Web Programming Technologies

Harshita Maheshwari

Session-8

jQuery

write less, do more.

Topics to be covered…

- ➢ Introduction to jQuery
- ➢ Why we use jQuery
- ➢ Ready function
- ➢ jQuery Selectors
- ➢ jQuery HTML
- ➢ jQuery Effects
- ➢ jQuery Events
- ➢ jQuery Form Validation
- ➢ jQuery UI

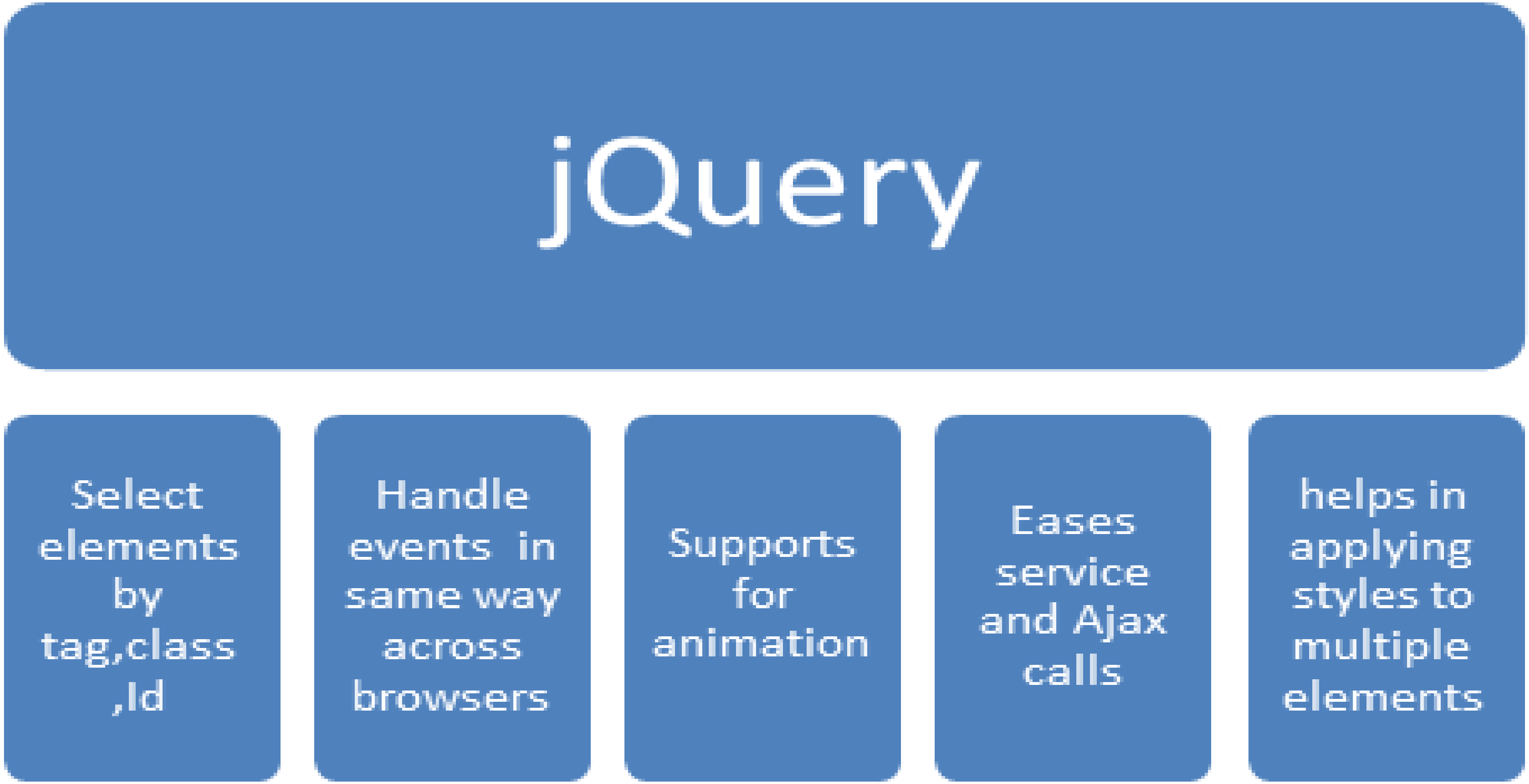
Introduction to jQuery

jQuery is a lightweight JavaScript library that emphasizes interaction between JavaScript and HTML

