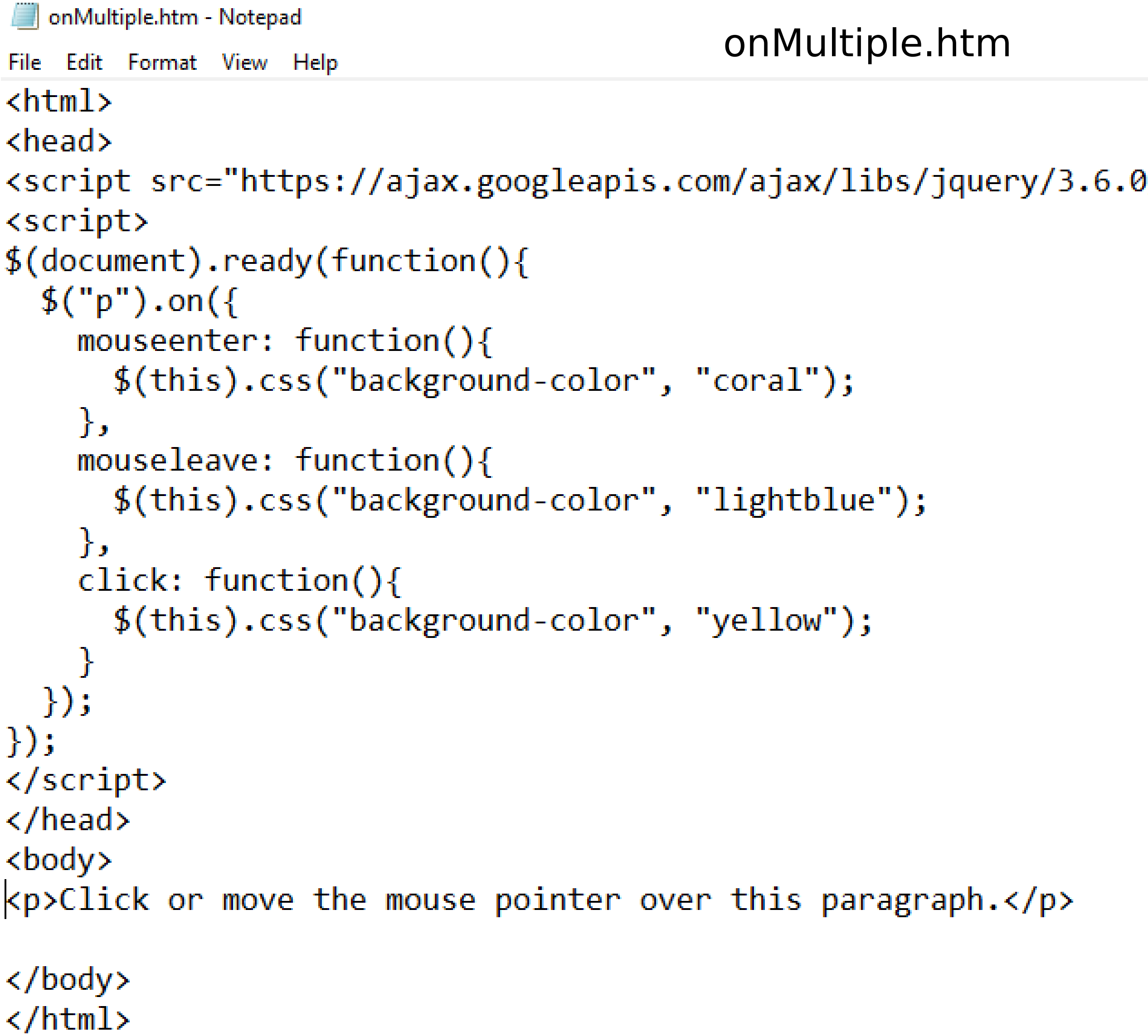
Padre Conceicao College of Engineering

Verna Goa 403722 India

# The on() Method

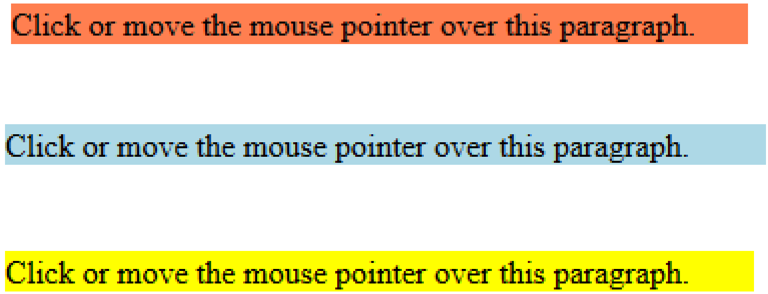
- Attach multiple event handlers to a <p> element:

onMultiple.htm



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").on({
    mouseenter: function(){
      $(this).css("background-color", "coral");
    },
    mouseleave: function(){
      $(this).css("background-color", "lightblue");
    },
    click: function(){
      $(this).css("background-color", "yellow");
    }
  });
});
</script>
</head>
<body>
<p>Click or move the mouse pointer over this paragraph.</p>

</body>
</html>
```

output



```
Click or move the mouse pointer over this paragraph.

Click or move the mouse pointer over this paragraph.

Click or move the mouse pointer over this paragraph.
```

# jQuery Effects

- jQuery Hide/Show
- jQuery Fade
- jQuery Slide
- jQuery Animate
- jQuery stop()
- jQuery Callback
- jQuery Chaining

# jQuery Effect Methods

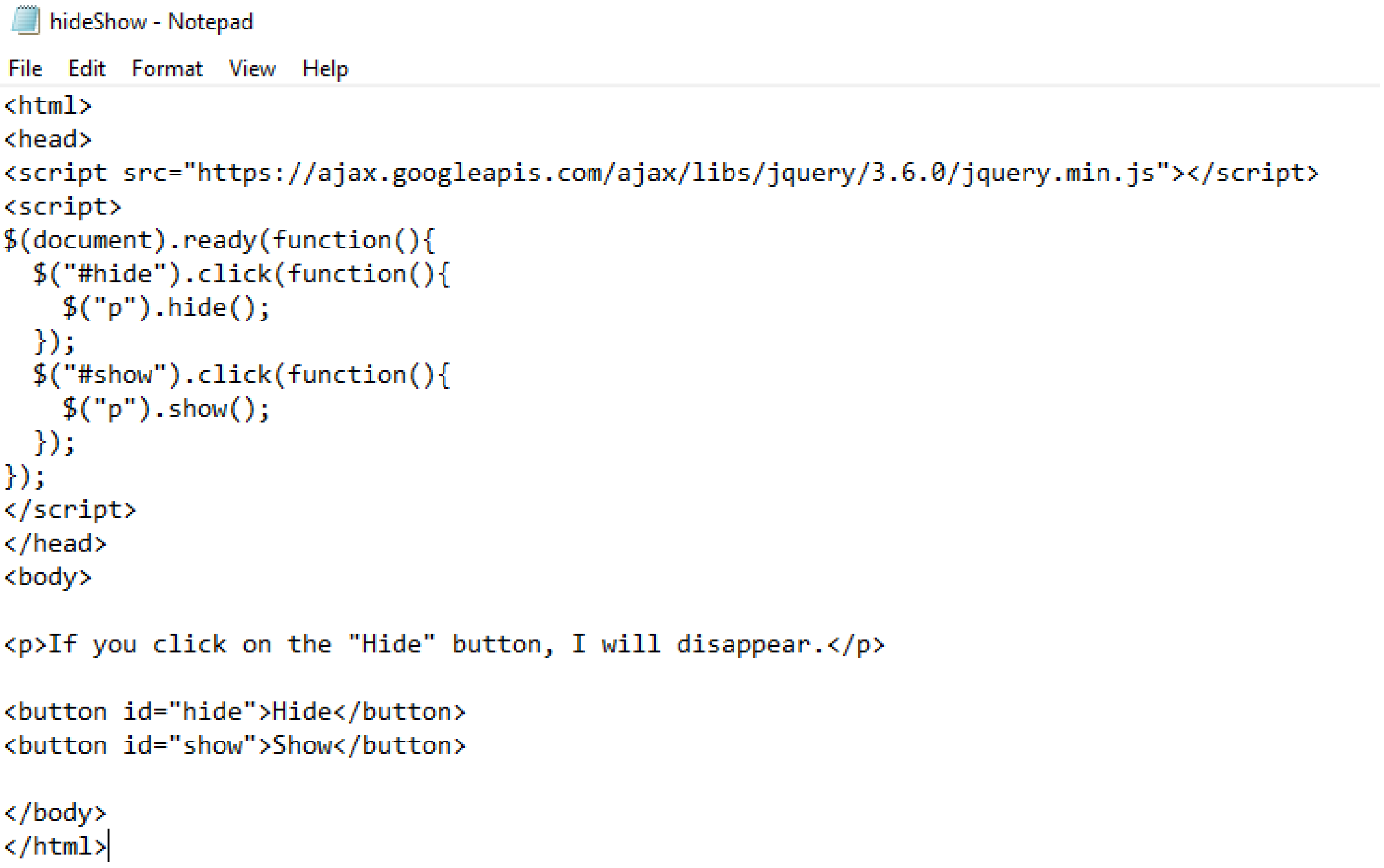
- The following table lists all the jQuery methods for creating animation effects.

| Method | Description |
|---|---|
| animate() | Runs a custom animation on the selected elements |
| clearQueue() | Removes all remaining queued functions from the selected elements |
| delay() | Sets a delay for all queued functions on the selected elements |
| dequeue() | Removes the next function from the queue, and then executes the function |
| fadeIn() | Fades in the selected elements |

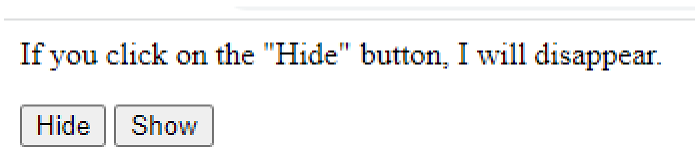
# jQuery Effect Methods

| Method | Description |
|---|---|
| fadeTo() | Fades in/out the selected elements to a given opacity |
| fadeToggle() | Toggles between the fadeIn() and fadeOut() methods |
| finish() | Stops, removes and completes all queued animations for the selected elements |
| hide() | Hides the selected elements |
| queue() | Shows the queued functions on the selected elements |
| show() | Shows the selected elements |
| slideDown() | Slides-down (shows) the selected elements |
| slideToggle() | Toggles between the slideUp() and slideDown() methods |
| slideUp() | Slides-up (hides) the selected elements |
| stop() | Stops the currently running animation for the selected elements |
| toggle() | Toggles between the hide() and show() methods |

# jQuery hide() and show()



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#hide").click(function(){
    $("p").hide();
  });
  $("#show").click(function(){
    $("p").show();
  });
});
</script>
</head>
<body>

<p>If you click on the "Hide" button, I will disappear.</p>

<button id="hide">Hide</button>
<button id="show">Show</button>

</body>
</html>
```



If you click on the "Hide" button, I will disappear.

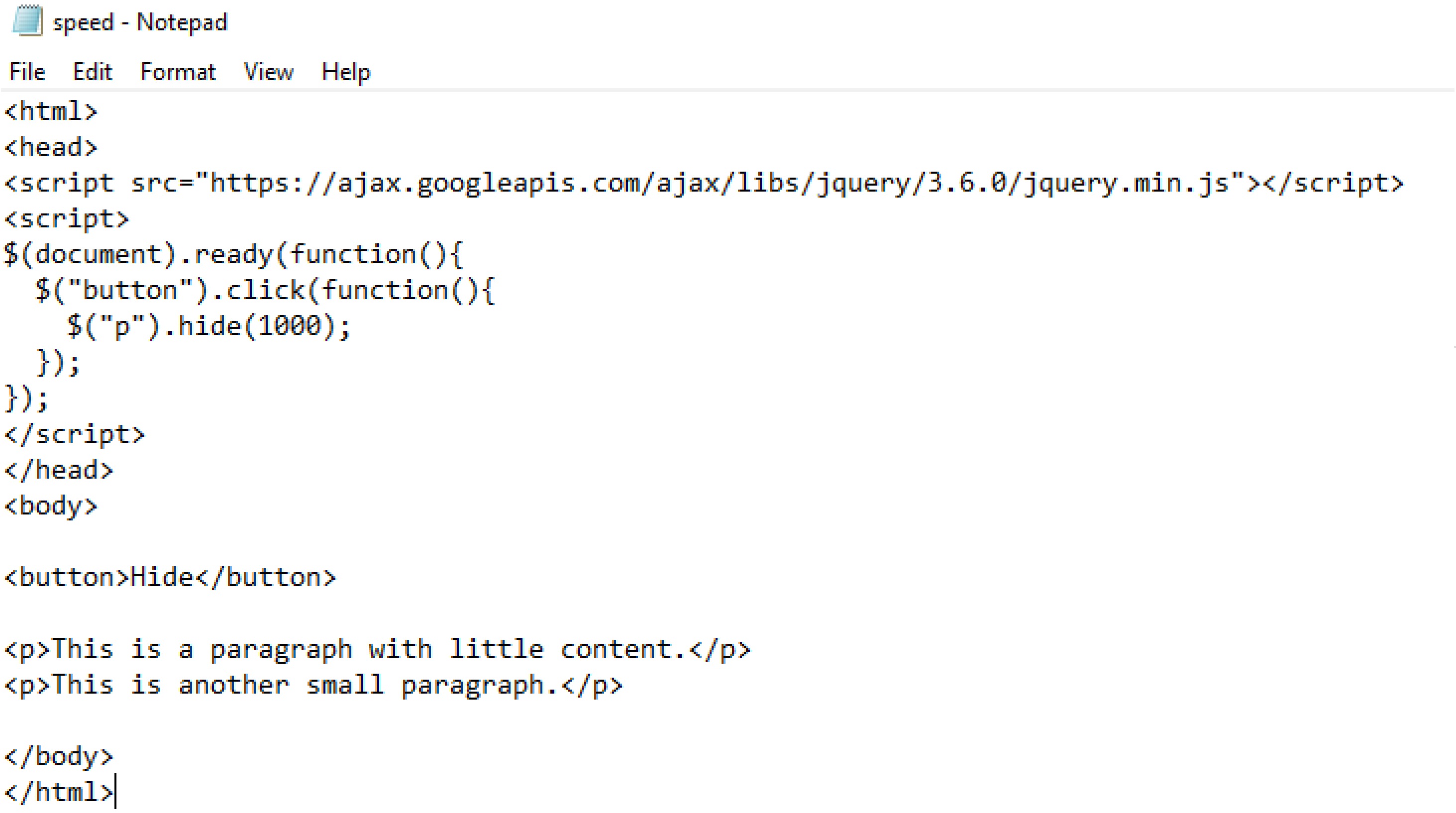
# jQuery hide() and show()

- With jQuery, you can hide and show HTML elements with the hide() and show() methods
- The optional speed parameter specifies the speed of the hiding/showing, and can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the hide or show method completes

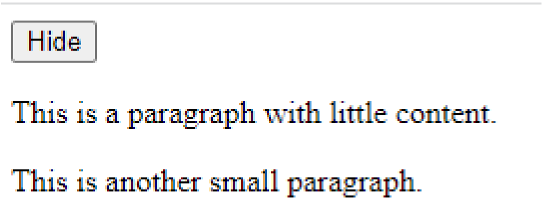
# Syntax

```
$(selector).hide(speed,callback);

$(selector).show(speed,callback);
```

# speed parameter with hide ()



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide(1000);
  });
});
</script>
</head>
<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>
<p>This is another small paragraph.</p>

</body>
</html>
```



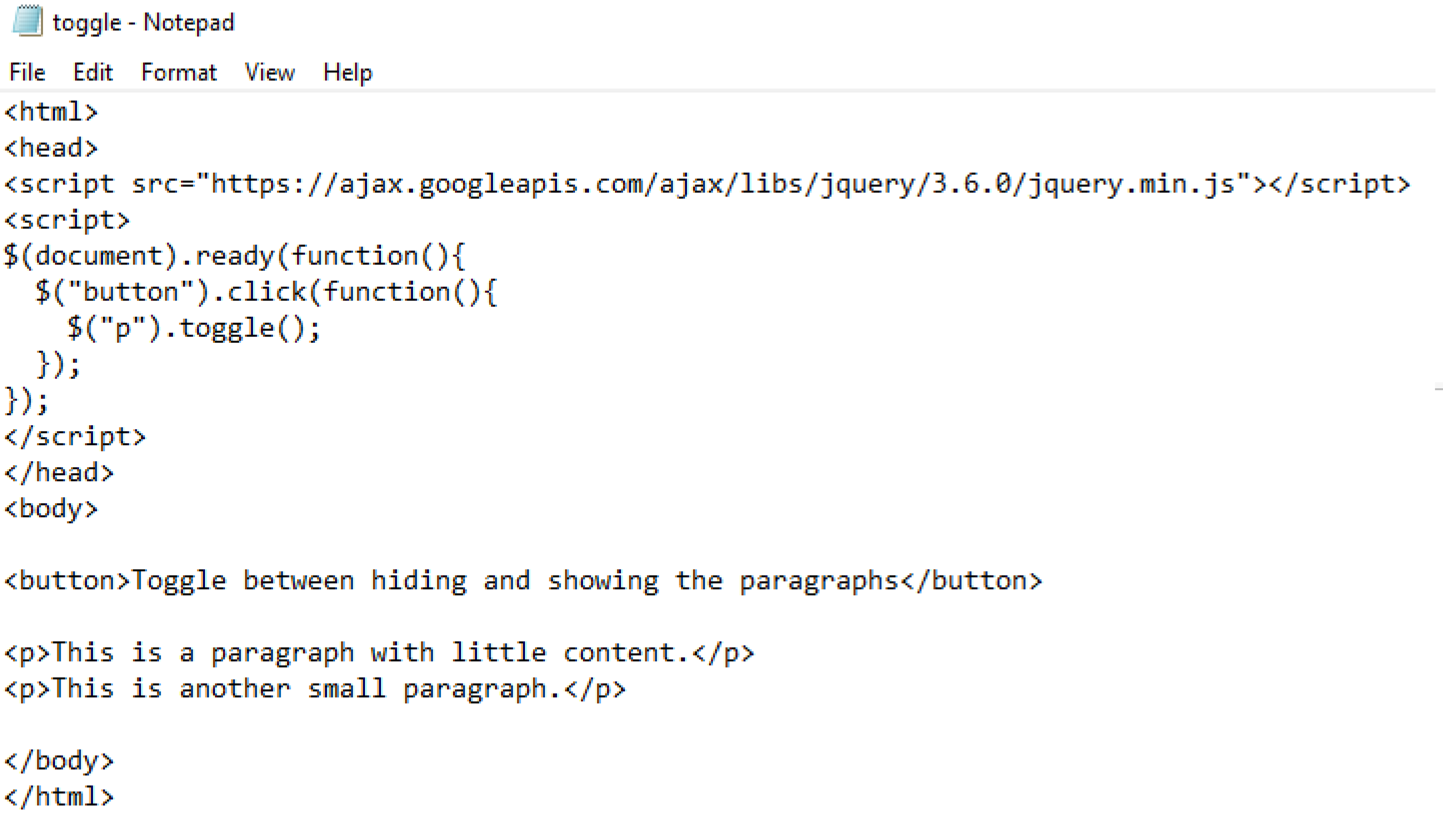
This is a paragraph with little content.

This is another small paragraph.

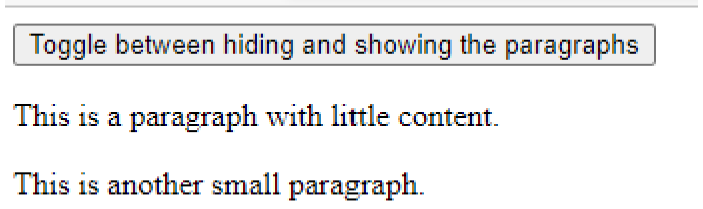
# jQuery toggle()

- You can also toggle between hiding and showing an element with the toggle() method.
- Shown elements are hidden and hidden elements are shown
- The optional speed parameter can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after toggle() completes

# jQuery toggle()



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").toggle();
  });
});
</script>
</head>
<body>

<button>Toggle between hiding and showing the paragraphs</button>

<p>This is a paragraph with little content.</p>
<p>This is another small paragraph.</p>

</body>
</html>
```



This is a paragraph with little content.

This is another small paragraph.

# syntax

$(*selector*).toggle(*speed,callback*);

# jQuery Effects - Fading

jQuery Fading Methods:

- With jQuery you can fade an element in and out of visibility.
- jQuery has the following fade methods:

1. fadeIn()
2. fadeOut()
3. fadeToggle()
4. fadeTo()

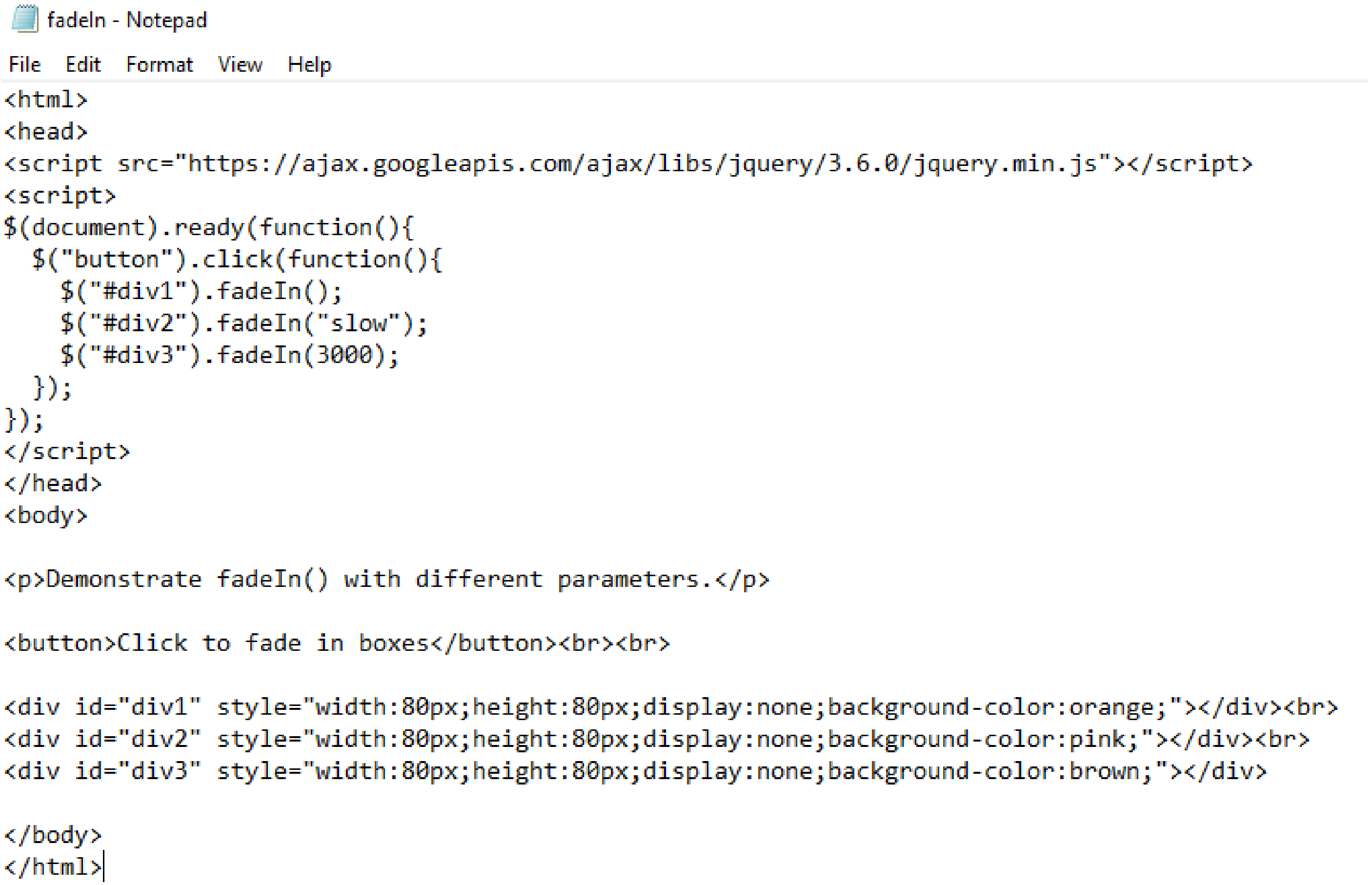
# jQuery fadeIn() Method

- The jQuery fa
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the fading completes.
- deIn() Method is used to fade in a hidden element.

