jQuery introduction
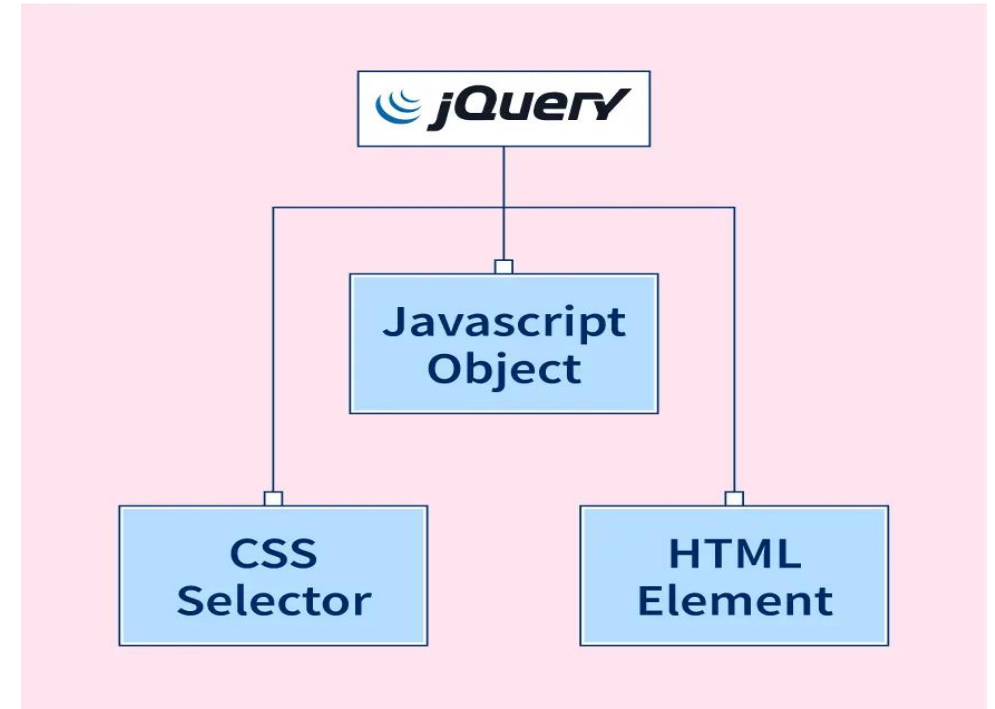
- **jQuery is a lightweight, "write less, do more", JavaScript library.**

- **The purpose of jQuery is to make it much easier to use JavaScript on your website.**

- jQuery takes **a lot of common tasks** that require many lines of JavaScript code to accomplish, and **wraps them into methods** that you can call with a single line of code.

- jQuery also **simplifies a lot of the complicated things** from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

<u>Adding jQuery to Your Web Pages</u>

There are several ways to start using **jQuery** on your web site.

1) Download the jQuery library 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):

```
<head>
    <script src="jquery-3.6.4.min.js"></script>
</head>
```

(or)

2) Include jQuery from a CDN, like Google:

- If you don't want to download and host jQuery yourself, you can include it from a **CDN (Content Delivery Network).**

```
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>
</head>
```

jQuery introduction

```html
<html>
<head>
 <title>First jQuery Example</title>
 <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
 </script>
 <script type="text/javascript"
language="javascript">

 $(document).ready(function() {
$("p").css("background-color", "cyan");
 });


</script>
 </head>
<body>
<p>The first paragraph is selected.</p>
<p>The second paragraph is selected.</p>
<p>The third paragraph is selected.</p>
</body>
</html>
```
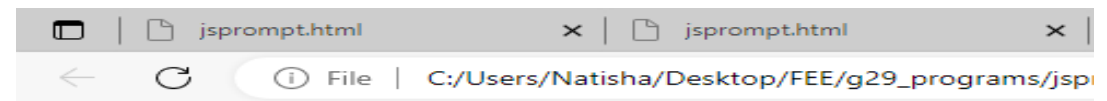
$(document).ready() and $()

▪ The code inserted between $(document).ready() is **executed only once** when page is ready for JavaScript code to execute.

▪ In place of **$(document).ready(),** we can use shorthand notation $() only.

▪ function() { $("p").css("background-color", "cyan"); }

▪ It changes the background-color of all <p> tag or paragraph to cyan.

jQuery Selectors

jQuery Selectors

- Used to **select and manipulate HTML elements**.

- jQuery selectors are used to "find" (or select) HTML elements based on their **name, id, classes, types, attributes, values** of attributes and much more.