It is cross-platform and supports different types of browsers

Developed in 2006 by John Resig at Rochester Institute of Technology

What can jQuery Do



jQuery

Select elements by tag,class ,Id

Handle events in same way across browsers

Supports for animation

Eases service and Ajax calls

helps in applying styles to multiple elements

Why jQuery

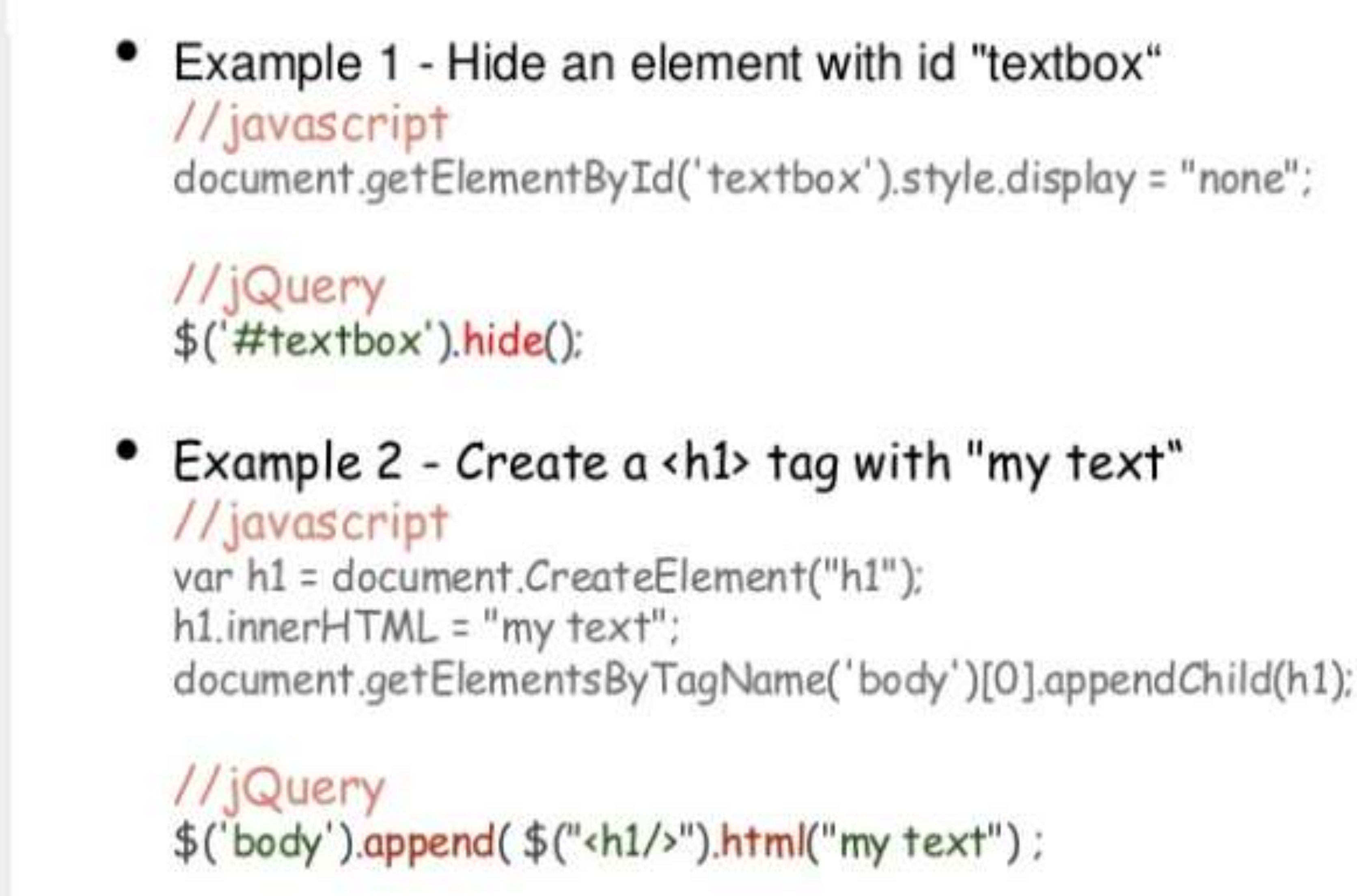
jQuery is like a pre-packaged set of JavaScript routines that you may have otherwise needed to write yourself, packaged in an easy-to-use way.