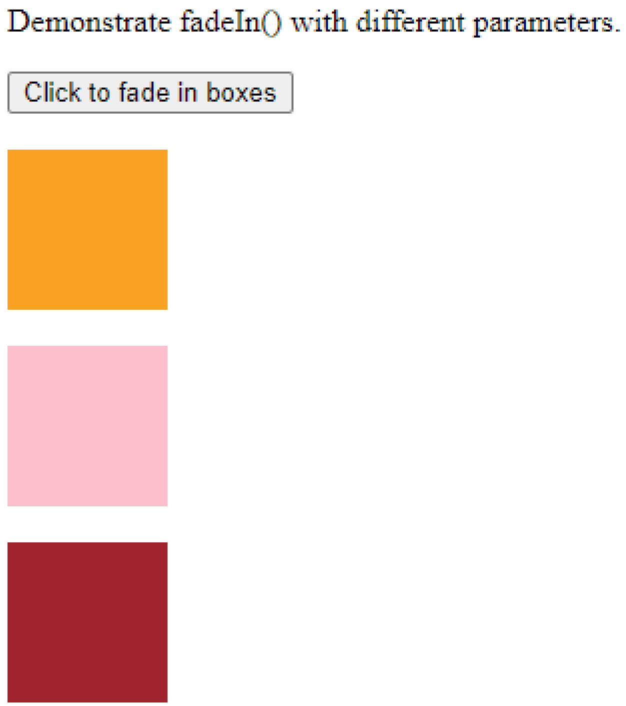
# Syntax

$(*selector*).fadeIn(*speed,callback*);

# jQuery fadeIn()



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeIn();
    $("#div2").fadeIn("slow");
    $("#div3").fadeIn(3000);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeIn() with different parameters.</p>

<button>Click to fade in boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-color:orange;"></div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-color:pink;"></div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:brown;"></div>

</body>
</html>
```



Demonstrate fadeIn() with different parameters.

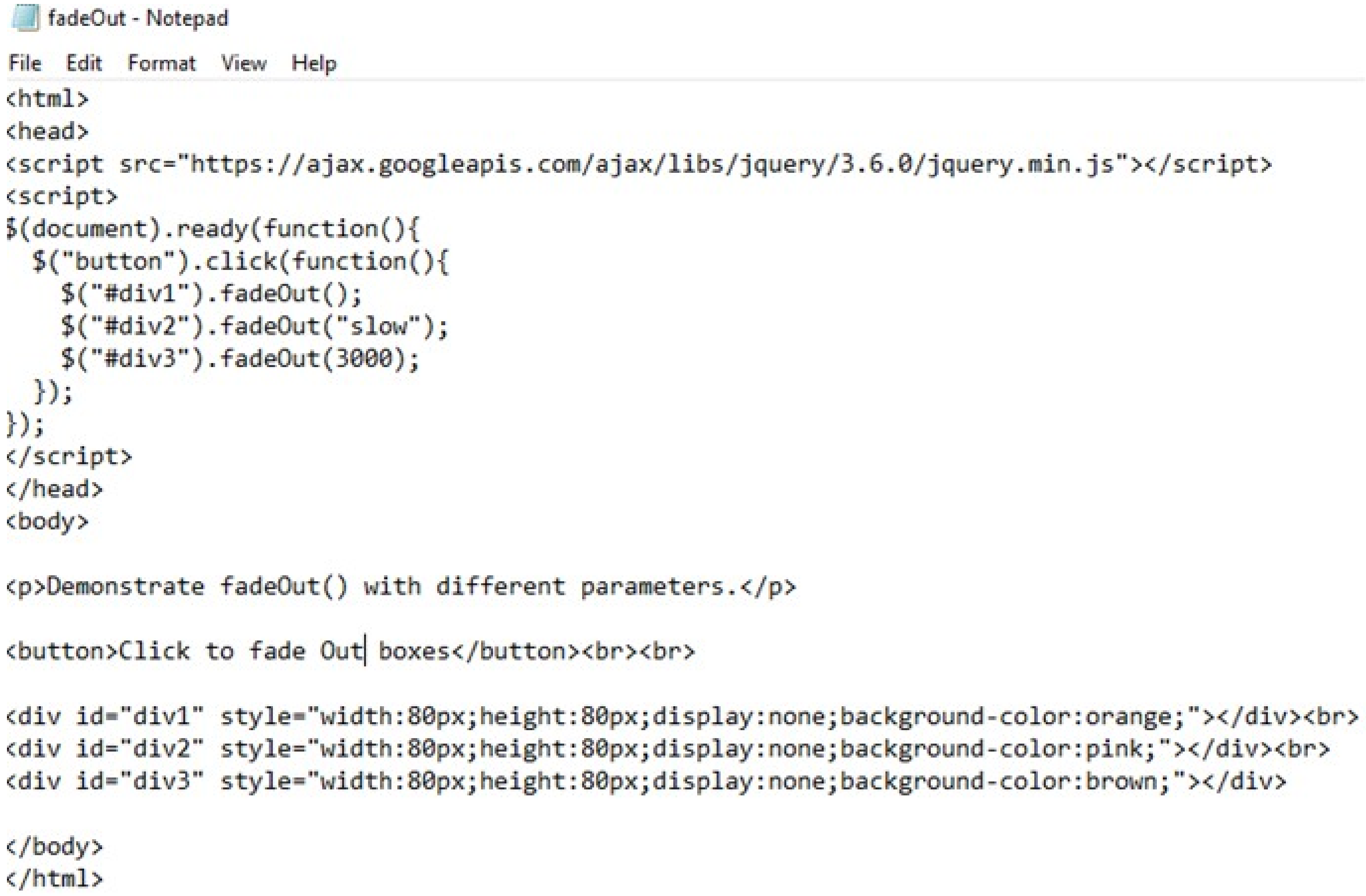
# jQuery fadeOut() Method

- The jQuery fadeOut() Method is used to fade out a visible element.
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the fading completes.

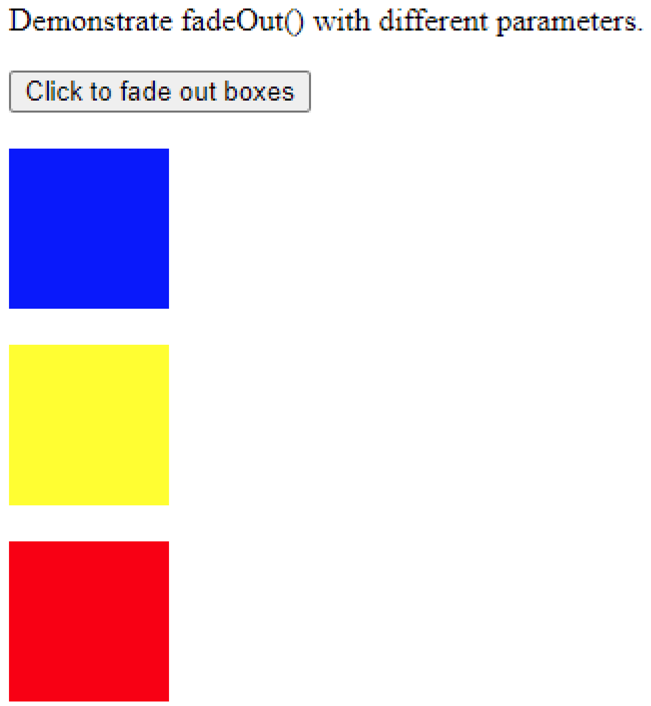
# Syntax

```
$(selector).fadeOut(speed,callback);
```

# jQuery fadeOut() Method



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeOut();
    $("#div2").fadeOut("slow");
    $("#div3").fadeOut(3000);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeOut() with different parameters.</p>

<button>Click to fade Out boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-color:orange;"></div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-color:pink;"></div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:brown;"></div>

</body>
</html>
```


Demonstrate fadeOut() with different parameters.

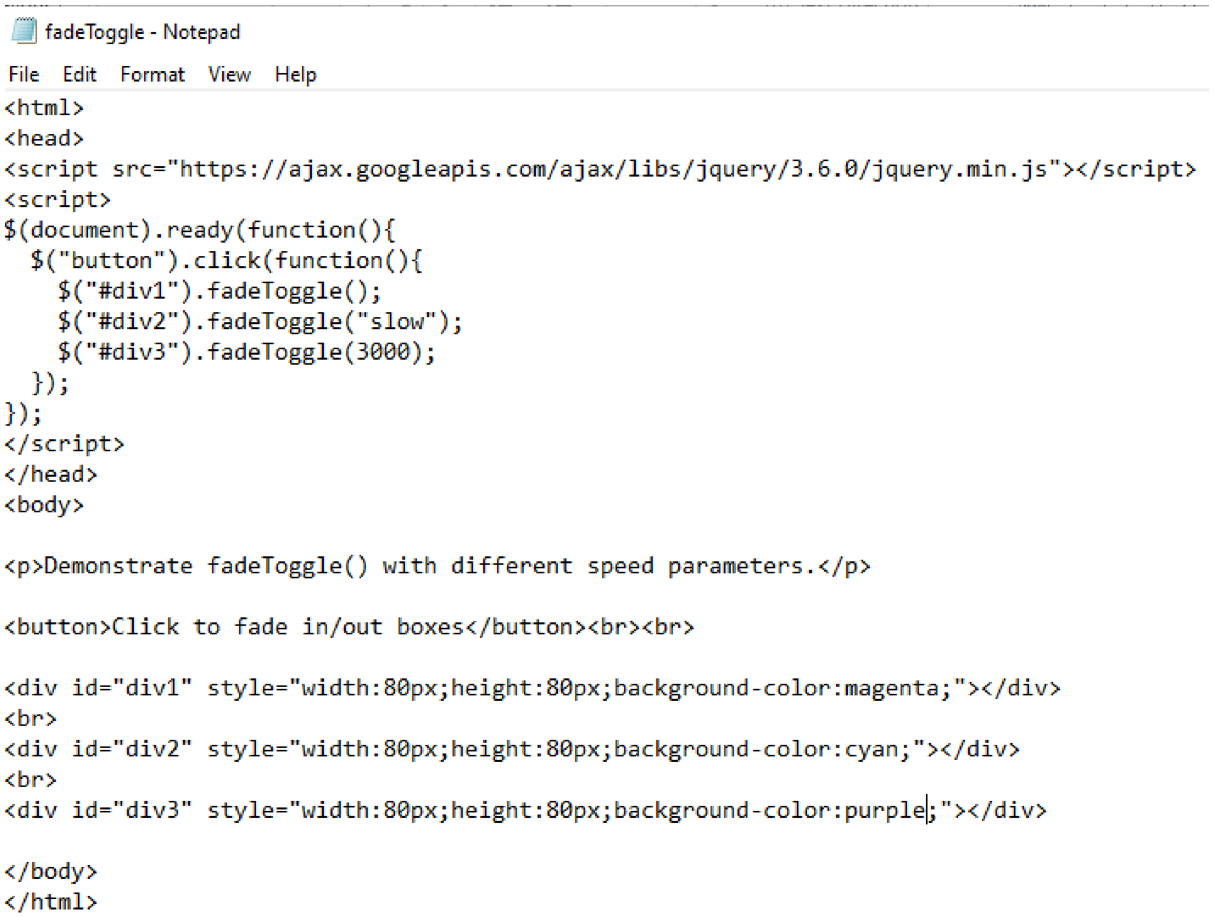
# jQuery fadeToggle() Method

- The jQuery fadeToggle() Method toggles between fadeIn() and fadeout() method.
- If the elements are faded out, fadeToggle()  will fade them in
- If the elements are faded in, fadeToggle()  will fade them out
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the fading completes.

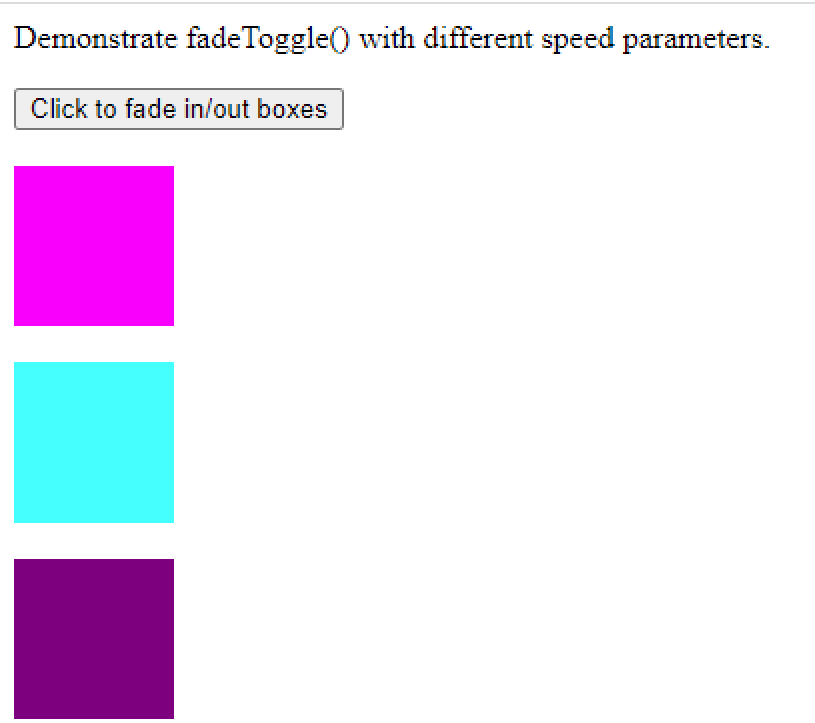
# Syntax

```
$(selector).fadeToggle(speed,callback);
```

# Demonstration of jQuery fadeToggle()



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeToggle();
    $("#div2").fadeToggle("slow");
    $("#div3").fadeToggle(3000);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeToggle() with different speed parameters.</p>

<button>Click to fade in/out boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;background-color:magenta;"></div>
<br>
<div id="div2" style="width:80px;height:80px;background-color:cyan;"></div>
<br>
<div id="div3" style="width:80px;height:80px;background-color:purple;"></div>

</body>
</html>
```



Demonstrate fadeToggle() with different speed parameters.

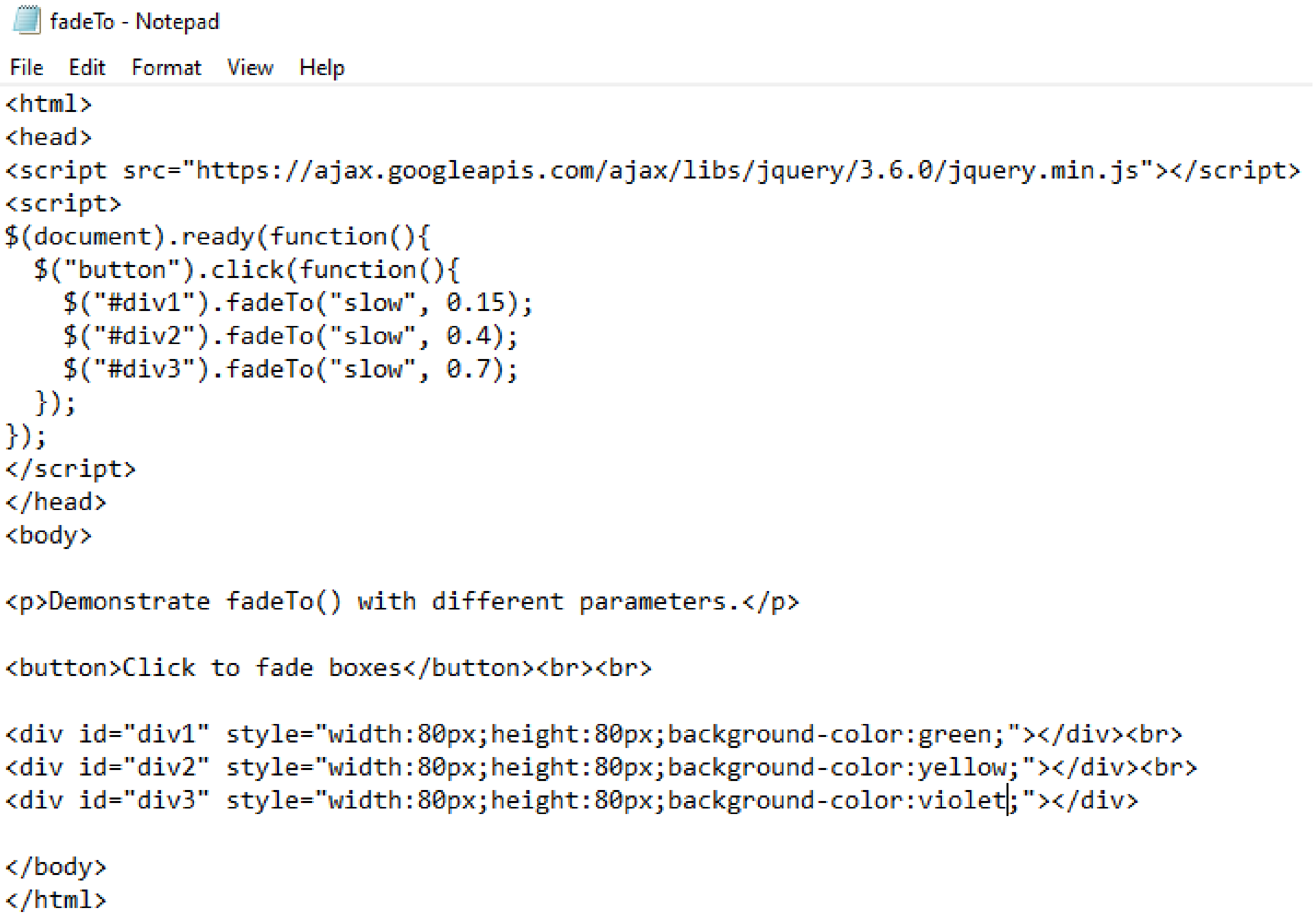
# jQuery fadeTo() Method

- The jQuery fadeTo() Method allows fading to a given opacity (value between 0 and 1).
- The required speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The required opacity parameter in the fadeTo() method specifies fading to a given opacity (value between 0 and 1).
- The optional callback parameter is a function to be executed after the function completes.

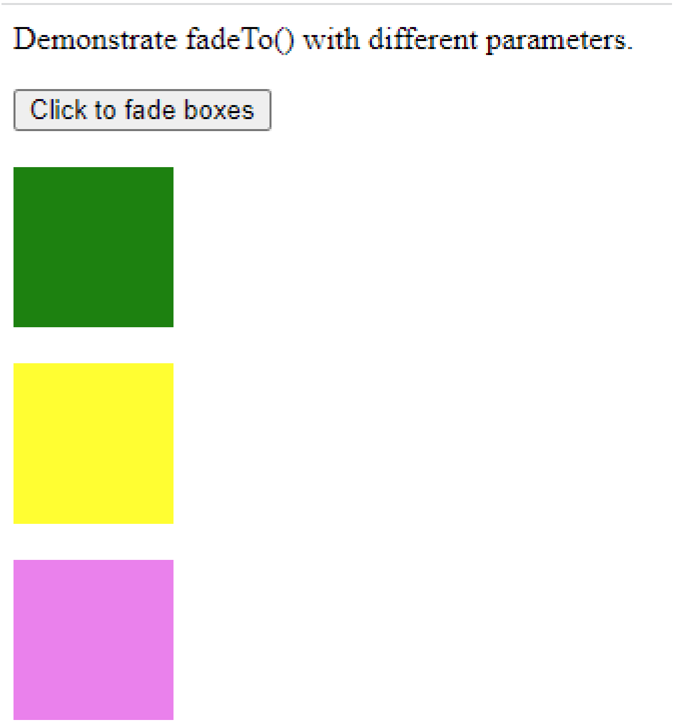
# Syntax

```
$(selector).fadeTo(speed,opacity,callback);
```

# Demonstration of fadeTo() method with different parameters:



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeTo("slow", 0.15);
    $("#div2").fadeTo("slow", 0.4);
    $("#div3").fadeTo("slow", 0.7);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeTo() with different parameters.</p>

<button>Click to fade boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:yellow;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:violet;"></div>

</body>
</html>
```



Demonstrate fadeTo() with different parameters.