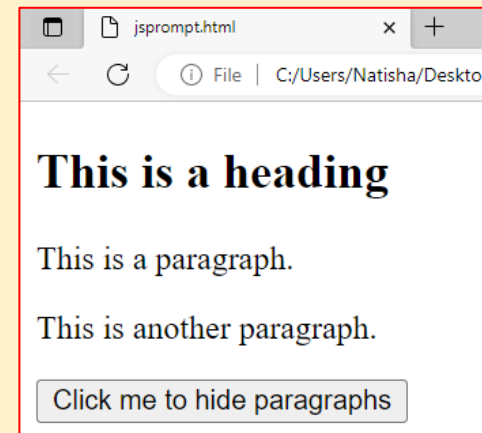- All selectors in jQuery start with the **dollar sign** and **parentheses**: **$()**.

The element Selector:

- The jQuery element selector **selects elements based on the element name.**
- We can select all **<p>** elements on a page like this:
      $("p") .

Ex.
When a user clicks on a button, all **<p>** elements will be hidden

```html
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/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>
```
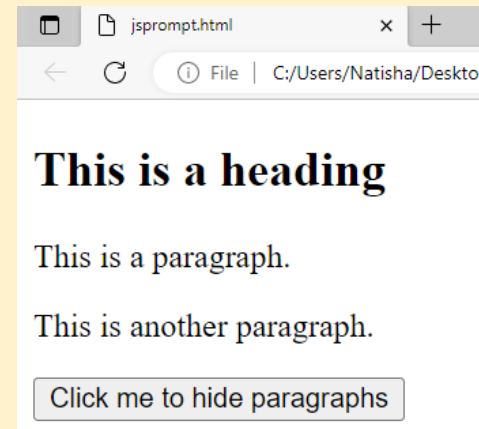
jQuery Selectors

The #id Selector:

- 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:

  `$("#test")` .

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#test").hide();
    });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>
<button>Click me to hide paragraphs</button>
</body> </html>
```
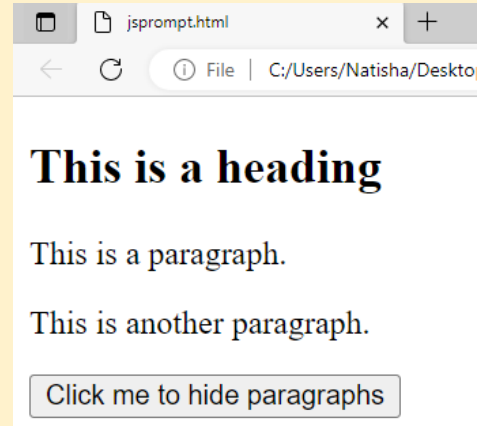
jQuery Selectors

The .class Selector

- The jQuery .class selector **finds elements with a specific class.**

- To find elements with a specific class, write a period character, followed by the name of the class:

  $(".test")

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $(".test").hide();
  });
});
</script>
</head>
<body>
<h2 class="test">This is a heading</h2>
<p class="test">This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me to hide paragraphs</button>
</body> </html>
```

This is a heading

This is a paragraph.

This is another paragraph.

Click me to hide paragraphs

jQuery Events

- Events are the **actions** that can be detected by your web application.
- They are used to create dynamic web pages. An event **shows the exact moment when something happens.**

These are some examples of events.

- **A mouse click**
- **An HTML form submission**
- **A web page loading**
- **A keystroke on the keyboard**
- **Scrolling of the web page etc.**

Syntax for event methods:

- Most of the DOM events have an equivalent jQuery method.
- To assign a click events to all paragraph on a page, do this:

$("p").click ();

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

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

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

jQuery click()

- When we click on an element, the **click event** occurs and once the click event occurs it execute the click () method or attaches a function to run.
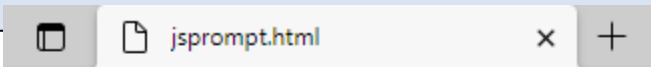
Syntax:

 $(selector).click()

- It is used to trigger the click event for the selected elements.

- **$(selector).click(function)**

```html
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("p").click(function(){
        alert("This paragraph was clicked.");
    });
});
</script>
</head>
<body>
<p>Click on the statement.</p>
</body>
</html>
```
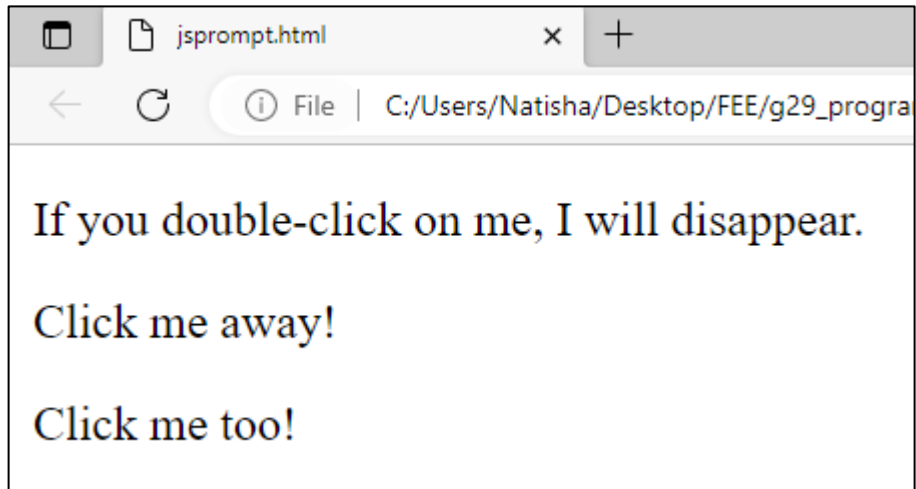
jsprompt.html

File | C:/Users/Natisha/Desk

Click on the statemen

This page says

This paragraph was clicked.