It uses CSS syntax for selection of element.

jQuery makes animated applications just like Flash

jQuery pages load faster

jQuery vs. JavaScript



- Example 1 - Hide an element with id "textbox“

```
//javascript
document.getElementById('textbox').style.display = "none";

//jQuery
$('#textbox').hide();
```

- Example 2 - Create a <h1> tag with "my text"

```
//javascript
var h1 = document.CreateElement("h1");
h1.innerHTML = "my text";
document.getElementsByTagName('body')[0].appendChild(h1);

//jQuery
$('body').append( $("<h1/>").html("my text") ;
```

Getting started with jQuery

Two ways to access it:

-Download it and reference:

```
<script src="jquery.js"></script>
```

(in the same directory)

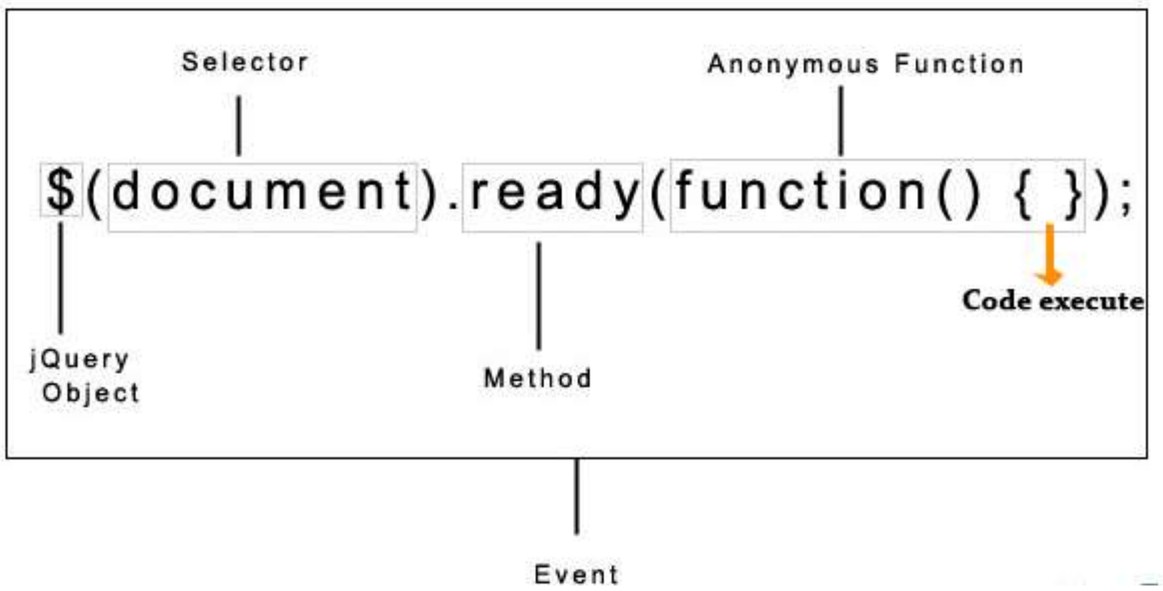
-Reference the JS file in your HTML (CDN):

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.js">
</script>
```

$(document).ready()

A page can't be manipulated safely until the document is "ready."

jQuery detects this state of readiness for you. Code included inside $(document).ready() will only run once the page Document Object Model is ready for JavaScript code to execute.