# jQuery Effects - Sliding

- The jQuery slide methods slide elements up and down.

jQuery Sliding Methods

- With jQuery you can create a sliding effect on elements.
- jQuery has the following slide methods:

1. slideDown()
2. slideUp()
3. slideToggle()

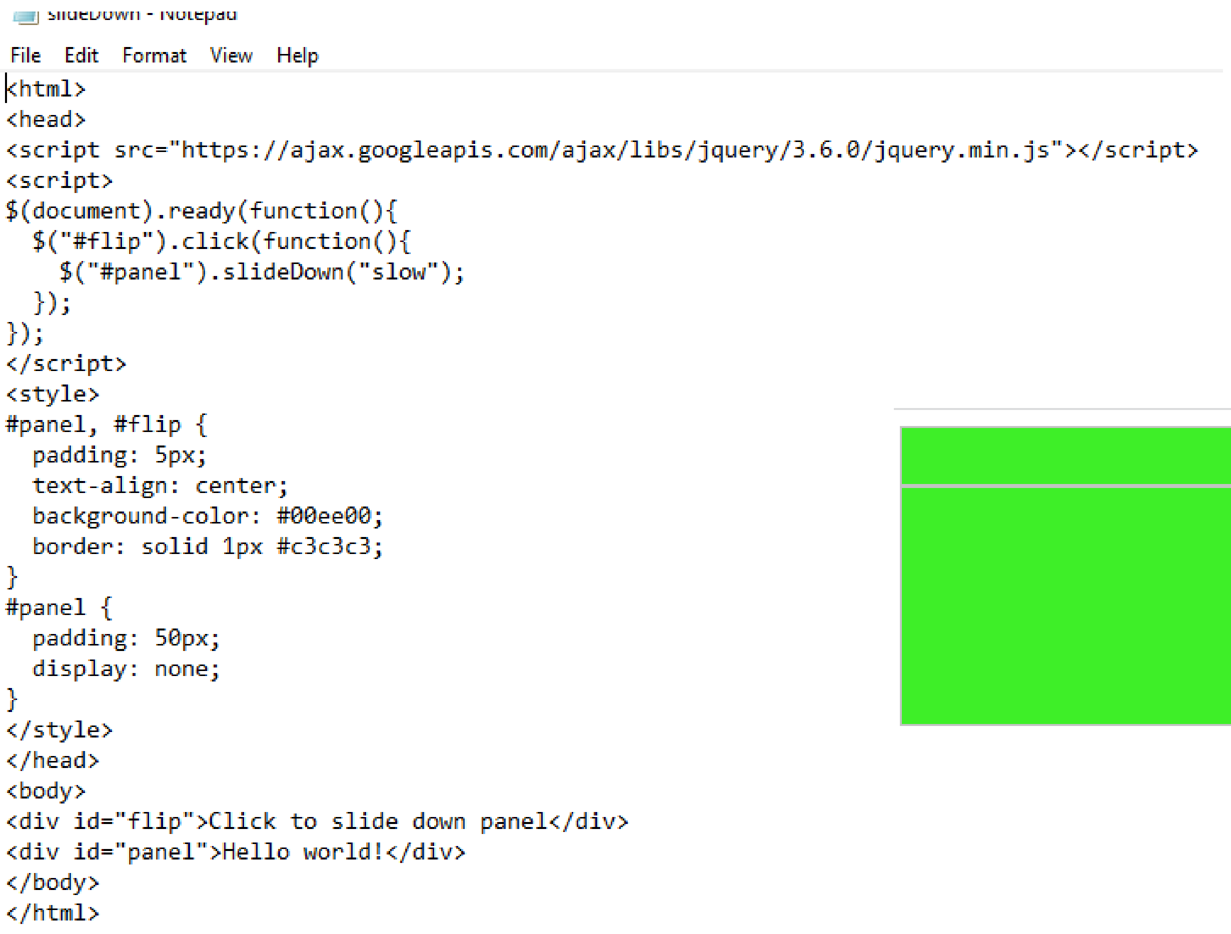
# jQuery slideDown() Method

- The jQuery slideDown() Method is used to slide down an element.
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the sliding completes.

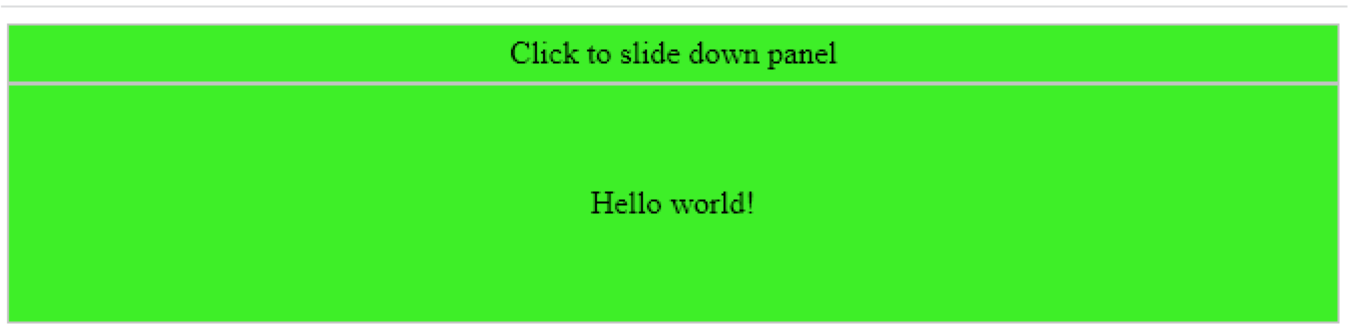
# Sytnax

```
$(selector).slideDown(speed,callback);
```

# Demonstration of jQuery slideDown() Method



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideDown("slow");
  });
});
</script>
<style>
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: #00ee00;
  border: solid 1px #c3c3c3;
}
#panel {
  padding: 50px;
  display: none;
}
</style>
</head>
<body>
<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>
</body>
</html>
```



Click to slide down panel

Hello world!

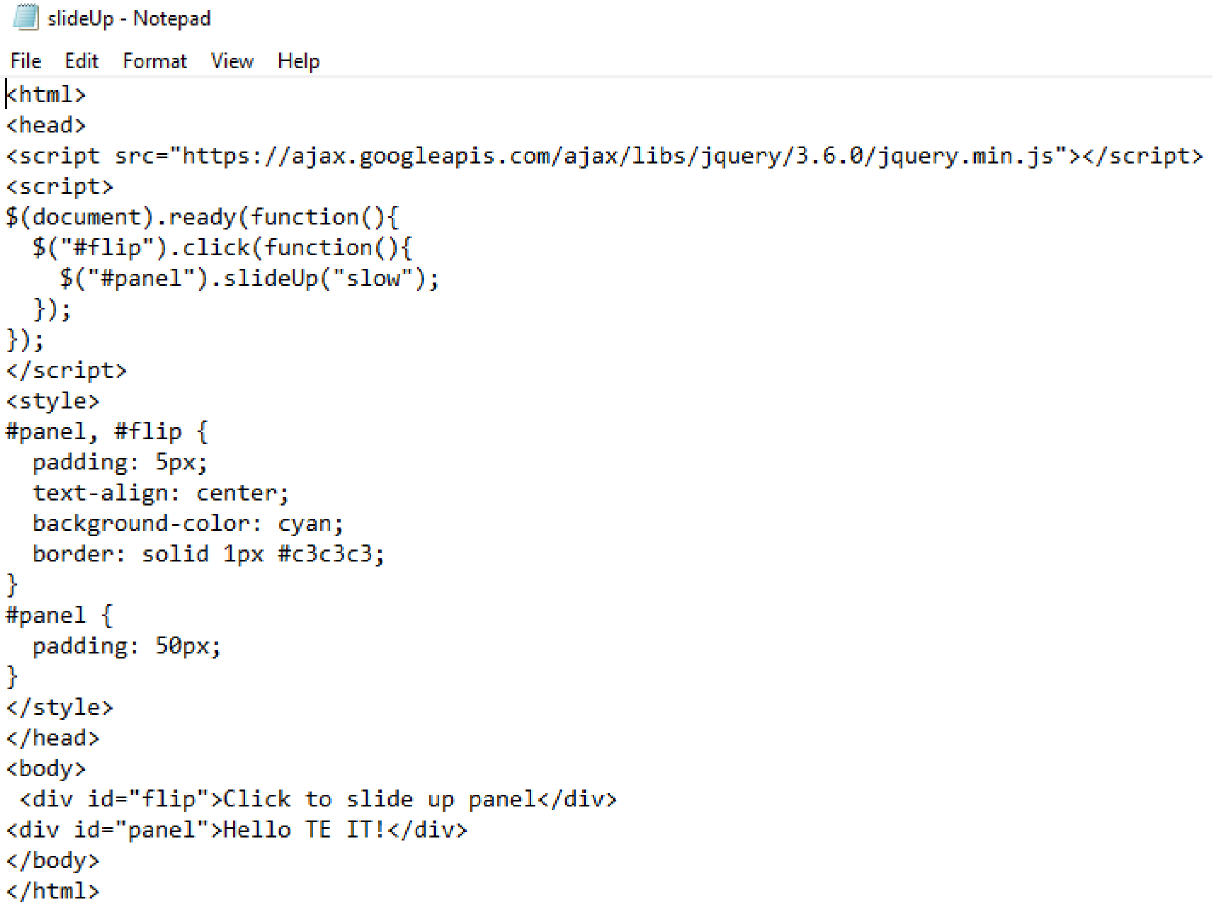
# jQuery slideUp() Method

- The jQuery slideUp() Method is used to slide up an element.
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the sliding completes.

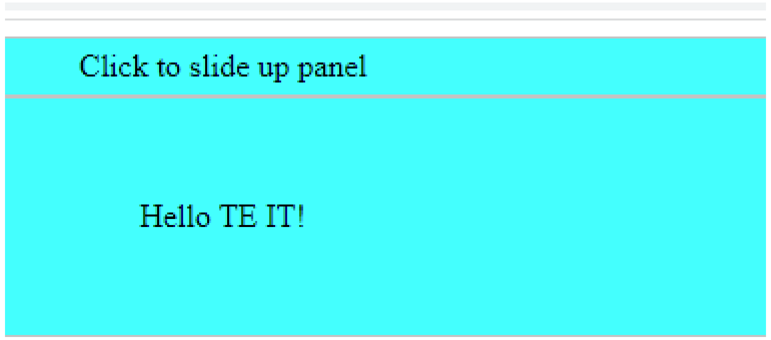
# Syntax

$(*selector*).slideUp(*speed,callback*);

# Demonstration of jQuery slideUp() Method



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideUp("slow");
  });
});
</script>
<style>
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: cyan;
  border: solid 1px #c3c3c3;
}
#panel {
  padding: 50px;
}
</style>
</head>
<body>
 <div id="flip">Click to slide up panel</div>
<div id="panel">Hello TE IT!</div>
</body>
</html>
```



Click to slide up panel

Hello TE IT!

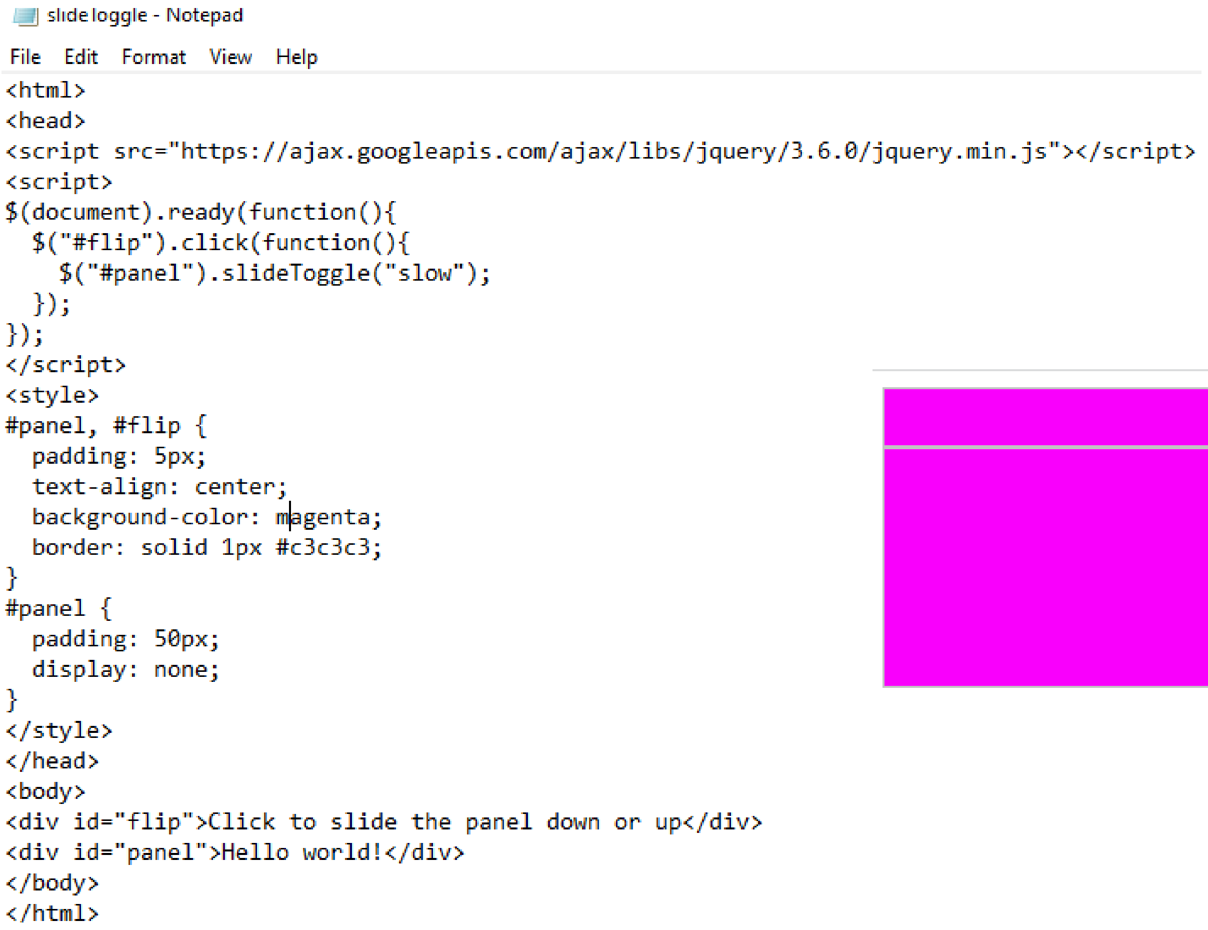
# jQuery slideToggle() Method

- The jQuery slideToggle() Method toggles between slideDown() and slideUp() methods.
- If the elements have been slid down, slideToggle() will slide them up
- If the elements have been slid up, slideToggle() will slide them down
- The optional speed parameter can take the following values: "slow", "fast", milliseconds.
- The optional callback parameter is a function to be executed after the sliding completes.

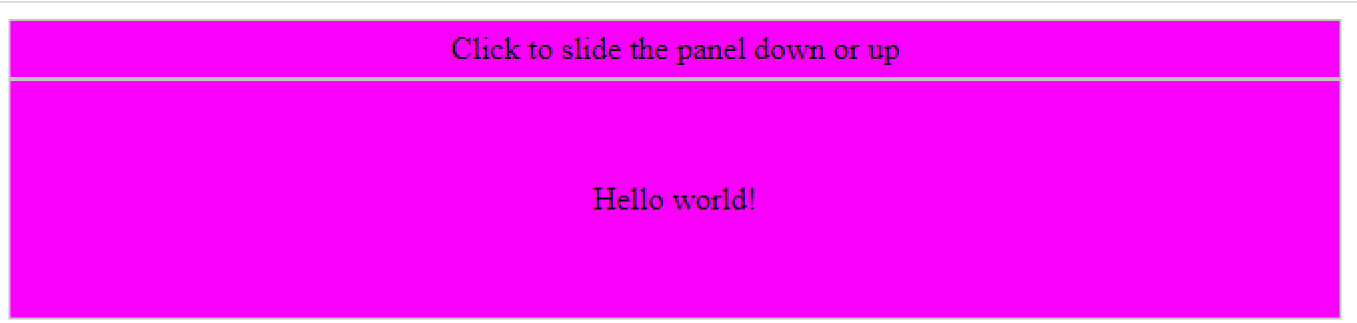
# Syntax

```
$(selector).slideToggle(speed,callback);
```

# Demonstration of jQuery slideToggle() Method



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideToggle("slow");
  });
});
</script>
<style>
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: magenta;
  border: solid 1px #c3c3c3;
}
#panel {
  padding: 50px;
  display: none;
}
</style>
</head>
<body>
<div id="flip">Click to slide the panel down or up</div>
<div id="panel">Hello world!</div>
</body>
</html>
```



Click to slide the panel down or up

Hello world!

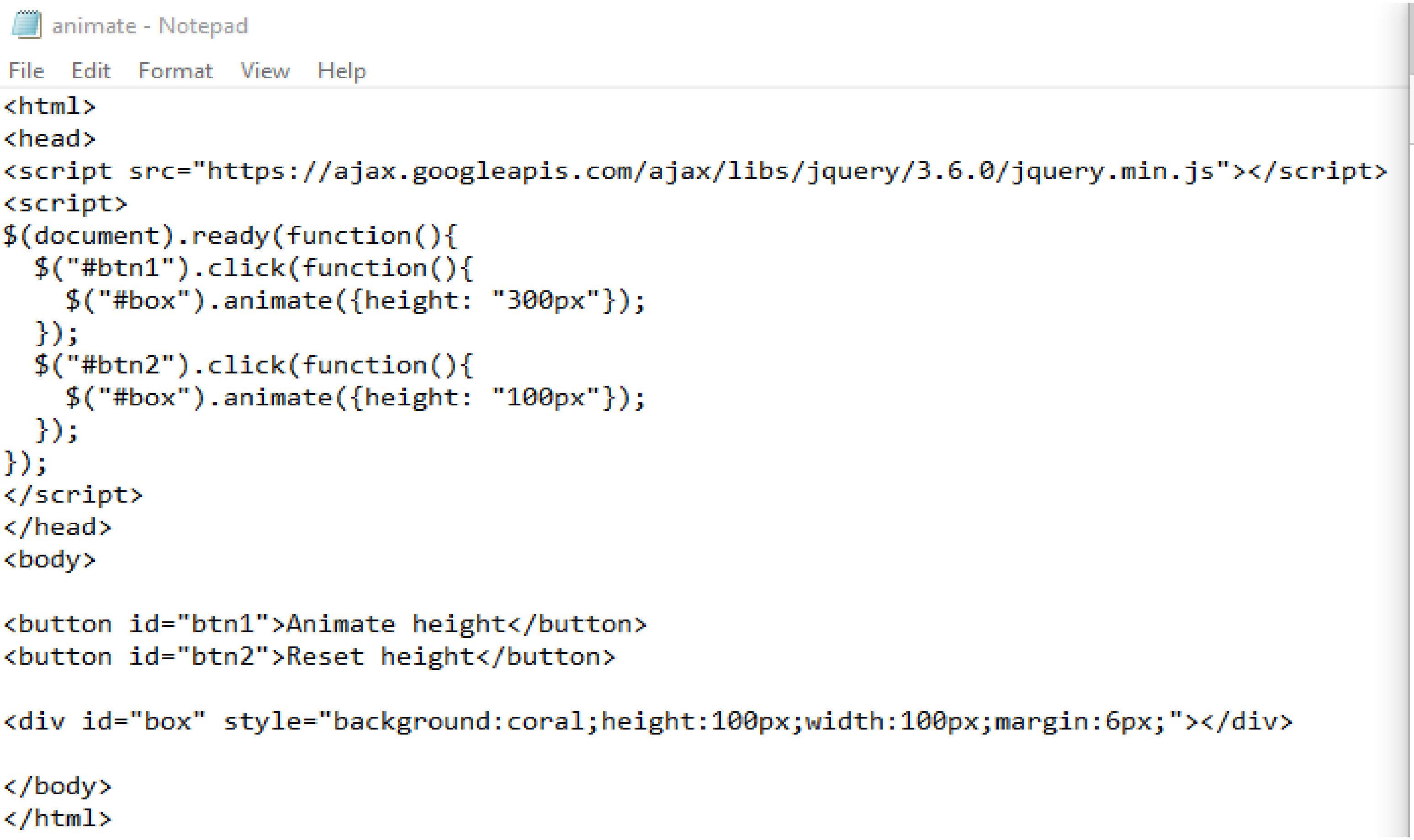
# Bachelor Of Engineering
# In
# Information Technology

# Semester Six, Third Year(Even semester)
# 04th May 2022
# Offline - 16th Lecture

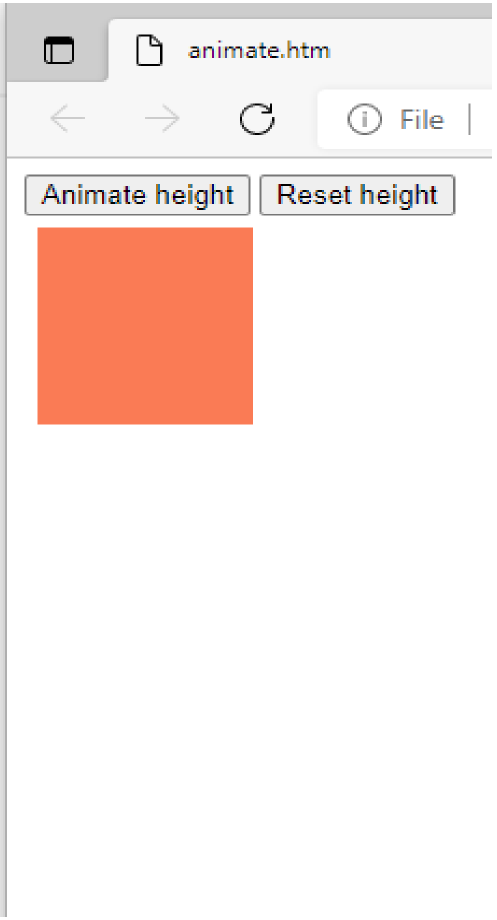
Padre Conceicao College of Engineering

Verna Goa 403722 India

# jQuery animate() Method



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#box").animate({height: "300px"});
  });
  $("#btn2").click(function(){
    $("#box").animate({height: "100px"});
  });
});
</script>
</head>
<body>

<button id="btn1">Animate height</button>
<button id="btn2">Reset height</button>

<div id="box" style="background:coral;height:100px;width:100px;margin:6px;"></div>

</body>
</html>
```

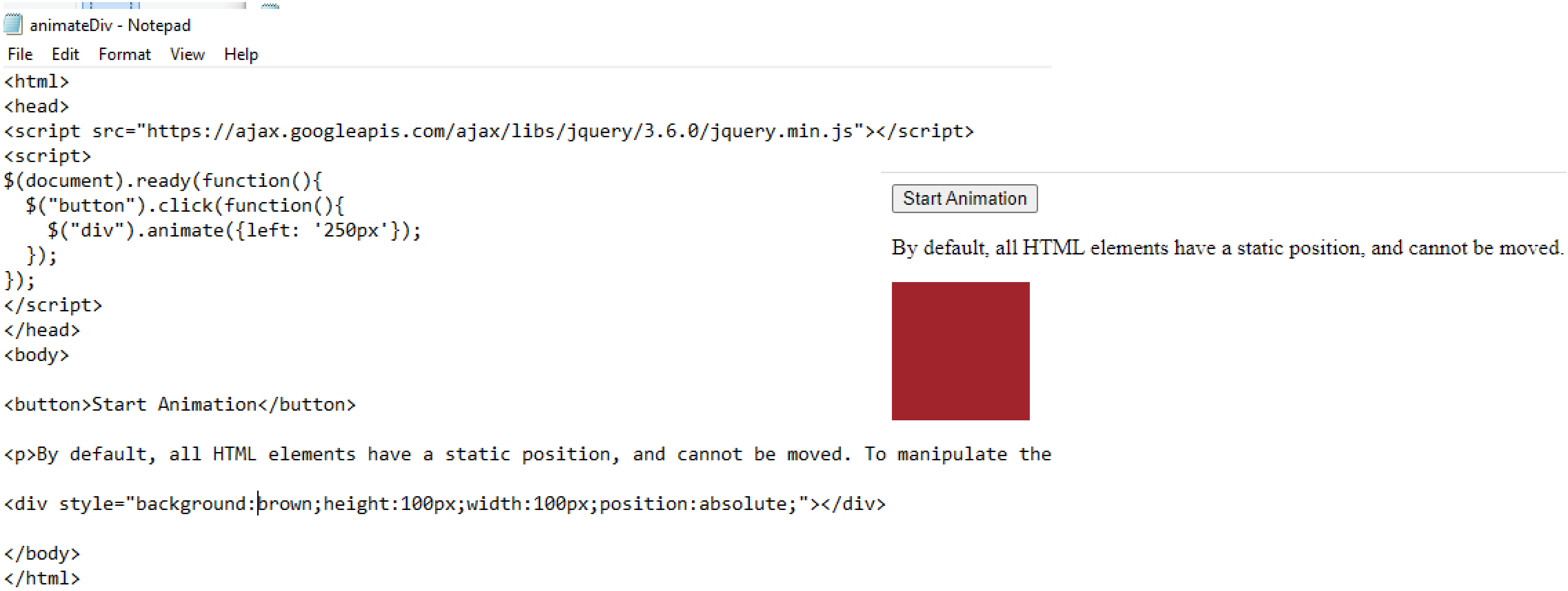


# Definition and Usage of jQuery animate() Method

- The animate() method performs a custom animation of a set of CSS properties.
- This method changes an element from one state to another with CSS styles. The CSS property value is changed gradually, to create an animated effect.
- Only numeric values can be animated (like "margin:30px").
- String values cannot be animated (like "background-color:red"), except for the strings "show", "hide" and "toggle".
- These values allow hiding and showing the animated element