OK

jQuery Events- **dblclick()**

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:**


```html
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/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>
```
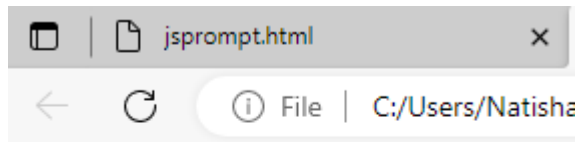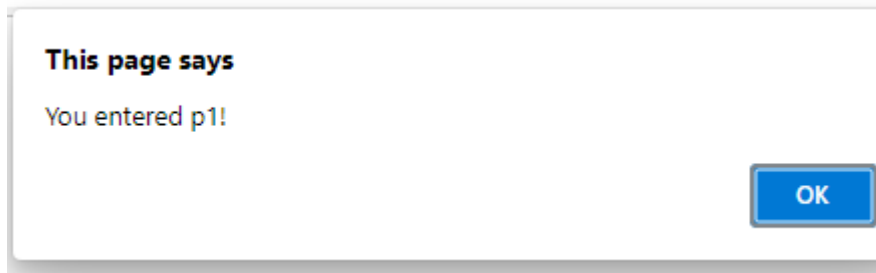
mouseenter()

- The mouseenter() method attaches an event handler function to an HTML element.

- **The function is executed when the mouse pointer enters the HTML element.**


```html
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/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>
```

jQuery Objects

- The jQuery object is a **collection of DOM elements** and **behaves like a special array.**
- Everything in jQuery is an object.
- When we create a new element or select an existing element the jQuery returns those elements in a collection.

The syntax of creating jQuery empty object:

```
var obj = { };
```
or
The syntax of creating an object using properties

```
var obj = { property1:value1, property2:value2... propertyN:valueN };
```

or
The syntax of creating an object using selector

```
var obj = $( "selector" );
```

Parameters:

- **property** – It specifies the property for the object to assign.

- **value** – It specifies the value to set for the specific property of the object. The property and value are separated by the colon(:) and each property-value pair is separated by the comma(,).

- **selector** – It specifies the element to select to return the object. it can be a single element or multiple elements.

Return value:

The return value of the object is **the object if selector element found**, else return **an empty object if selector element not found.**

jQuery Objects

Working on the JQuery object with Example

The JQuery object can be created as..

var obj = ["One", "Two", "Three", "Four"];

```html
<html lang = "en">
<head>
<script src = "https://code.jquery.com/jquery-3.5.0.js"></script>
<title  > This is an example for jQuery object
</title>
<style>
#h3{
width : 150px;
height : 100px;
border : 2px solid red;
font-size : 20px;
}
</style>
</head>
<body>
<h3> This is an Example for jQuery object: </h3>
<h3 id = "h3" > This is a example heading. </h3>
<p > This is a paragraph example. </p>
<button onclick="obj()"> click here to create
and display jQuery object </button>
<div id = "div1"> </div>
<div id = "div2"> </div>
<script>
function obj()
{
var headings = $("h3");
// disply the number of h3 tags on the page.
$("#div1").text("The no.of h3 are:"+headings.length);
// Selecting only the first h1 element on the page
$("#div2").text("The first h3 heading is:" +$("h3")[0]);
}
</script>
</body></html>
```

jQuery Objects

Example of object to an object with some properties

```html
<!doctype html>
<html lang = "en">
<head>
<meta charset = "utf-8">
<script src = "https://code.jquery.com/jquery-3.5.0.js"></script>
<title > This is an example for jQuery object </title>
<style>
#p1 {
color : blue;
}
#p2 {
color : red;
background-color : yellow;
}
</style>
</head>
<body>
<h3> This is an Example for jQuery object: </h3>
<p id = "p1"> </p>
<button onclick = "checkRes()" > Click here to create and display the object and its properties. </button>
<p id = "p2"> </p>
<script>
var obj = { Sub1: "JavaScript", Sub2: "CSS", Sub3: "jQuery", Sub4: "Java" };
function checkRes()
{
$("#p2" ).text( "The object properties are : " +obj.Sub1+ " " +obj.Sub2+ " " +obj.Sub3+ " " +obj.Sub4);
}
</script>
</body>
</html>
```