```
$(document).ready(function() { });
```

Basic Selectors

jQuery selectors allow you to select and manipulate HTML element(s).

- TagName

document.getElementsByTagName("tagName");

$("tagName")- $("div"), $("p"), $("div"),.....

- Tag ID

document.getElementById("id");

$("#id") - $("#name"), $("#address")

- Tag Class

document.getElementsByClassName("className");

$(".className") - $(".comment"), $(".code")

- To select all elements - $("*")

Selectors - Combined

$("tagName,.className")

Example: $("h1,.mainTitle")

$("tagName,.className,#tagId")

Example: $("h1,.mainTitle,#firstHeading")

	Examples:
$("selector:first")	$("p:first")
$("selector:last")	$("p:last")
$("selector:odd")	$("p:odd")
$("selector:even")	$("p:even")
$("selector[attribute]")	$("p:[name]")

Condition filters - Form filters

$("selector:visible")

$("selector:disabled")

$("selector:enabled")

$("selector:checked")

$("selector:button")

$("selector:file")

$("selector:input")

$("selector:text")

$("selector:password")

$("selector:checkbox")

$("selector:submit")

$("selector:reset")

:checkbox selector

```
<script>
$(document).ready(function(){
$("button").click(function(){
  $("input:checkbox").each(function(k,v){
  alert(k+v+v.value);
 });  });  });
</script>
```

```
$(document).ready(function(){
      $(":checkbox").click(function () {
          var x = "";
          $("#span1").text("");
          $(":checked").each(function (k, v) {
              x += v.value;
          });
          $("#span1").text(x);
      })
   });
</script>
```

Retrieve, Set and Remove attributes

```
$("selector").attr("name")

$("selector").attr("key", "val")

$("selector").attr(properties)


$("selector").removeAttr(attr)
```

Examples:

```
$("img").attr("src")

$("p").attr("class", "source")

$("img").attr({ "src" : "/path/",

"title" : "My Img" });

 $("div").removeAttr("class“ )
```

Class, HTML, Text, Value - Functions

```
$("selector").addClass("className")
$("selector").removeClass("className")
$("selector").toggleClass("className")
```

html() - Sets or returns the content of selected elements (including HTML markup)

```
$("selector").html()
 $("selector").html("html code")
```

text() - Sets or returns the text content of selected elements