- The required params parameter defines the CSS properties to be animated.
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the animation completes.
- Syntax
- $(*selector*).animate({*params*},*speed,callback*);

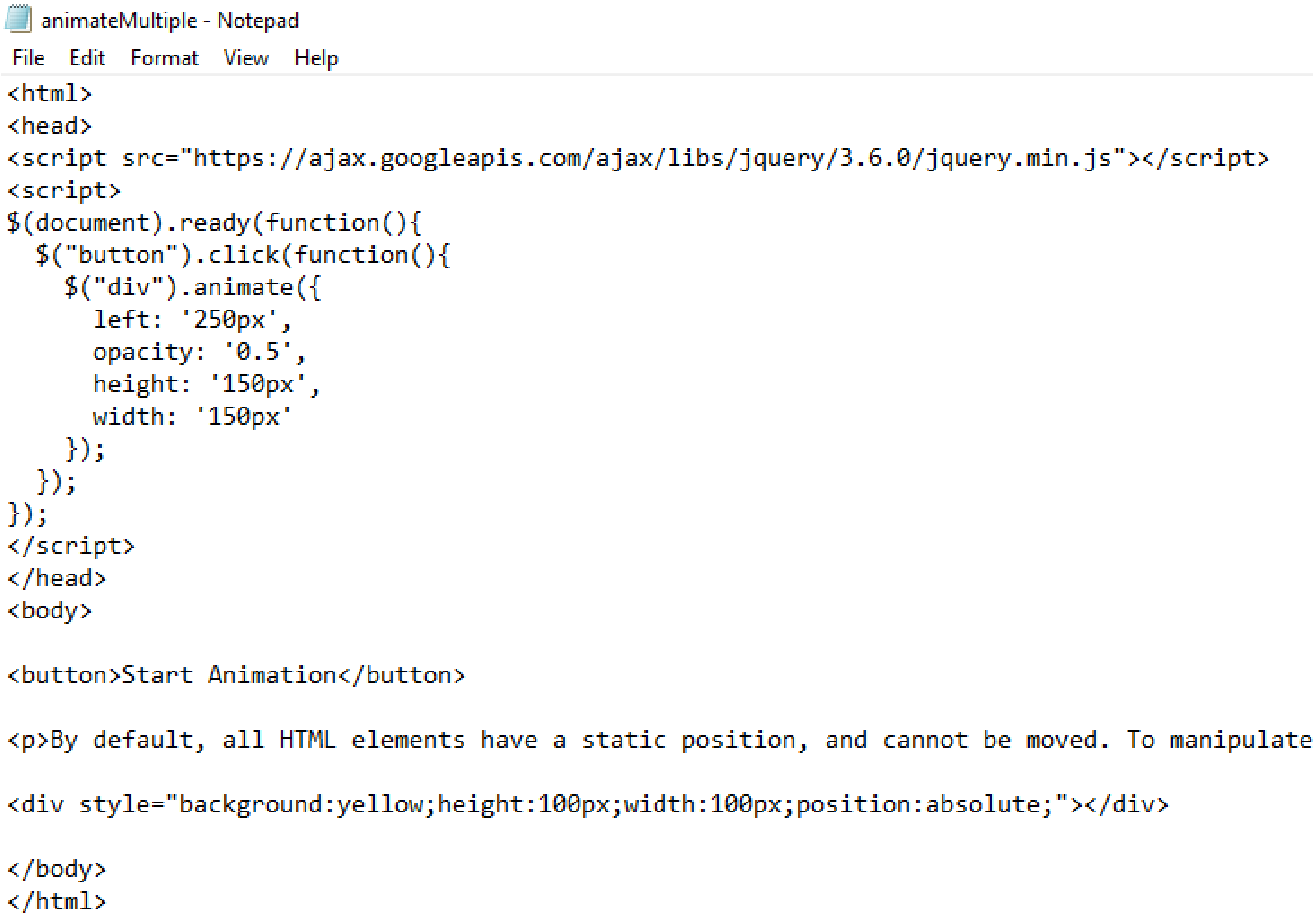
# Demonstration of jQuery animate() Method



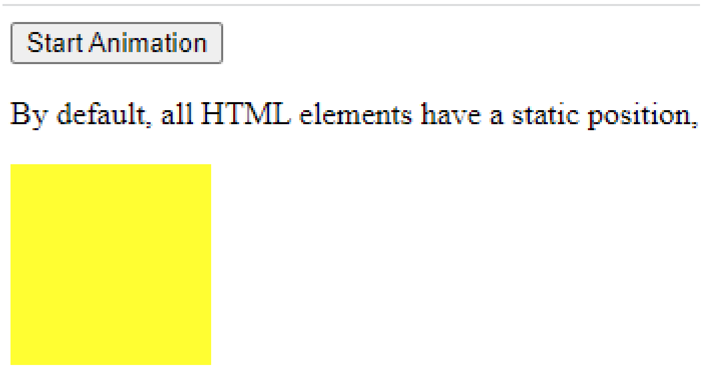
```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({left: '250px'});
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the

<div style="background:brown;height:100px;width:100px;position:absolute;"></div>

</body>
</html>
```

By default, all HTML elements have a static position, and cannot be moved.

# jQuery animate() - Manipulate Multiple Properties



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({
      left: '250px',
      opacity: '0.5',
      height: '150px',
      width: '150px'
    });
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate

<div style="background:yellow;height:100px;width:100px;position:absolute;"></div>

</body>
</html>
```

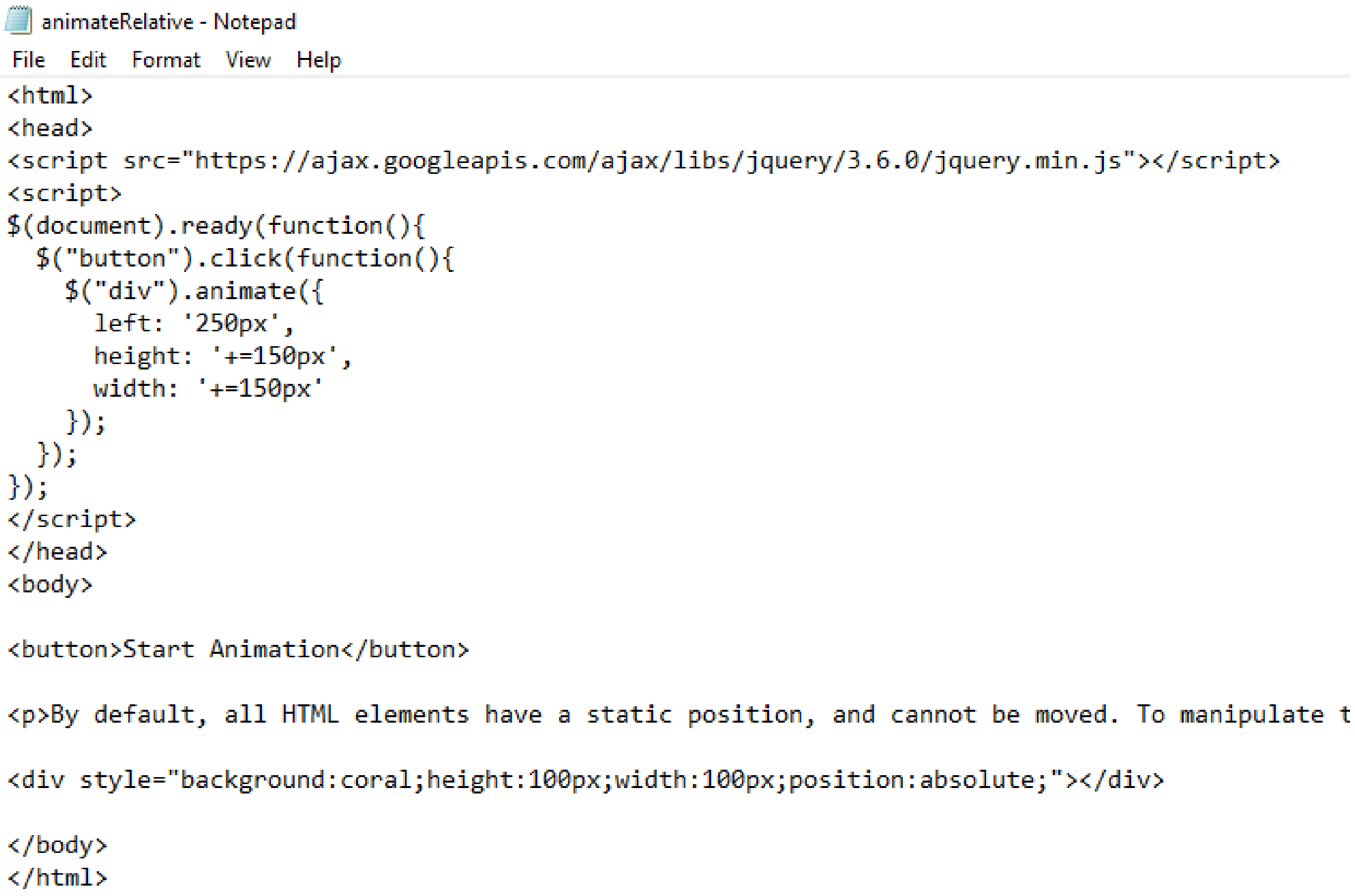


By default, all HTML elements have a static position,

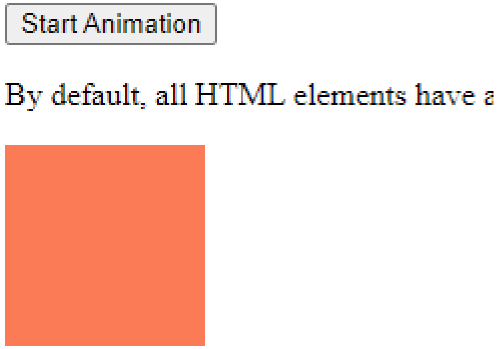
# jQuery animate() - Using Relative Values

- It is also possible to define relative values (the value is then relative to the element's current value). This is done by putting += or -= in front of the value:

# Demonstration of jQuery animate() - Using Relative Values



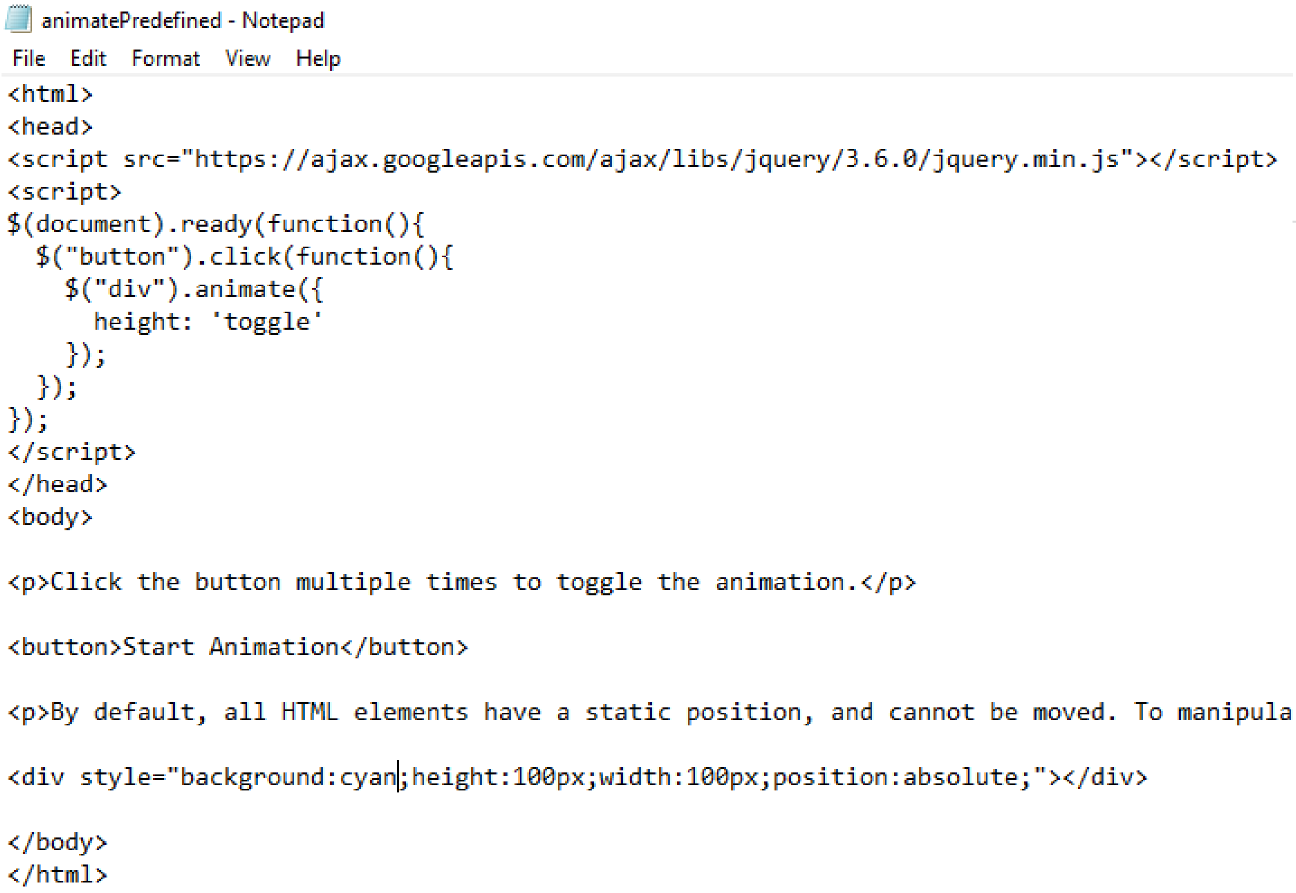
```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({
      left: '250px',
      height: '+=150px',
      width: '+=150px'
    });
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate t

<div style="background:coral;height:100px;width:100px;position:absolute;"></div>

</body>
</html>
```



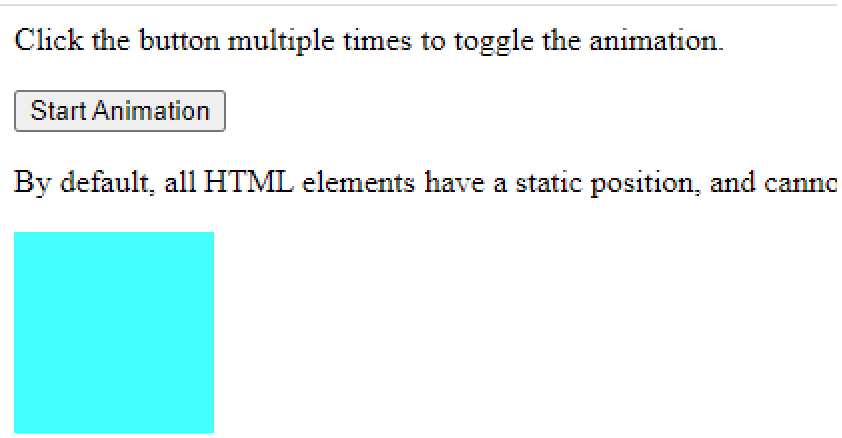
By default, all HTML elements have a

# jQuery animate() - Using Pre-defined Values

- You can specify a property's animation value as slow,hide or toggle



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({
      height: 'toggle'
    });
  });
});
</script>
</head>
<body>

<p>Click the button multiple times to toggle the animation.</p>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipula

<div style="background:cyan;height:100px;width:100px;position:absolute;"></div>

</body>
</html>
```

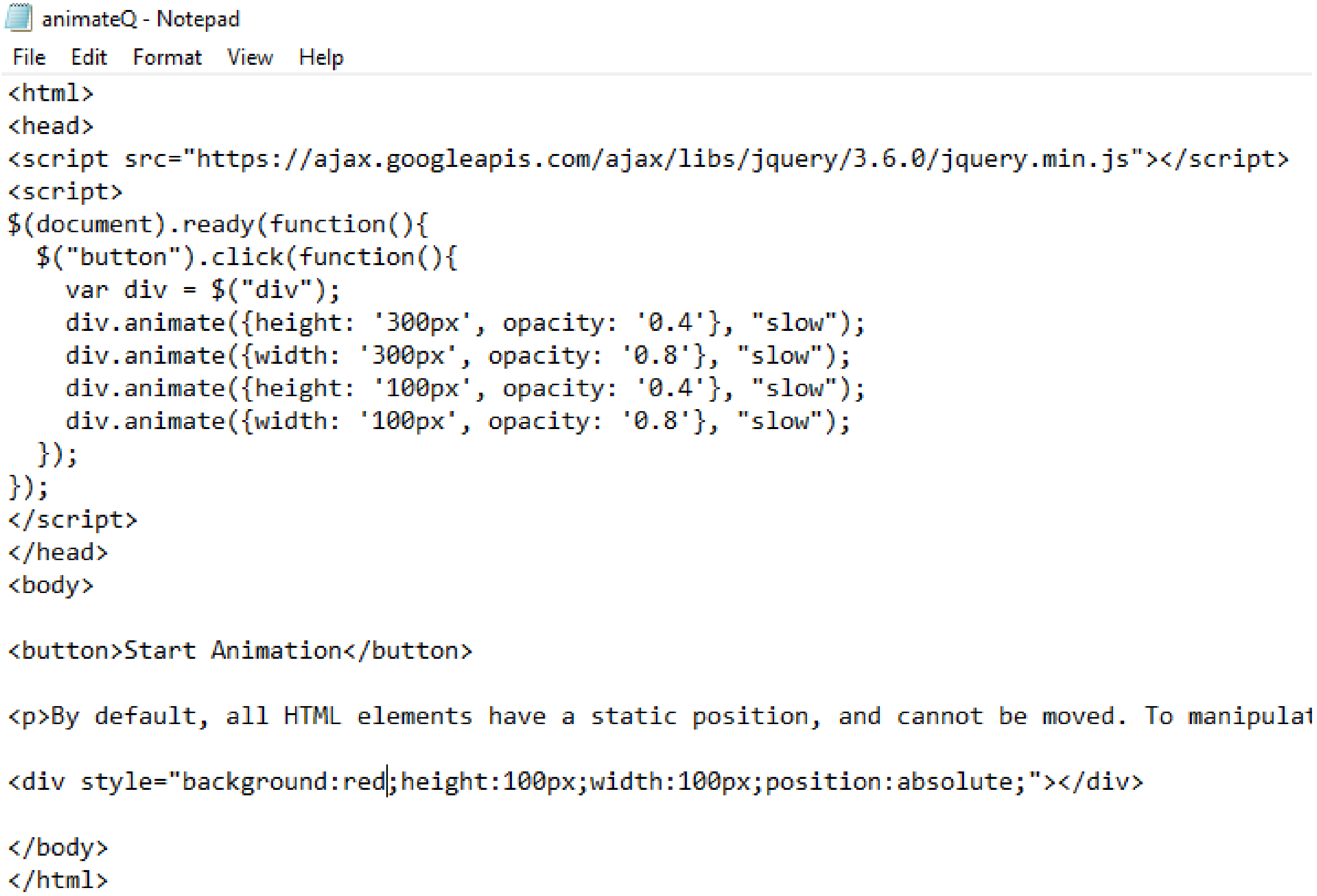

Click the button multiple times to toggle the animation.