```
$("selector").text()
$("selector").text("text content")
```

val() - Sets or returns the value of form fields

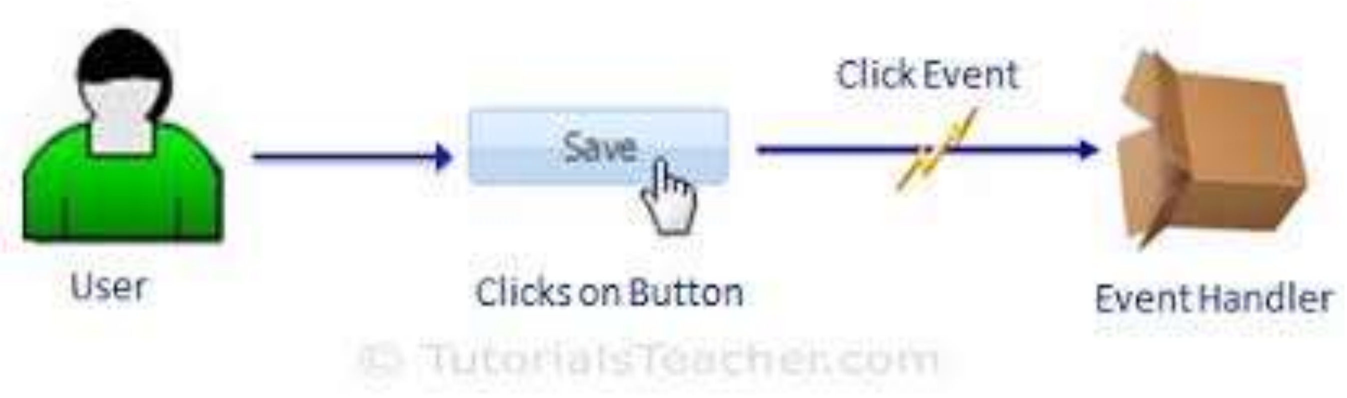
```
$("selector").val()
 $("selector").val("value")
```

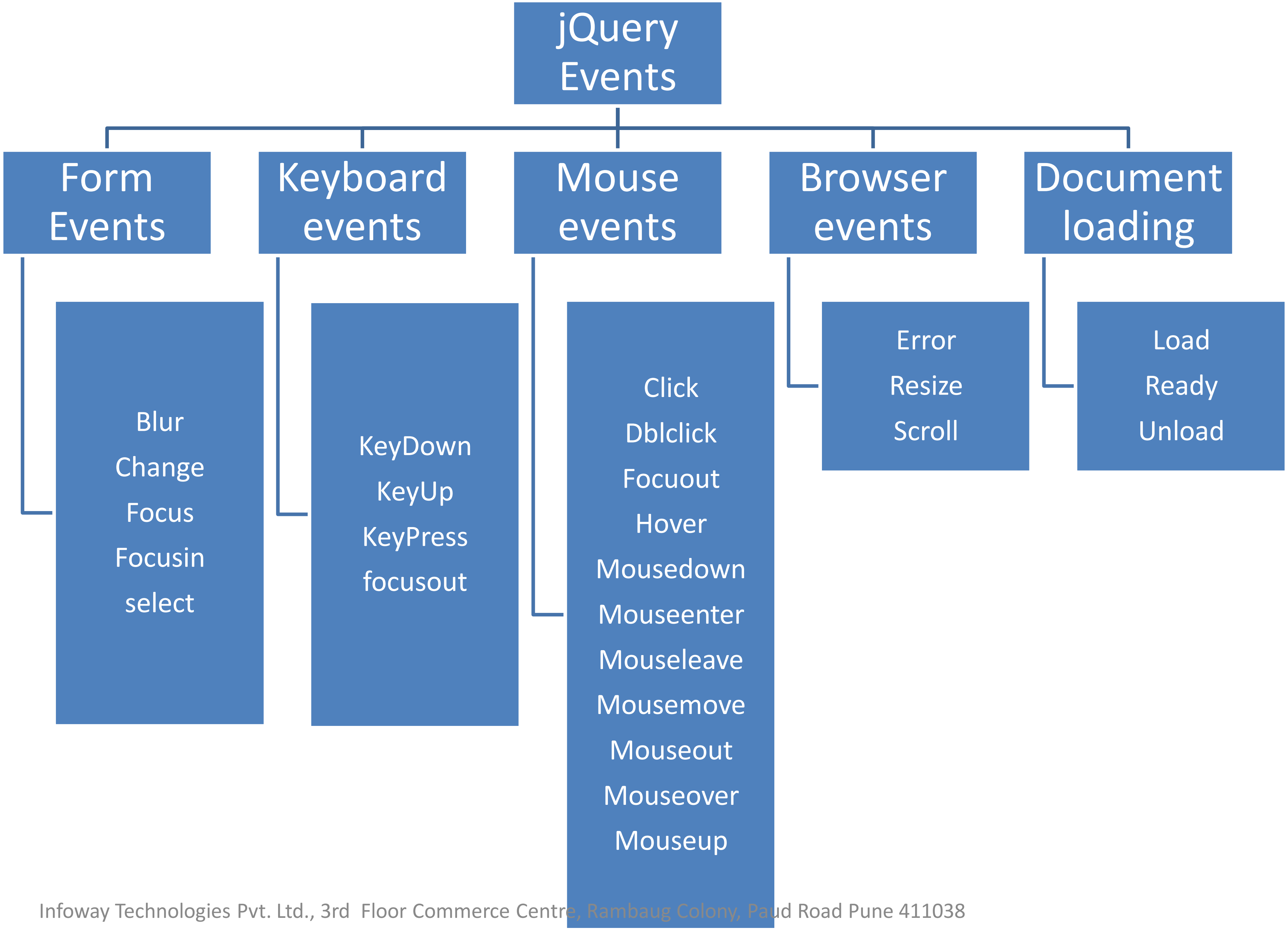
```
<script>
   $(document).ready(function () {
      $("#button1").click(function () {
           alert($("#p1").text());
           alert($("#div1").html());
           alert($("#text1").val());
       });
   });
</script>

         <script>
                $(document).ready(function () {
               $("#btn1").click(function(){
               $("#p1").text("Hello world!");
               });
               $("#btn2").click(function(){
               $("#p2").html("<b>Welcome</b>");
               });
               $("#btn3").click(function(){
               $(“#text1").val(“Infoway");
               });
                });
            </script>
```

jQuery Events

Events are a part of the Document Object Model (DOM) and every HTML element contains a set of events which can trigger JavaScript Code.





jQuery Events

Form Events

- Blur
- Change
- Focus
- Focusin
- select

Keyboard events

- KeyDown
- KeyUp
- KeyPress
- focusout

Mouse events

- Click
- Dblclick
- Focuout
- Hover
- Mousedown
- Mouseenter
- Mouseleave
- Mousemove
- Mouseout
- Mouseover
- Mouseup

Browser events

- Error
- Resize
- Scroll

Document loading

- Load
- Ready
- Unload

```
// Event setup using a convenience method
$( "p" ).click(function() {
console.log( "You clicked a paragraph!" );
});
```

```
// Equivalent event setup using the `.on()` method
$("p" ).on( "click", function() {
console.log( "You clicked a paragraph!" );
});
```

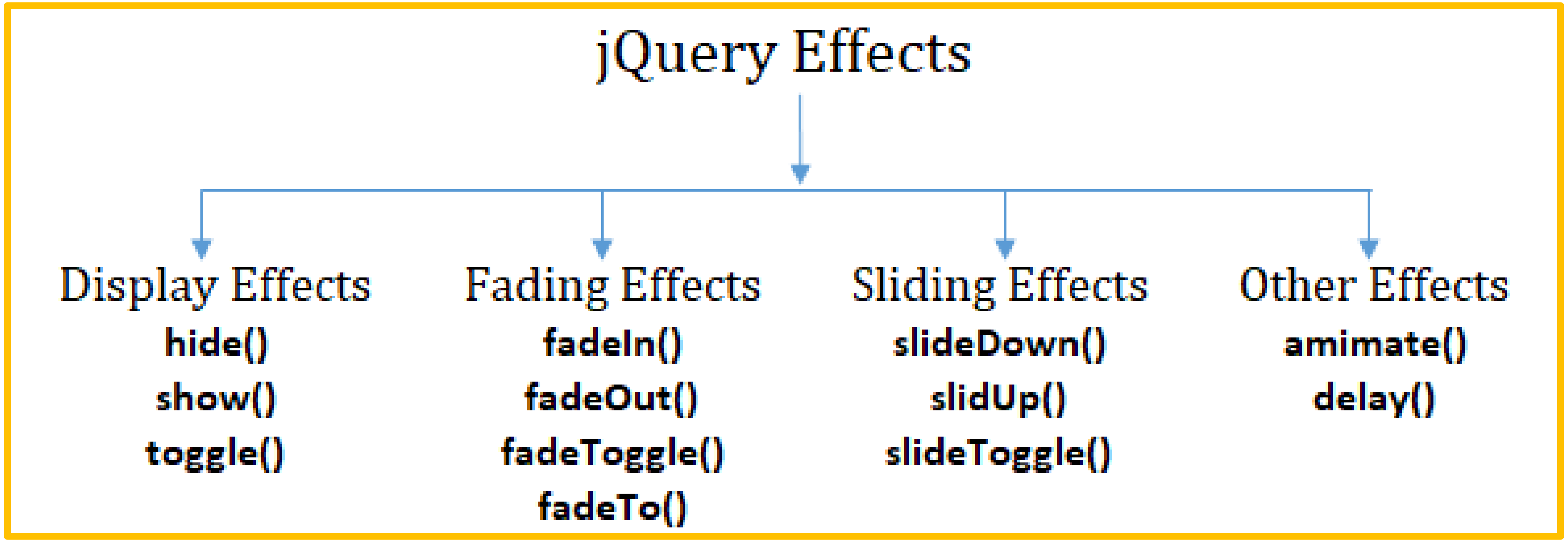
Focusout event

The focusout event is sent to an element when it, or any element inside of it, loses focus. Bind an event handler to the "focusout" event.