By default, all HTML elements have a static position, and cannc

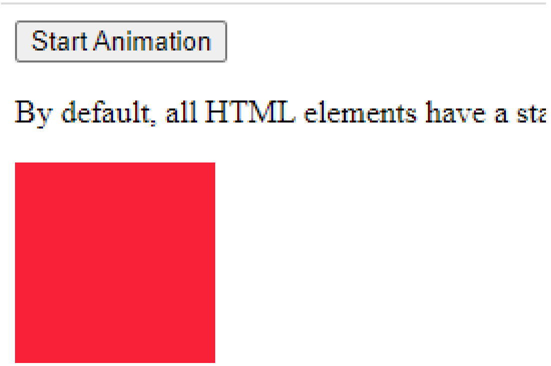
# jQuery animate() - Uses Queue Functionality

- By default, jQuery comes with queue functionality for animations.
- This means that if you write multiple animate() calls after each other, jQuery creates an "internal" queue with these method calls. Then it runs the animate calls ONE by ONE.
- So, if you want to perform different animations after each other, we take advantage of the queue functionality:

# Demonstration of jQuery animate() Queue Functionality

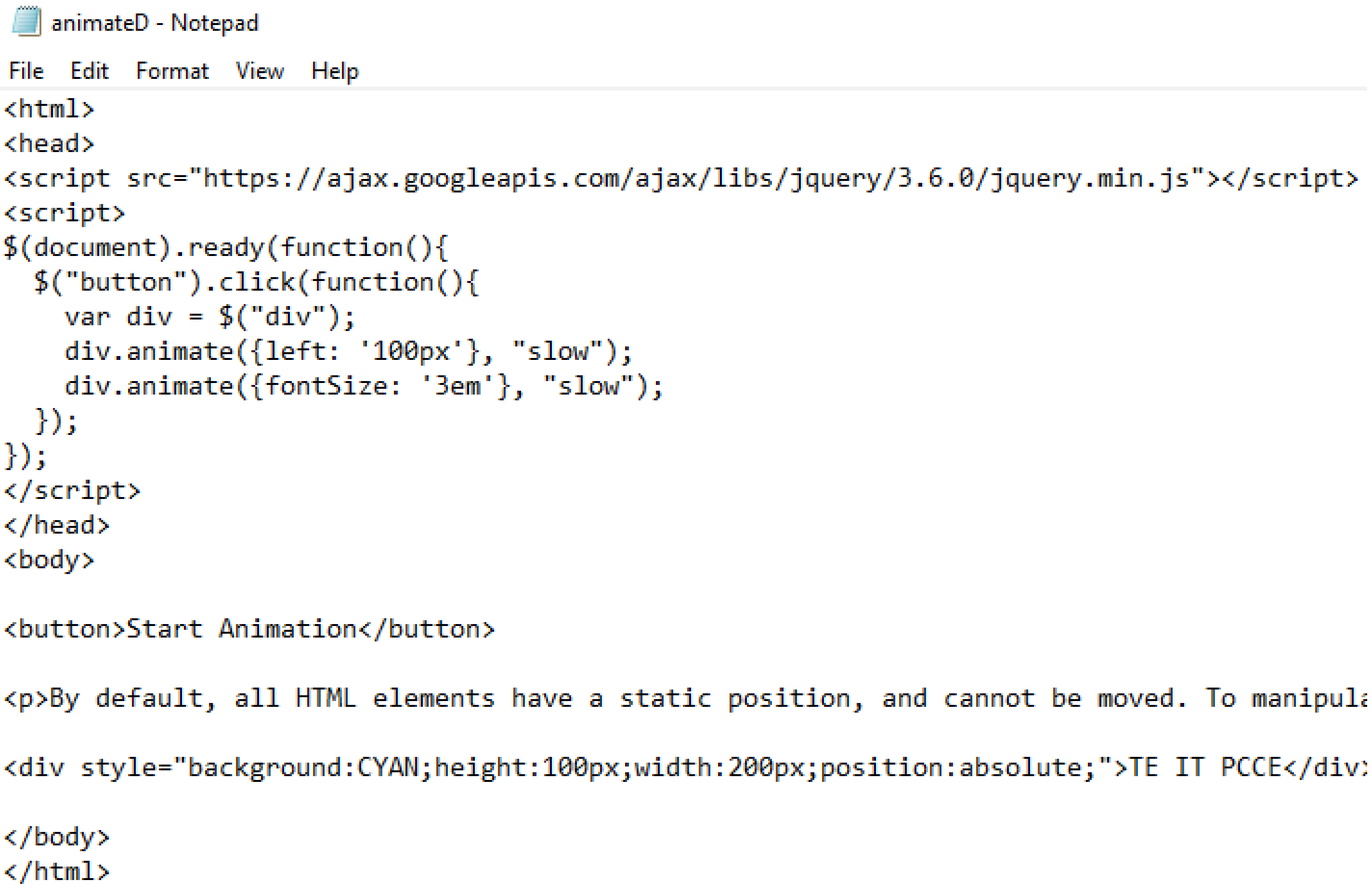


```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var div = $("div");
    div.animate({height: '300px', opacity: '0.4'}, "slow");
    div.animate({width: '300px', opacity: '0.8'}, "slow");
    div.animate({height: '100px', opacity: '0.4'}, "slow");
    div.animate({width: '100px', opacity: '0.8'}, "slow");
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulat

<div style="background:red;height:100px;width:100px;position:absolute;"></div>

</body>
</html>
```

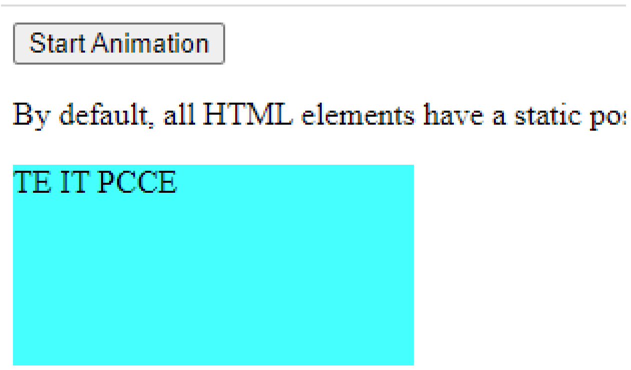


By default, all HTML elements have a sta

- The example below first moves the <div> element to the right, and then increases the font size of the text:



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var div = $("div");
    div.animate({left: '100px'}, "slow");
    div.animate({fontSize: '3em'}, "slow");
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipula

<div style="background:CYAN;height:100px;width:200px;position:absolute;">TE IT PCCE</div

</body>
</html>
```



By default, all HTML elements have a static po

TE IT PCCE

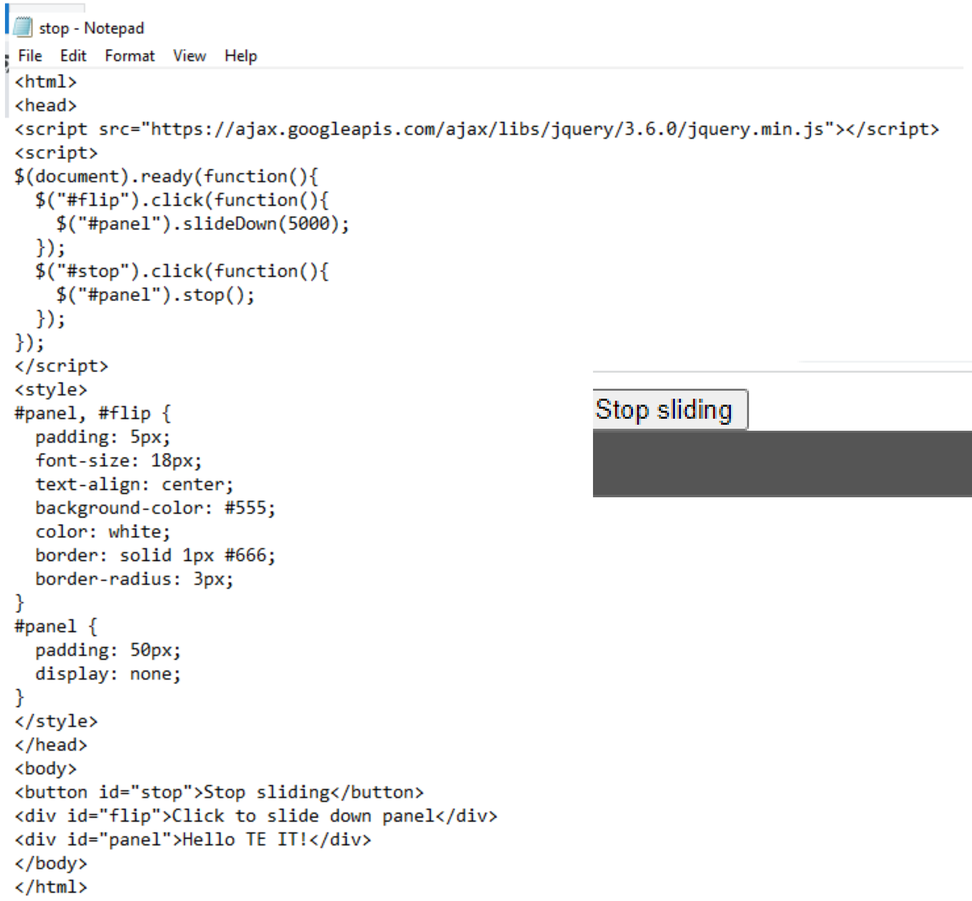
# jQuery Stop Animations

- The jQuery stop() method is used to stop animations or effects before it is finished.
- The jQuery stop() method works for all jQuery effect functions, including sliding, fading and custom animations.
- The optional stopAll parameter specifies whether also the animation queue should be cleared or not. Default is false, which means that only the active animation will be stopped, allowing any queued animations to be performed afterwards.
- The optional goToEnd parameter specifies whether or not to complete the current animation immediately. Default is false.
- By default, the stop() method kills the current animation

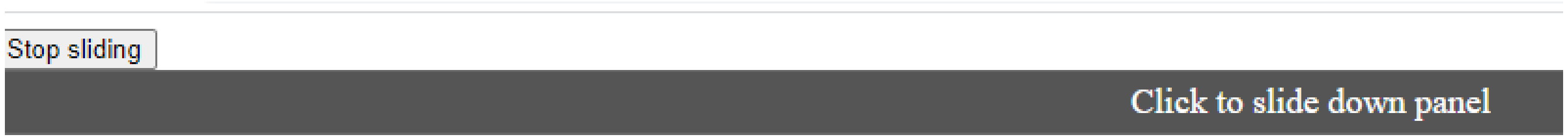
# Syntax

```
$(selector).stop(stopAll,goToEnd);
```

# Demonstration of jQuery Stop()



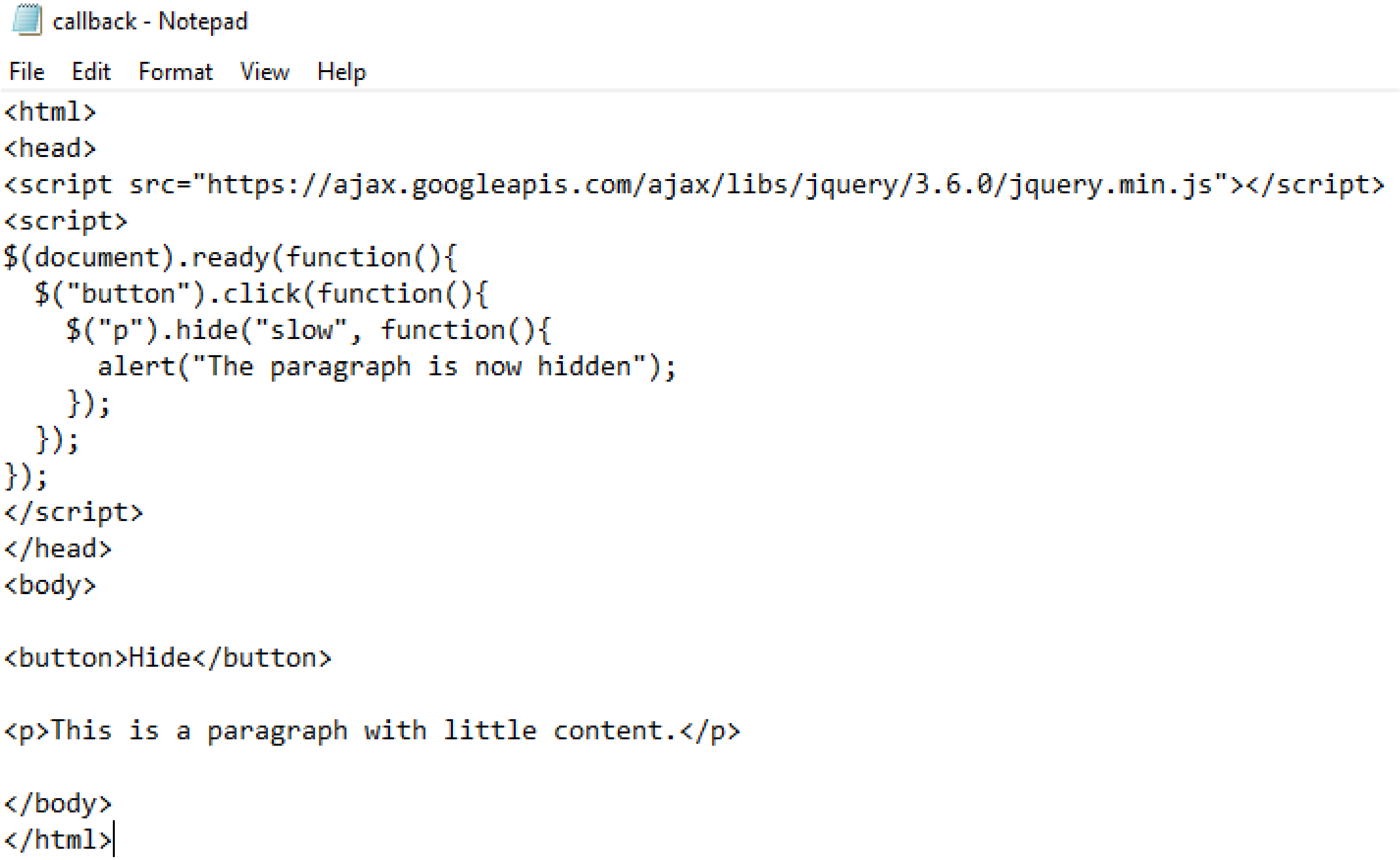
```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideDown(5000);
  });
  $("#stop").click(function(){
    $("#panel").stop();
  });
});
</script>
<style>
#panel, #flip {
  padding: 5px;
  font-size: 18px;
  text-align: center;
  background-color: #555;
  color: white;
  border: solid 1px #666;
  border-radius: 3px;
}
#panel {
  padding: 50px;
  display: none;
}
</style>
</head>
<body>
<button id="stop">Stop sliding</button>
<div id="flip">Click to slide down panel</div>
<div id="panel">Hello TE IT!</div>
</body>
</html>
```



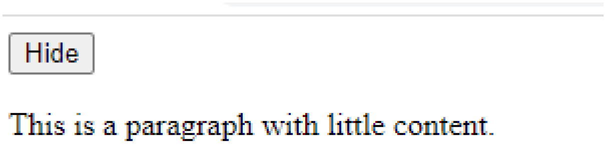
# jQuery Callback Functions

- A callback function is executed after the current effect is 100% finished.
- JavaScript statements are executed line by line. However, with effects, the next line of code can be run even though the effect is not finished. This can create errors.
- To prevent this, you can create a callback function.
- A callback function is executed after the current effect is finished.
- Syntax $(*selector*).hide(*speed,callback*);

- The example below has a callback parameter that is a function that will be executed after the hide effect is completed
- Example with Callback

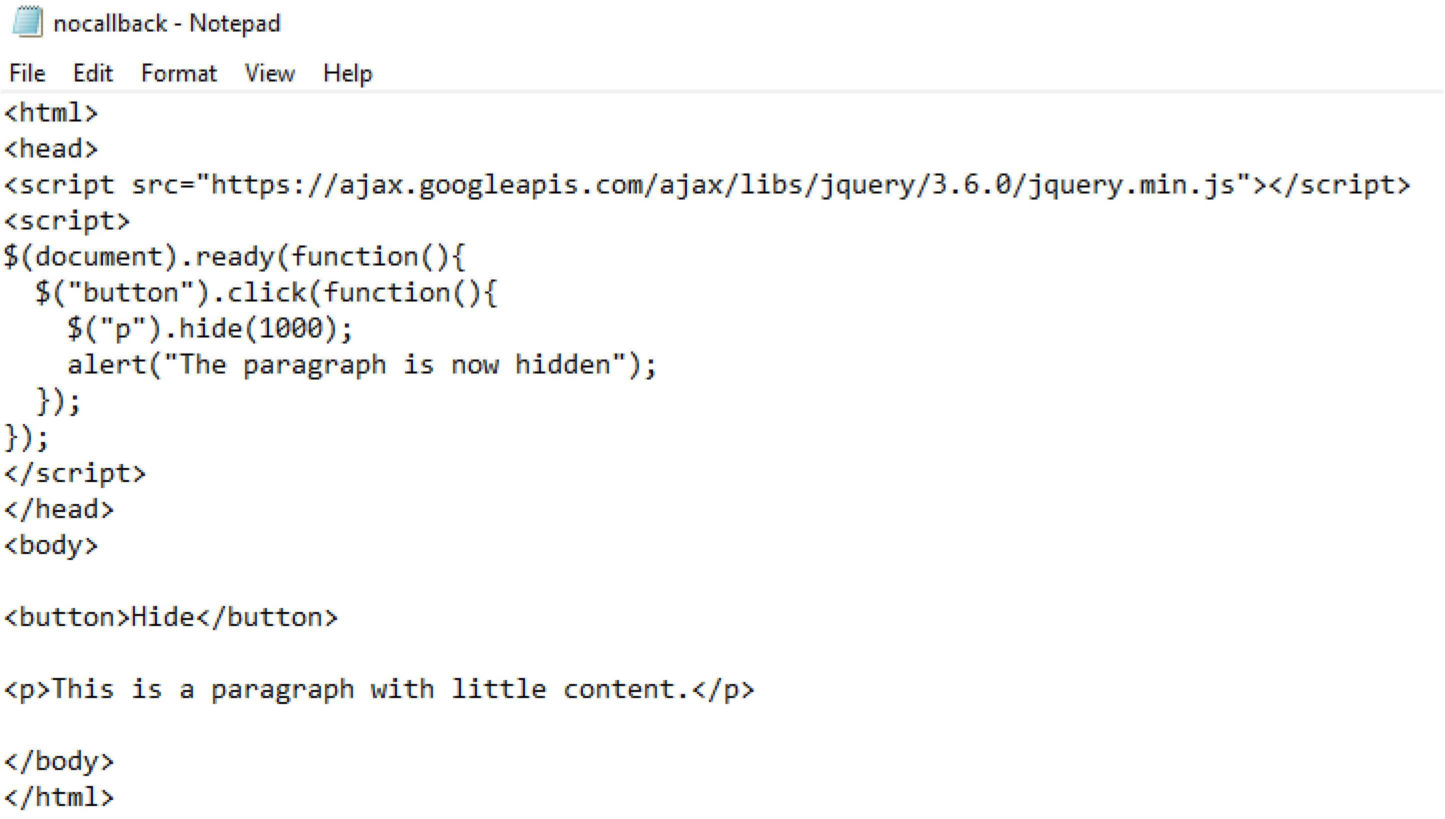


```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide("slow", function(){
      alert("The paragraph is now hidden");
    });
  });
});
</script>
</head>
<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>

</body>
</html>
```

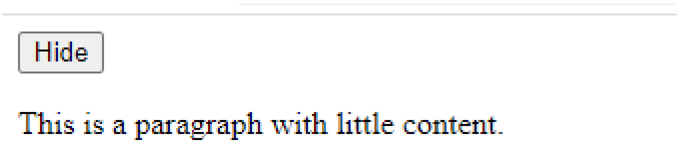


This is a paragraph with little content.

- The example below has no callback parameter, and the alert box will be displayed before the hide effect is completed.
- Example without Callback



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide(1000);
    alert("The paragraph is now hidden");
  });
});
</script>
</head>
<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>

</body>
</html>
```

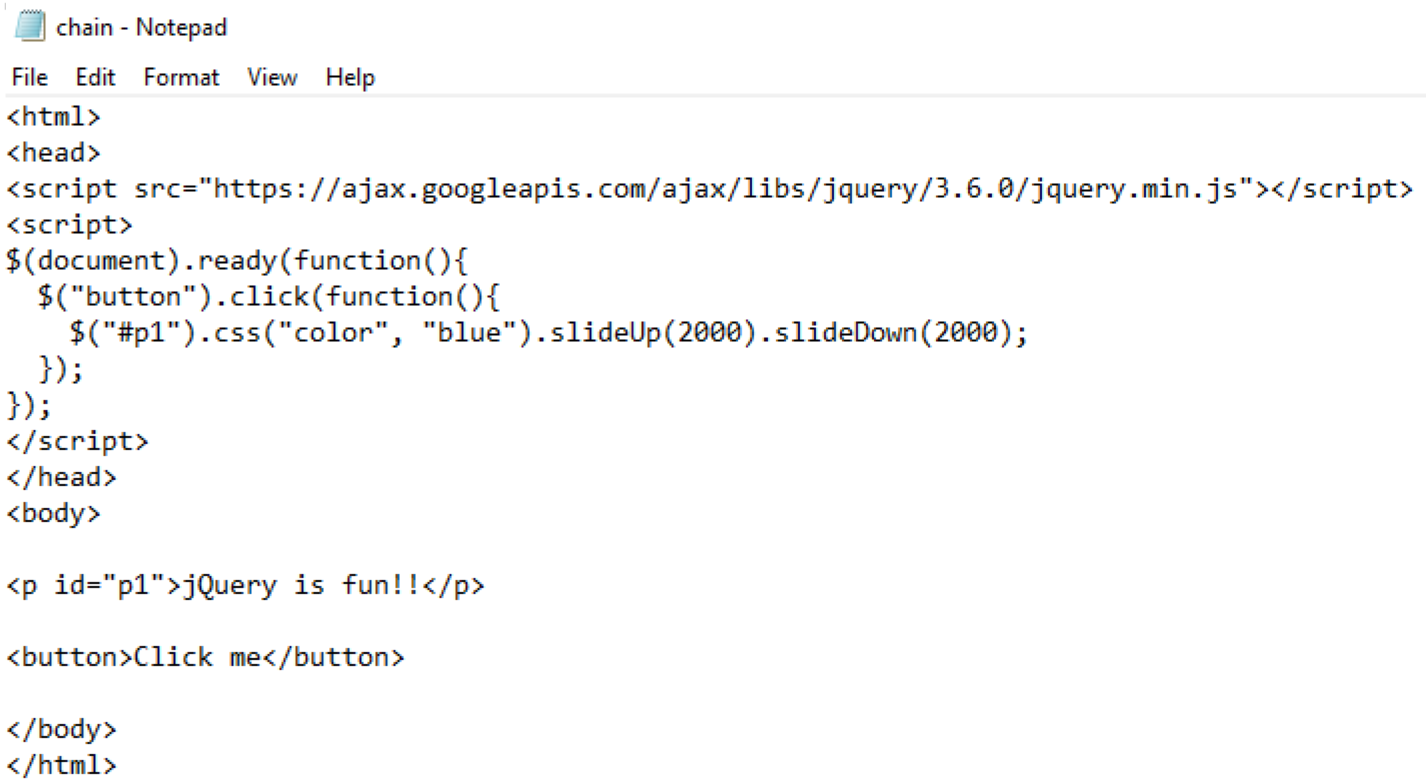


This is a paragraph with little content.

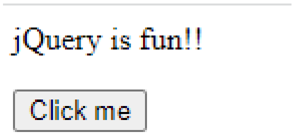
# jQuery - Chaining

- With jQuery, you can chain together actions/methods.
- There is a technique called chaining, that allows us to run multiple jQuery commands, one after the other, on the same element(s) within a single statement.
- **Tip:** This way, browsers do not have to find the same element(s) more than once.
- To chain an action, you simply append the action to the previous action.

The following example chains together the css(), slideUp(), and slideDown() methods. The "p1" element first changes to red, then it slides up, and then it slides down



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#p1").css("color", "blue").slideUp(2000).slideDown(2000);
  });
});
</script>
</head>
<body>

<p id="p1">jQuery is fun!!</p>

<button>Click me</button>

</body>
</html>
```



jQuery is fun!!

# jQuery HTML

- jQuery Get
- jQuery Set
- jQuery Add
- jQuery Remove

# jQuery - Get Content and Attributes

- jQuery contains powerful methods for changing and manipulating HTML elements and attributes.

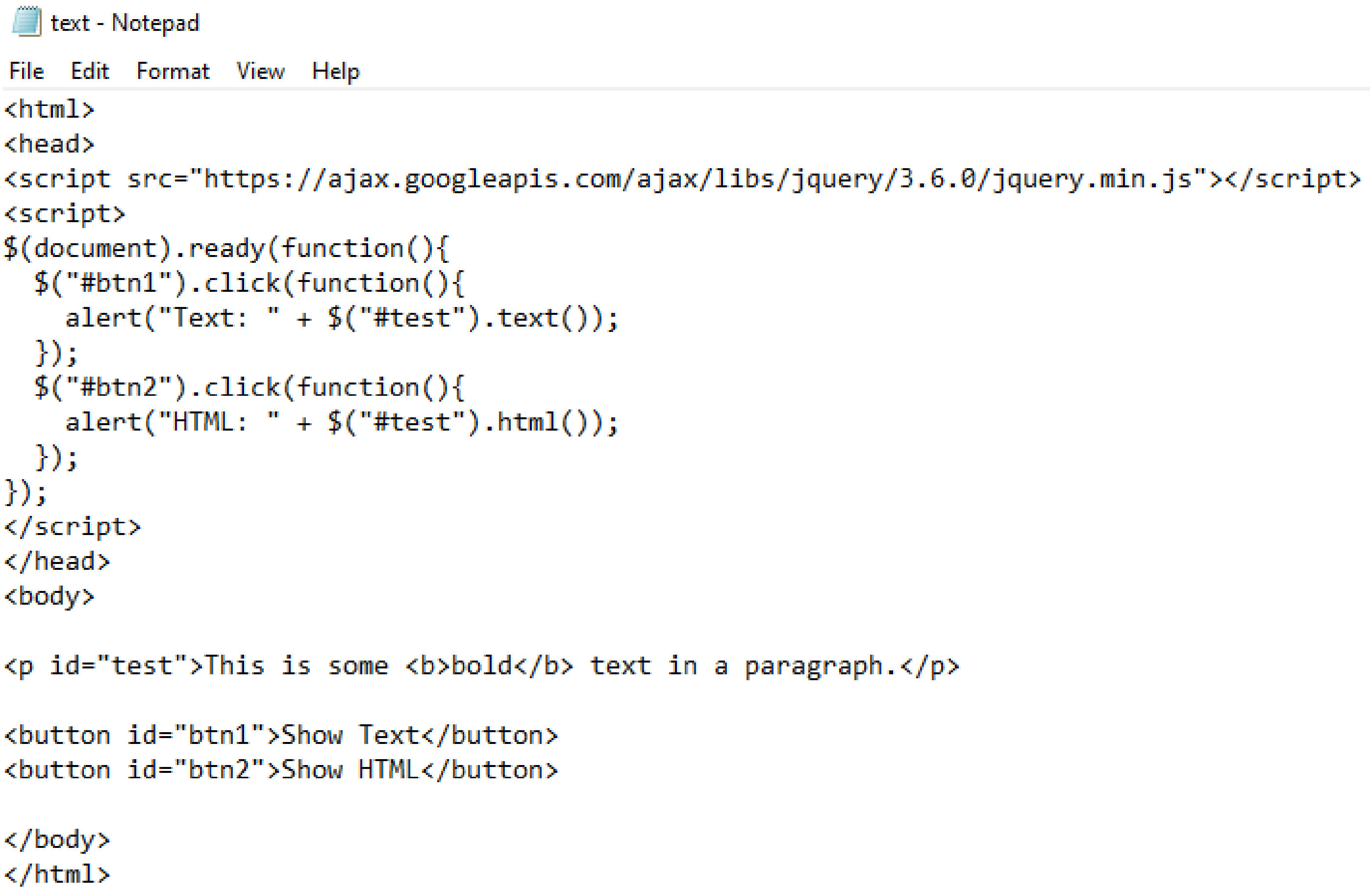
# Get Content - text(), html(), and

Three simple, but useful, jQuery methods for DOM manipulation are:

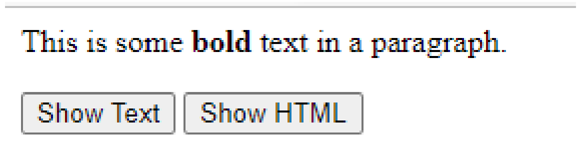
1. text() - Sets or returns the text content of selected elements
2. html() - Sets or returns the content of selected elements (including HTML markup)
3. val() - Sets or returns the value of form fields

- The following example demonstrates how to get content with the jQuery text() and html() methods:

# Demonstration of Get Content - text(), html()



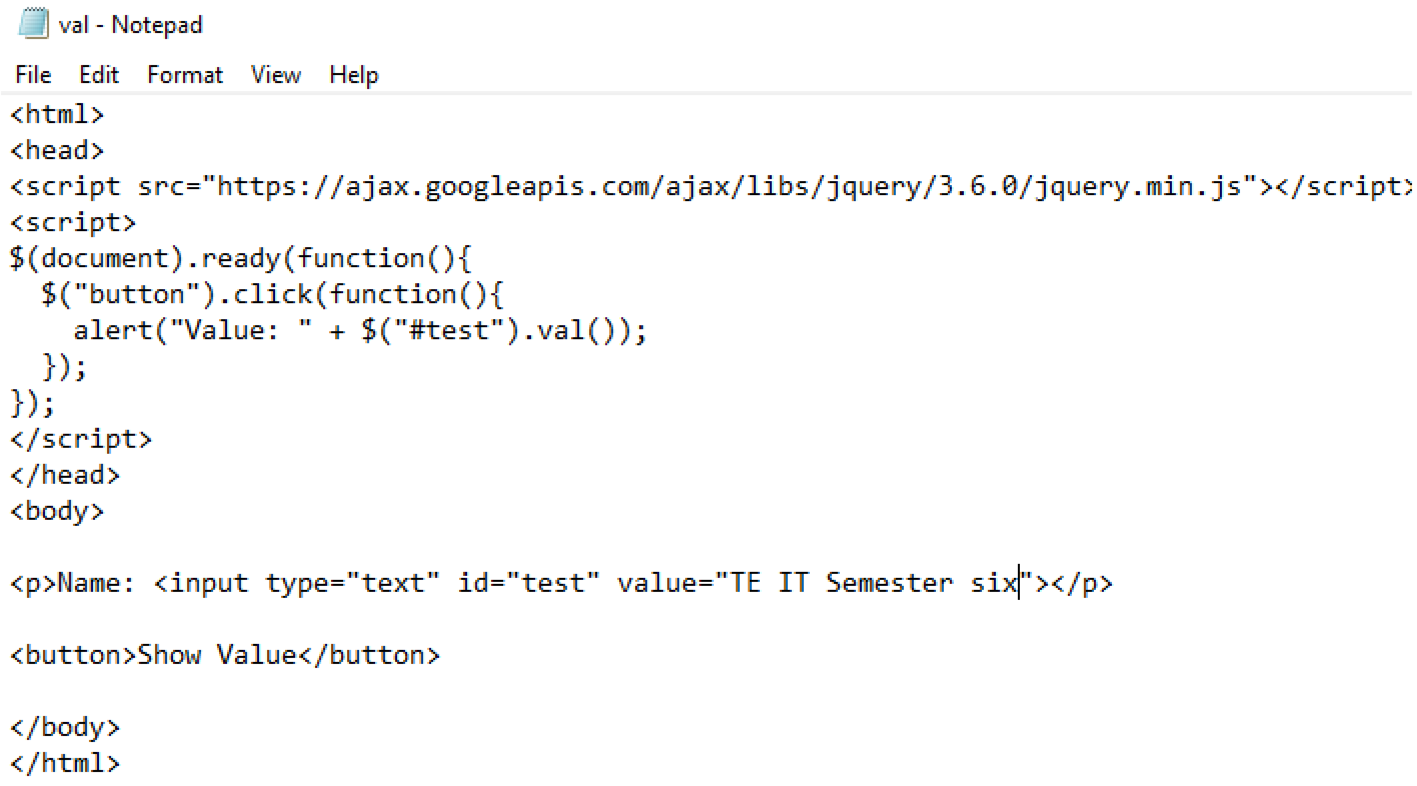
```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    alert("Text: " + $("#test").text());
  });
  $("#btn2").click(function(){
    alert("HTML: " + $("#test").html());
  });
});
</script>
</head>
<body>

<p id="test">This is some <b>bold</b> text in a paragraph.</p>

<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>

</body>
</html>
```



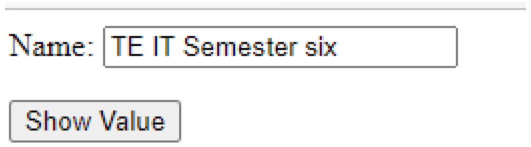
This is some **bold** text in a paragraph.

# Get Content - val() method

- The following example demonstrates how to get the value of an input field with the jQuery val() method.

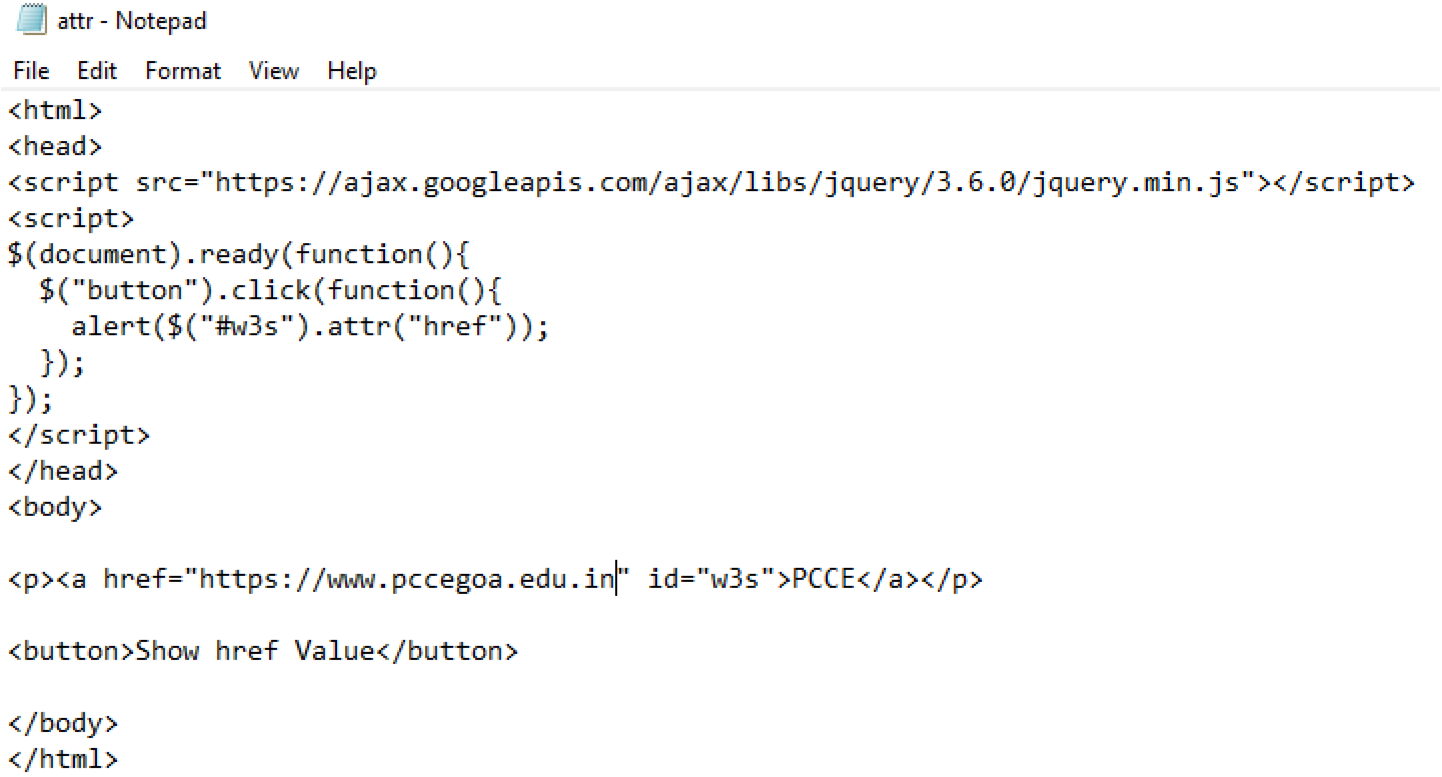


```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    alert("Value: " + $("#test").val());
  });
});
</script>
</head>
<body>

<p>Name: <input type="text" id="test" value="TE IT Semester six"></p>

<button>Show Value</button>

</body>
</html>
```

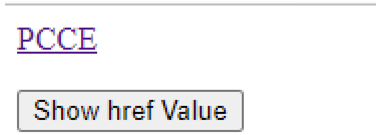

Name: TE IT Semester six

# Get Attributes - attr()

- The jQuery attr() method  is used to get attribute values.
- The following example demonstrates how to get the value of the href attribute in a link:



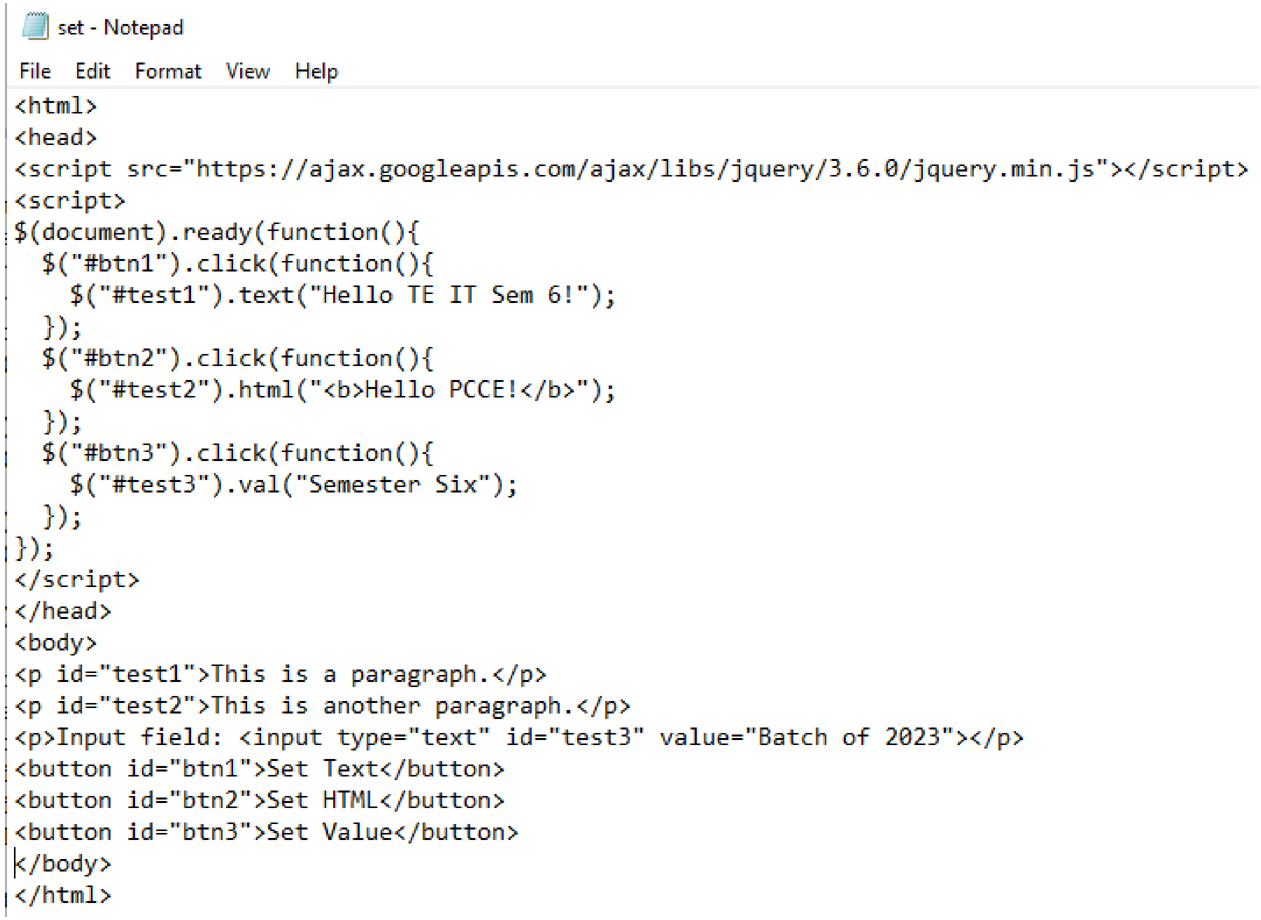
```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    alert($("#w3s").attr("href"));
  });
});
</script>
</head>
<body>

<p><a href="https://www.pccegoa.edu.in" id="w3s">PCCE</a></p>

<button>Show href Value</button>

</body>
</html>
```



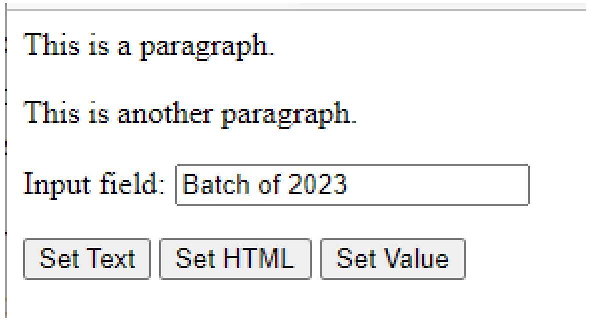
# jQuery - Set Content and Attributes

- Set Content - text(), html(), and val()
- text() - Sets or returns the text content of selected elements
- html() - Sets or returns the content of selected elements (including HTML markup)
- val() - Sets or returns the value of form fields
- The following example demonstrates how to set content with the jQuery text(), html(), and val() methods

# Demonstration of Set Content - text(), html(),val()



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#test1").text("Hello TE IT Sem 6!");
  });
  $("#btn2").click(function(){
    $("#test2").html("<b>Hello PCCE!</b>");
  });
  $("#btn3").click(function(){
    $("#test3").val("Semester Six");
  });
});
</script>
</head>
<body>
<p id="test1">This is a paragraph.</p>
<p id="test2">This is another paragraph.</p>
<p>Input field: <input type="text" id="test3" value="Batch of 2023"></p>
<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>
</body>
</html>
```



This is a paragraph.

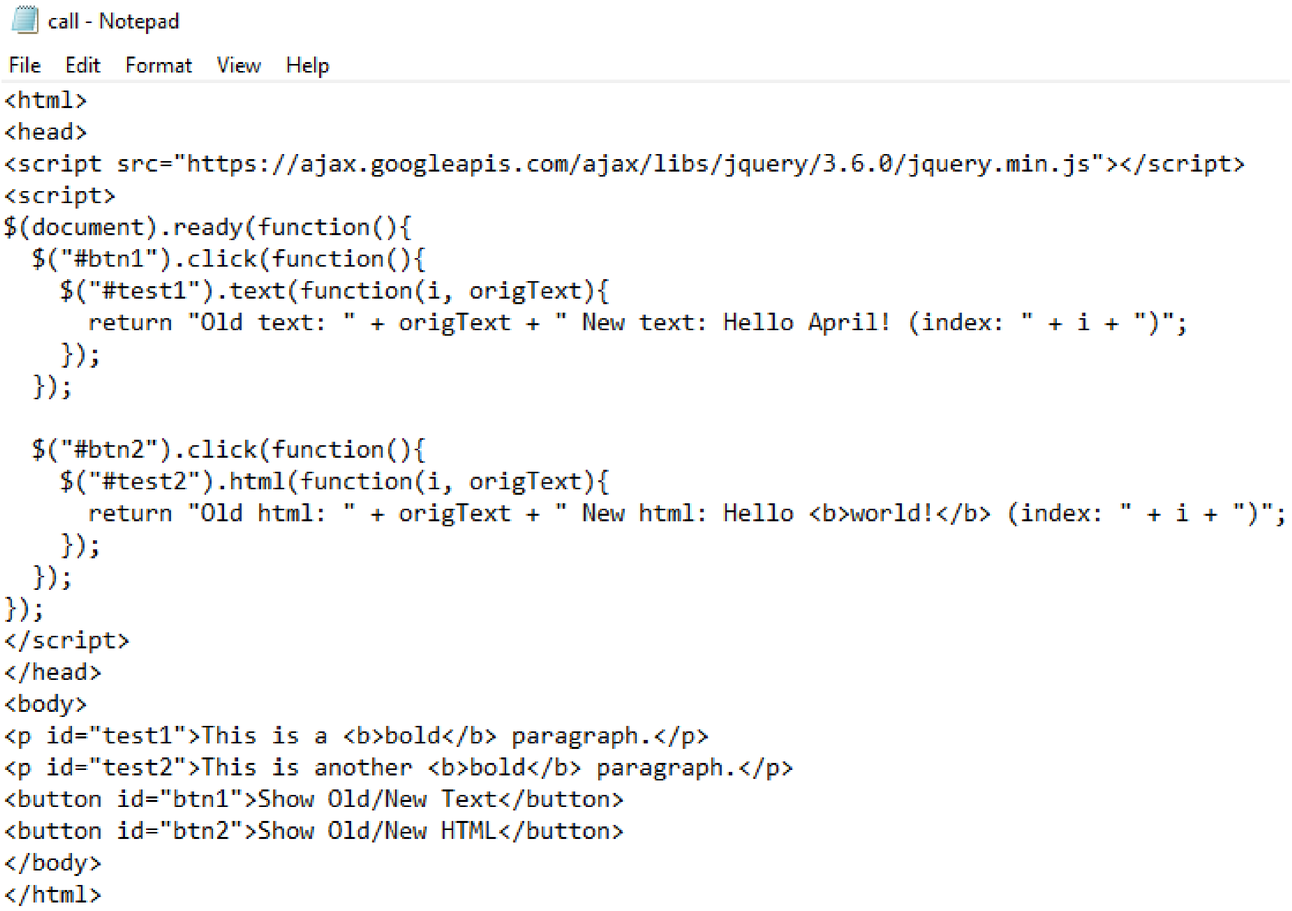
This is another paragraph.

Input field: Batch of 2023

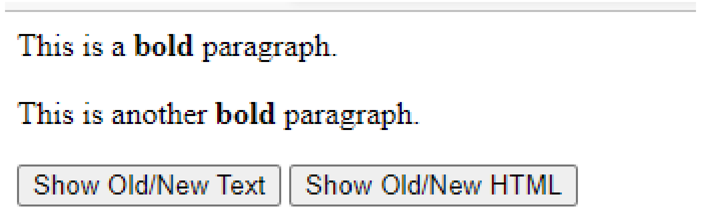
# A Callback Function for text(), html(), and val()

- All of the three jQuery methods text(), html(), and val() also come with a callback function.
- The callback function has two parameters: the index of the current element in the list of elements selected and the original (old) value. You then return the string you wish to use as the new value from the function.
- The following example demonstrates text() and html() with a callback function:

# Demonstration of text(), html() with callback function



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#test1").text(function(i, origText){
      return "Old text: " + origText + " New text: Hello April! (index: " + i + ")";
    });
  });

  $("#btn2").click(function(){
    $("#test2").html(function(i, origText){
      return "Old html: " + origText + " New html: Hello <b>world!</b> (index: " + i + ")";
    });
  });
});
</script>
</head>
<body>
<p id="test1">This is a <b>bold</b> paragraph.</p>
<p id="test2">This is another <b>bold</b> paragraph.</p>
<button id="btn1">Show Old/New Text</button>
<button id="btn2">Show Old/New HTML</button>
</body>
</html>
```



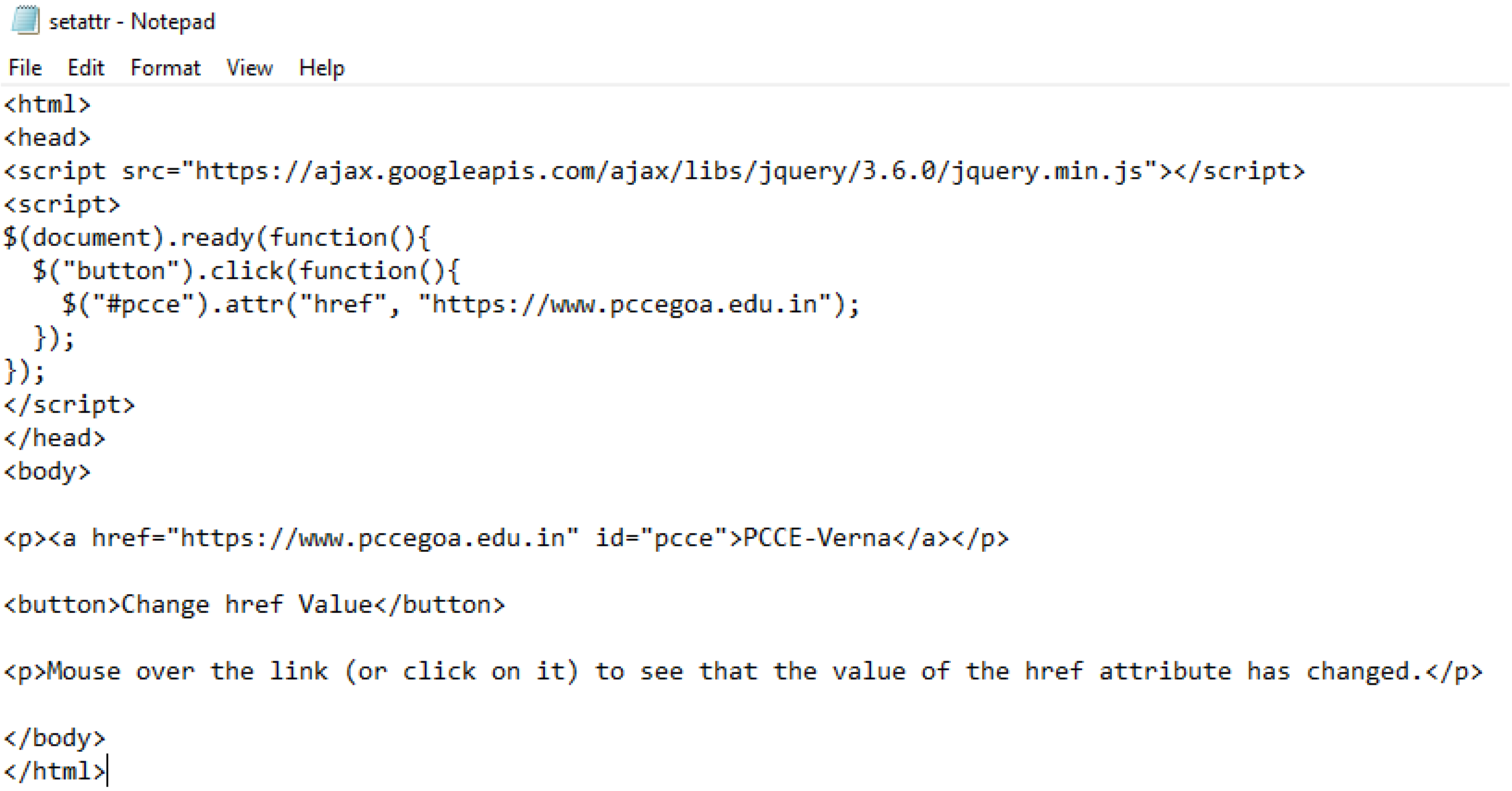
This is a **bold** paragraph.

This is another **bold** paragraph.

# Set Attributes - attr()

- The jQuery attr() method is also used to set/change attribute values.
- The following example demonstrates how to change (set) the value of the href attribute in a link:

# Set Attributes - attr()



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#pcce").attr("href", "https://www.pccegoa.edu.in");
  });
});
</script>
</head>
<body>

<p><a href="https://www.pccegoa.edu.in" id="pcce">PCCE-Verna</a></p>

<button>Change href Value</button>

<p>Mouse over the link (or click on it) to see that the value of the href attribute has changed.</p>

</body>
</html>
```

# Set Attributes - attr() to set multiple attributes

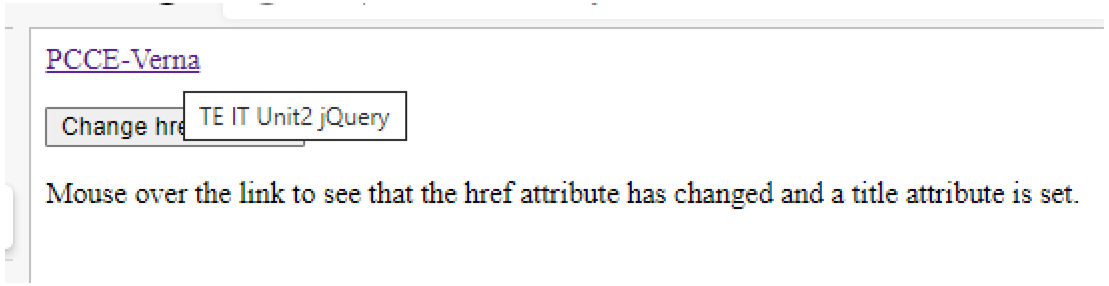
- The jQuery attr() method also allows you to set multiple attributes at the same time.
- The following example demonstrates how to set both the href and title attributes at the same time:

# Set Attributes - attr() to set multiple attributes

setManyAttr - Notepad

File Edit Format View Help

```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#pcce").attr({
      "href" : "https://www.pccegoa.edu.in",
      "title" : "TE IT Unit2 jQuery"
    });
  });
});
</script>
</head>
<body>

<p><a href="https://www.pccegoa.edu.in" title="some title" id="pcce">PCCE-Verna-Goa</a></p>

<button>Change href and title</button>

<p>Mouse over the link to see that the href attribute has changed and a title attribute is set.</p>

</body>
</html>
```

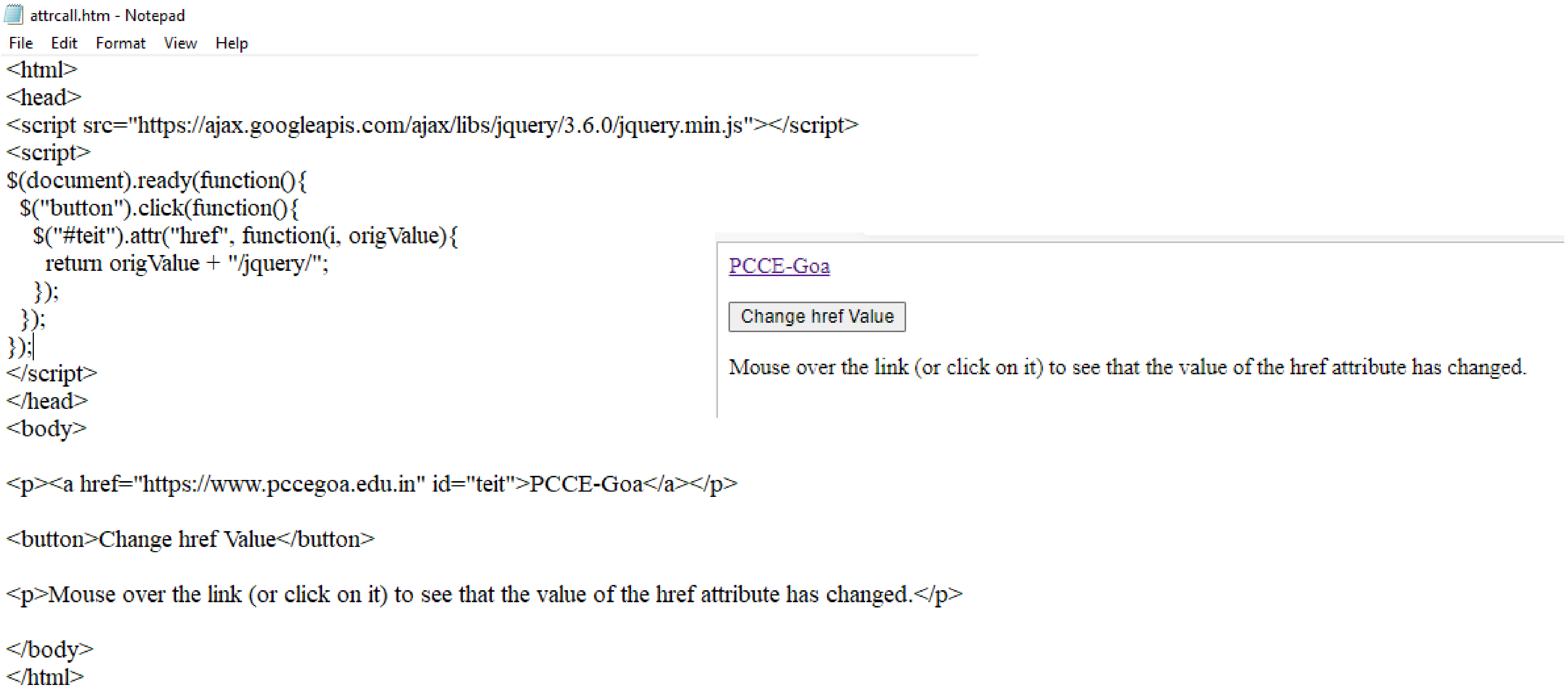


Mouse over the link to see that the href attribute has changed and a title attribute is set.

# A Callback Function for attr()

- The jQuery method attr() also comes with a callback function. The callback function has two parameters: the index of the current element in the list of elements selected and the original (old) attribute value. You then return the string you wish to use as the new attribute value from the function.
- The following example demonstrates attr() with a callback function:

# Demonstration of attr() with a callback function



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#teit").attr("href", function(i, origValue){
      return origValue + "/jquery/";
    });
  });
});
</script>
</head>
<body>

<p><a href="https://www.pccegoa.edu.in" id="teit">PCCE-Goa</a></p>

<button>Change href Value</button>

<p>Mouse over the link (or click on it) to see that the value of the href attribute has changed.</p>

</body>
</html>
```

PCCE-Goa

Mouse over the link (or click on it) to see that the value of the href attribute has changed.

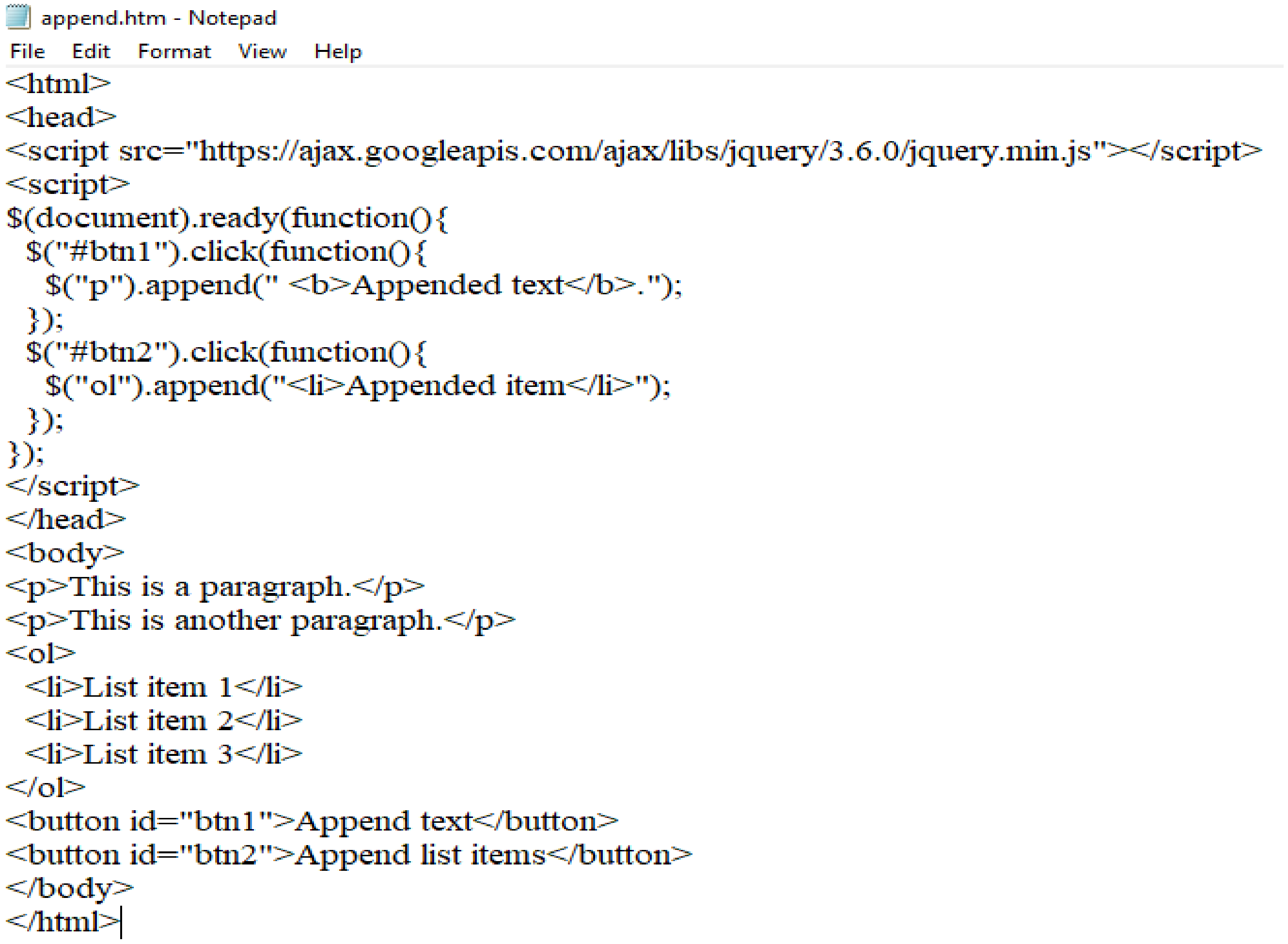
# jQuery - Add Elements

Add New HTML Content

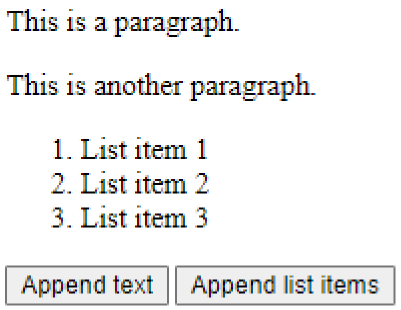
- We will look at four jQuery methods that are used to add new content:
- append() - Inserts content at the end of the selected elements
- prepend() - Inserts content at the beginning of the selected elements
- after() - Inserts content after the selected elements
- before() - Inserts content before the selected elements

# jQuery append() Method

- The jQuery append() Method inserts content AT THE END of the selected HTML elements.



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("p").append(" <b>Appended text</b>.");
  });
  $("#btn2").click(function(){
    $("ol").append("<li>Appended item</li>");
  });
});
</script>
</head>
<body>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
<button id="btn1">Append text</button>
<button id="btn2">Append list items</button>
</body>
</html>
```



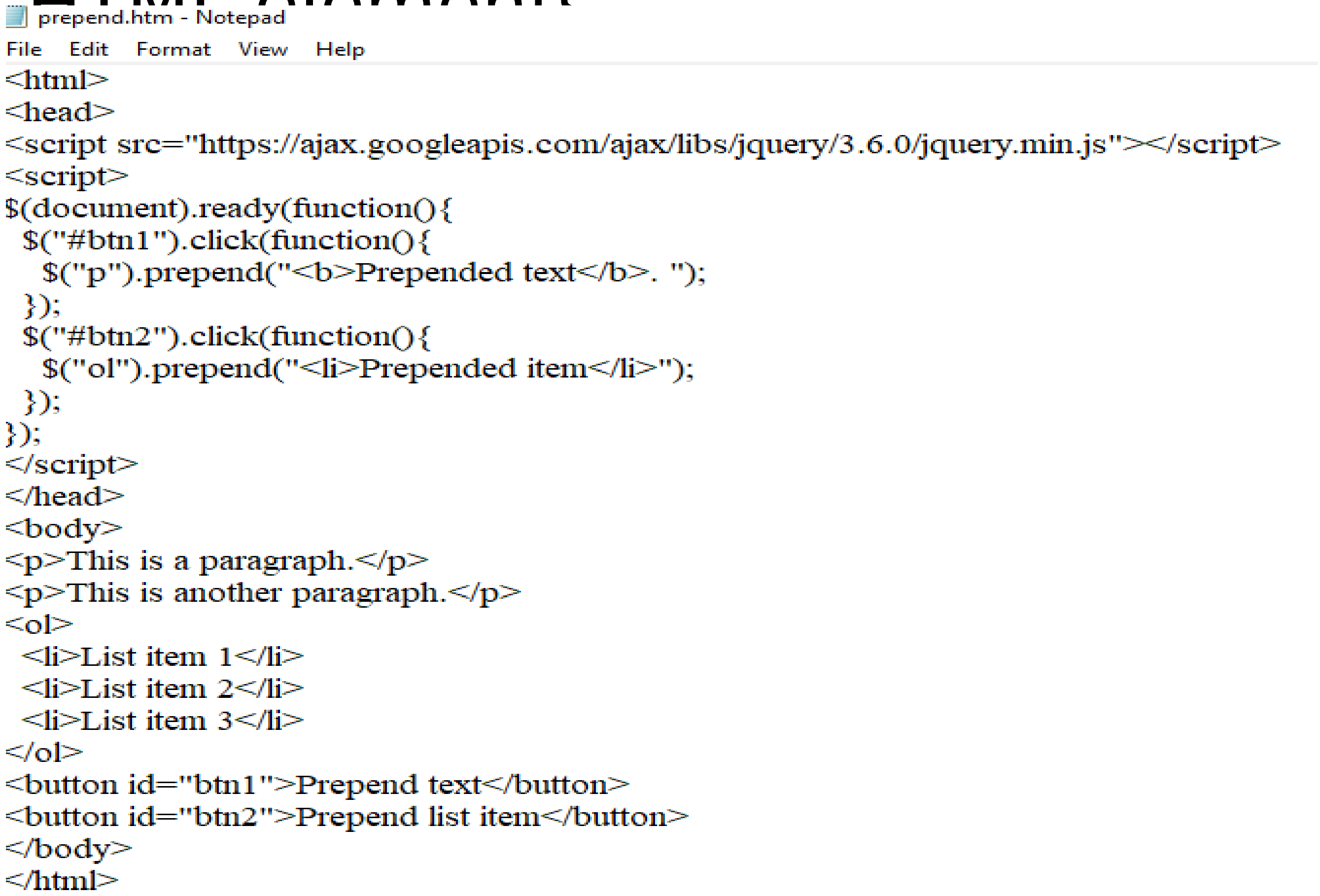
This is a paragraph.

This is another paragraph.

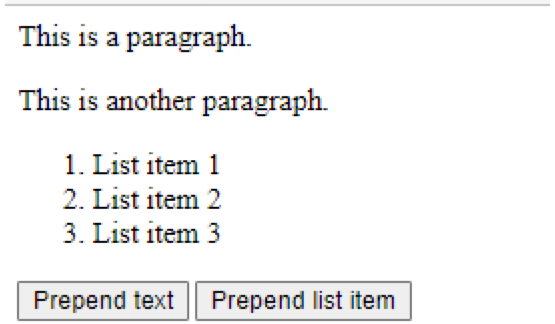
1. List item 1
2. List item 2
3. List item 3

# jQuery prepend() Method

- It inserts content AT THE BEGINNING of the selected HTML elements



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("p").prepend("<b>Prepended text</b>. ");
  });
  $("#btn2").click(function(){
    $("ol").prepend("<li>Prepended item</li>");
  });
});
</script>
</head>
<body>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
<button id="btn1">Prepend text</button>
<button id="btn2">Prepend list item</button>
</body>
</html>
```



This is a paragraph.

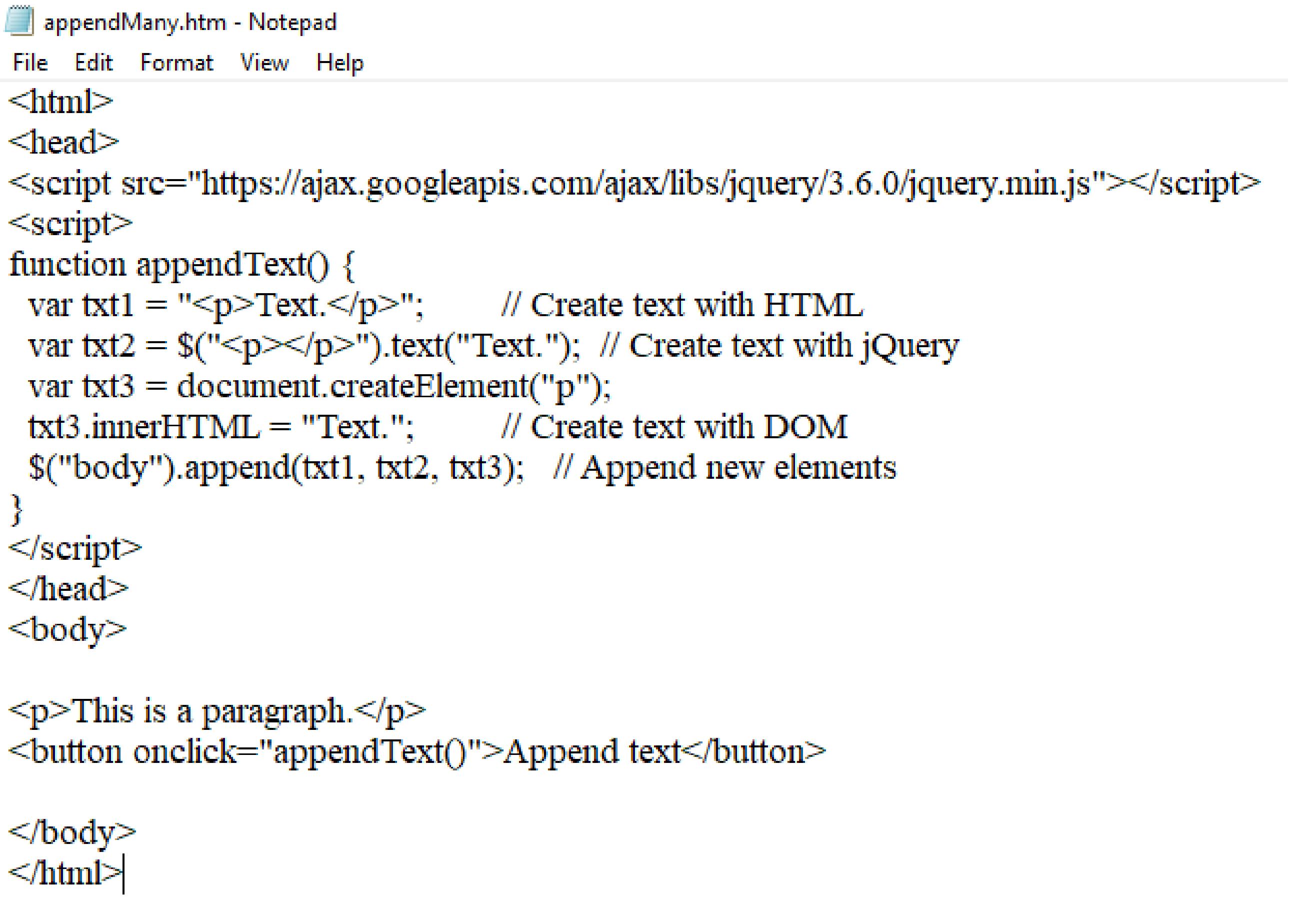
This is another paragraph.

1. List item 1
2. List item 2
3. List item 3

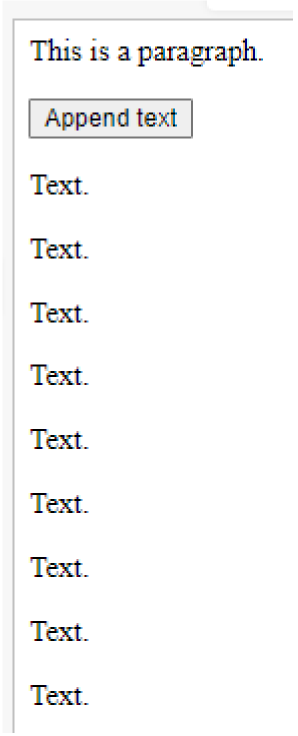
# Add Several New Elements With append()

- both the append() and prepend() methods can take an infinite number of new elements as parameters. The new elements can be generated with text/HTML (like we have done in the examples above), with jQuery, or with JavaScript code and DOM elements.
- In the following example, we create several new elements. The elements are created with text/HTML, jQuery, and JavaScript/DOM. Then we append the new elements to the text with the append() method.
- This would have worked for prepend() too.

# Demonstration to Add Several New Elements With append()



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
function appendText() {
  var txt1 = "<p>Text.</p>";        // Create text with HTML
  var txt2 = $("<p></p>").text("Text.");  // Create text with jQuery
  var txt3 = document.createElement("p");
  txt3.innerHTML = "Text.";         // Create text with DOM
  $("body").append(txt1, txt2, txt3);   // Append new elements
}
</script>
</head>
<body>

<p>This is a paragraph.</p>
<button onclick="appendText()">Append text</button>

</body>
</html>
```

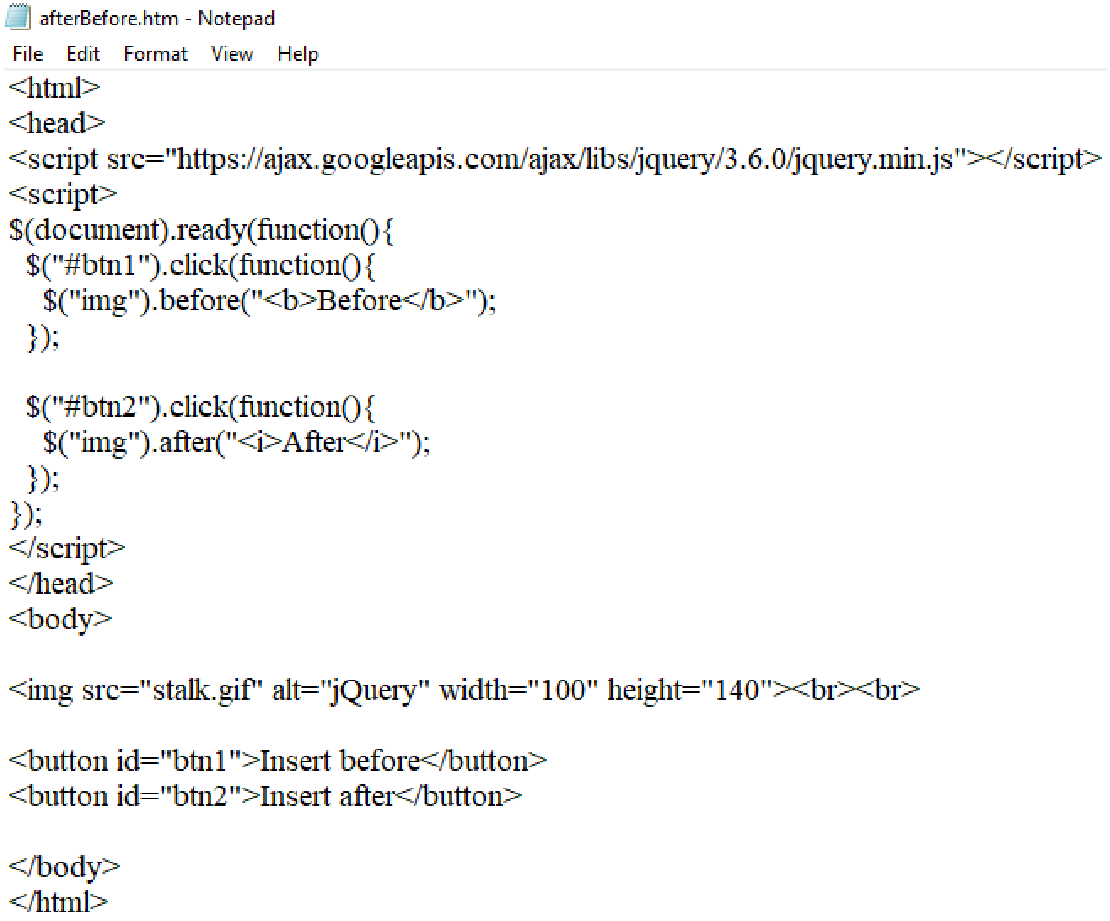


```
This is a paragraph.

Text.

Text.

Text.

Text.

Text.

Text.

Text.

Text.

Text.
```

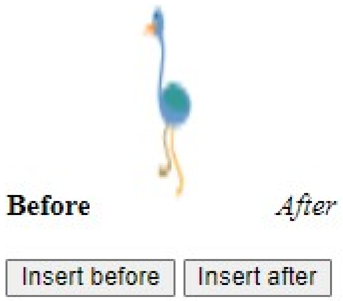
# jQuery after() and before() Methods

- The jQuery after() method inserts content AFTER the selected HTML elements.
- The jQuery before() method inserts content BEFORE the selected HTML elements.

# Demonstration of jQuery after() and before() Methods



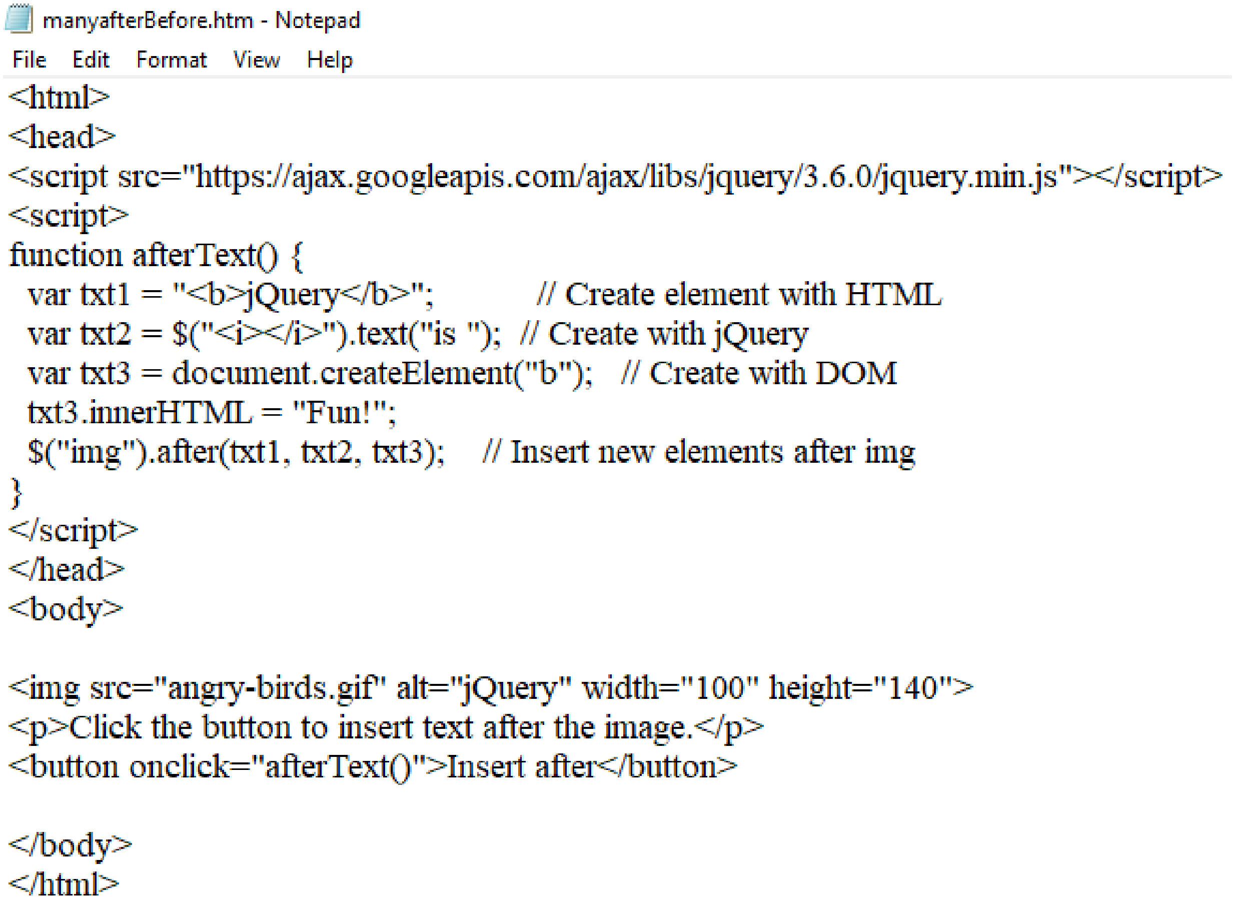
```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("img").before("<b>Before</b>");
  });

  $("#btn2").click(function(){
    $("img").after("<i>After</i>");
  });
});
</script>
</head>
<body>

<img src="stalk.gif" alt="jQuery" width="100" height="140"><br><br>

<button id="btn1">Insert before</button>
<button id="btn2">Insert after</button>

</body>
</html>
```



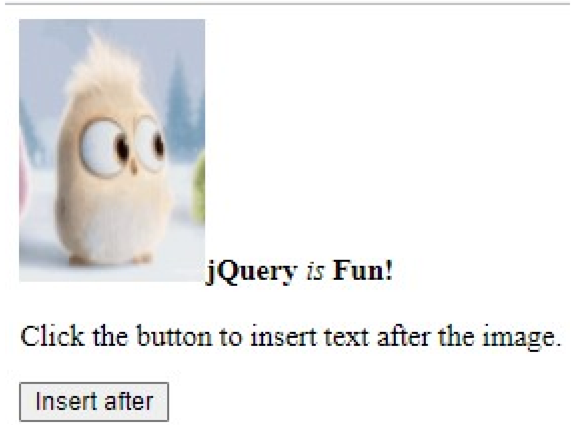
# Add New Elements With after() and before()

- Both after() and before() methods can take an infinite number of new elements as parameters. The new elements can be generated with text/HTML (like we have done in the example above), with jQuery, or with JavaScript code and DOM elements.
- In the following example, we create several new elements. The elements are created with text/HTML, jQuery, and JavaScript/DOM. Then we insert the new elements to the text with the after() method
- This would have worked for before() also

# Demonstration of adding New Elements with after()



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
function afterText() {
  var txt1 = "<b>jQuery</b>";           // Create element with HTML
  var txt2 = $("<i></i>").text("is ");  // Create with jQuery
  var txt3 = document.createElement("b");   // Create with DOM
  txt3.innerHTML = "Fun!";
  $("img").after(txt1, txt2, txt3);    // Insert new elements after img
}
</script>
</head>
<body>

<img src="angry-birds.gif" alt="jQuery" width="100" height="140">
<p>Click the button to insert text after the image.</p>
<button onclick="afterText()">Insert after</button>

</body>
</html>
```



**jQuery** ***is*** **Fun!**

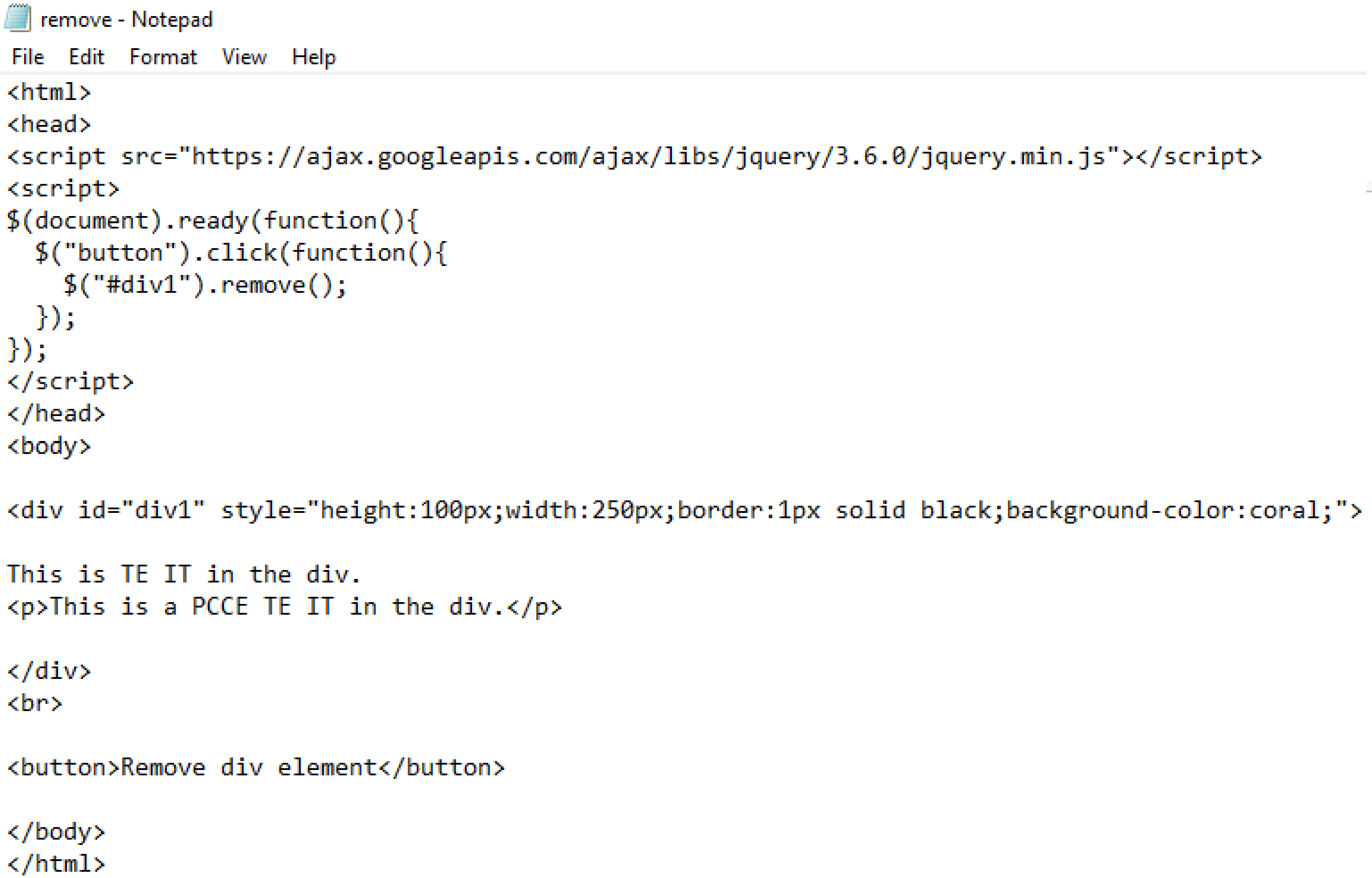
Click the button to insert text after the image.

# jQuery - Remove Elements

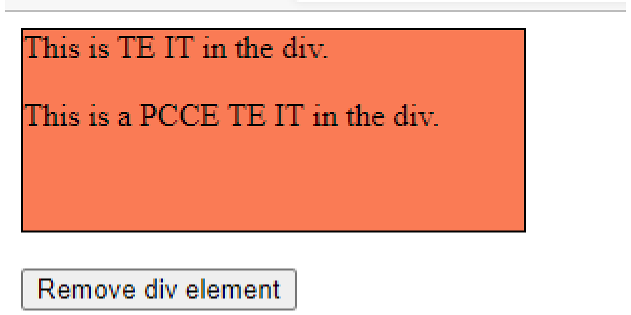
- To remove elements and content, there are mainly two jQuery methods:
- remove()- Removes the selected element (and its child elements)
- empty()- Removes the child elements from the selected element

# jQuery remove() Method

- The jQuery remove() Method removes the selected element(s) and its child elements.



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").remove();
  });
});
</script>
</head>
<body>

<div id="div1" style="height:100px;width:250px;border:1px solid black;background-color:coral;">

This is TE IT in the div.
<p>This is a PCCE TE IT in the div.</p>

</div>
<br>

<button>Remove div element</button>

</body>
</html>
```



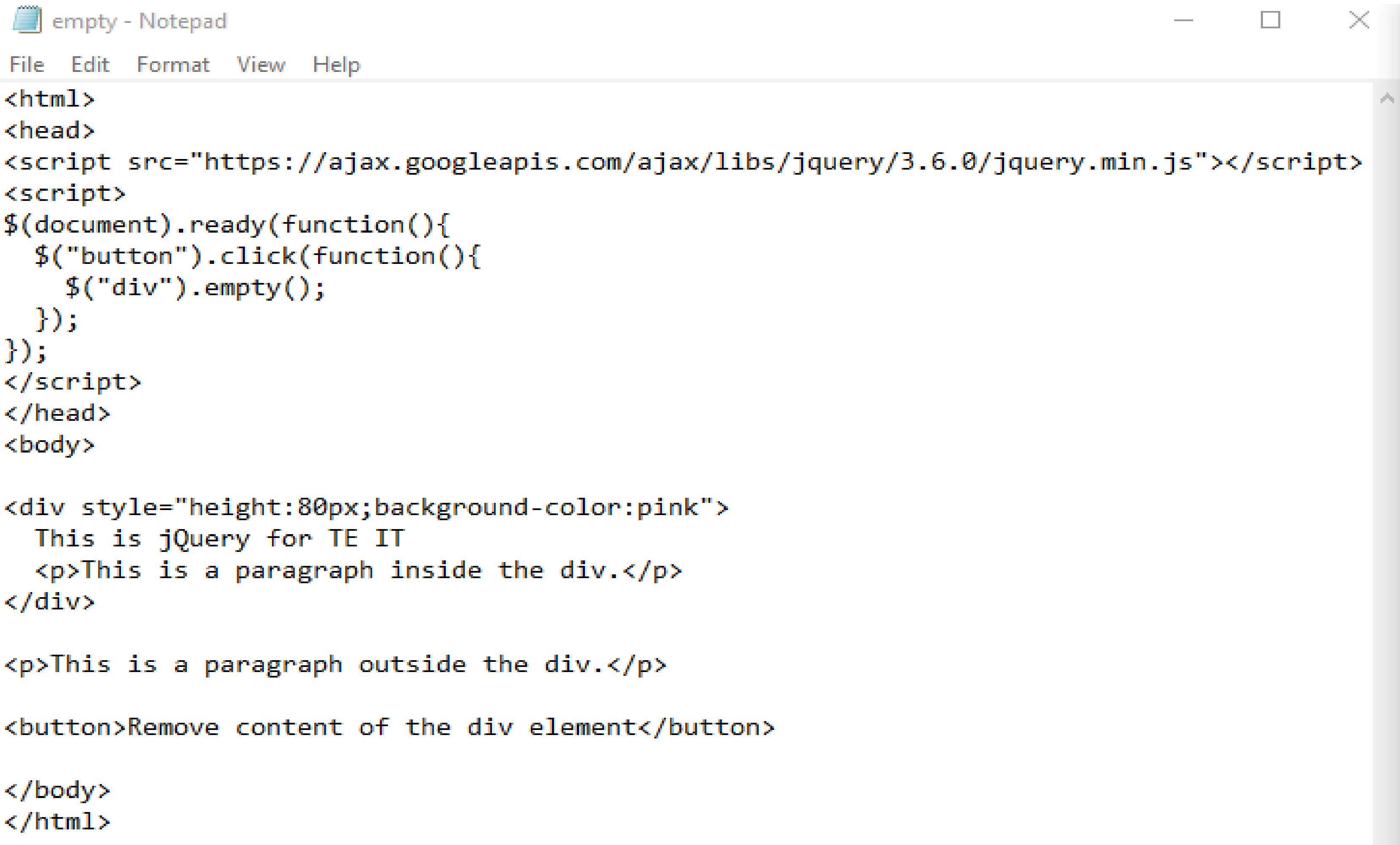
This is TE IT in the div.

This is a PCCE TE IT in the div.

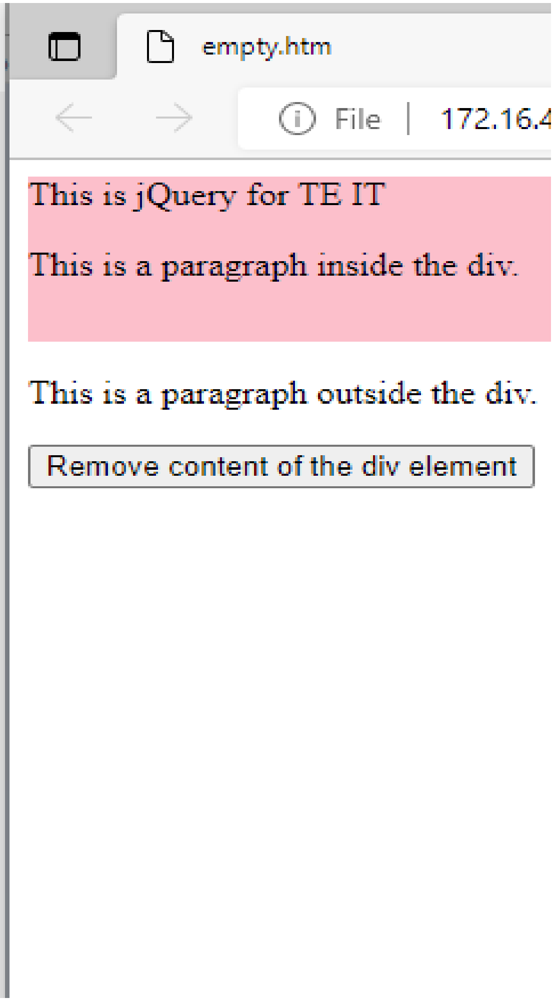
# jQuery empty() Method

- jQuery empty() Method removes all child nodes and content from the selected elements. This method does not remove the element itself, or its attributes.
- Syntax $(*selector*).empty()

# jQuery empty() Method



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").empty();
  });
});
</script>
</head>
<body>

<div style="height:80px;background-color:pink">
  This is jQuery for TE IT
  <p>This is a paragraph inside the div.</p>
</div>

<p>This is a paragraph outside the div.</p>

<button>Remove content of the div element</button>

</body>
</html>
```



This is jQuery for TE IT

This is a paragraph inside the div.

This is a paragraph outside the div.

# Assignment 2

- Q3 ) Demonstrate how to animate height using jQuery animate method (5 marks)
- Q4 ) Demonstrate jQuery chaining using the css(), slideUp(), and slideDown() methods (5 marks)

- *Assignment Announced to students : AA :04th May 2022*
- *Assignment to be Submitted by students : AS: 11th May 2022*