```
<script>
   $(document).ready(function () {
       $(":text").focusin(function () {
           $(this).css("background-color", “red");
       });
       $(":text").focusout(function () {
           $(this).css("background-color", "white");
       });
   });
</script>
```

jQuery Effects

The jQuery library provides several techniques for adding animation to a web page. These include simple, standard animations that are frequently used, and the ability to craft sophisticated custom effects.



jQuery Effects

Display Effects
hide()
show()
toggle()

Fading Effects
fadeIn()
fadeOut()
fadeToggle()
fadeTo()

Sliding Effects
slideDown()
slidUp()
slideToggle()

Other Effects
amimate()
delay()

Display Effects - .hide() and .show()

Hide the matched elements.
Show the matched elements

Speed is given in milliseconds; higher values indicate slower animations, not faster ones. The strings 'fast' and 'slow' can be supplied to indicate durations of 200 and 600 milliseconds, respectively.

Syntax:

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

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

jQuery toggle():

Syntax:

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

Fading Effects - .fadeIn(), .fadeOut()

Display the matched elements by fading them to opaque.
Hide the matched elements by fading them to transparent.

syntax

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

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

Speed is given in milliseconds; higher values indicate slower animations, not faster ones. The strings 'fast' and 'slow' can be supplied to indicate durations of 200 and 600 milliseconds, respectively. If any other string is supplied, or if the duration parameter is omitted, the default duration of 400 milliseconds is used.

Fading Effects - .fadeTo(), .fadeToggle()

Adjust the opacity of the matched elements.
Display or hide the matched elements by animating their opacity.

opacity: A number between 0 and 1 denoting the target opacity.

Syntax:
$(*selector*).fadeTo(*speed,opacity,callback*);
$(*selector*).fadeToggle(*speed,callback*);

```
<script>
  $(document).ready(function () {
    $("#toggle").on("click", function () {
      $("#div1").fadeTo("slow",0.5, function () {
          alert("Hello World!");
      });
    });
  });
</script>
```

Sliding Effects - .slideDown(), .slideUp()

Display the matched elements with a sliding motion.
Hide the matched elements with a sliding motion.

Syntax:
$(*selector*).slideDown(*speed,callback*);

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

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

Stop()

The jQuery stop() method is used to stop animations or effects before it is finished.

```
Syntax:
$(selector).stop();
```

Other Effects - .animate()

Perform a custom animation of a set of CSS properties.

Syntax:

$(*selector*).animate({*params*},*speed,callback*);

By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!

HTML

```
<button id="left">left</button>
<button id="right">right</button>
<div class="block"></div>
```

JS

```
$(document).ready(function(){
    $(".block").css({
        'position': 'absolute',
        'backgroundColor': "#abc",
        'left': '100px',
        'width': '90px',
        'height': '90px',
        'margin': '5px' });
    $("#left").click(function(){
        $(".block").animate({left: "-=50px"}, "slow");
    });
    $("#right").click(function(){
        $(".block").animate({left: "+=50px"}, "slow");
    });
});
```

jQuery-CSS

These methods get and set CSS-related properties of elements.

Set a CSS Property:
Syntax:

```
css("propertyname","value");
```

```
$("p").css("background-color", "yellow");
```

jQuery - Add Elements

- 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 - Remove Elements

remove() - Removes the selected element (and its child elements)

empty() - Removes the child elements from the selected element

jQuery - Widgets

A jQuery UI widget is a specialized jQuery plug-in.Using plug-in, we can apply behaviours to the elements.

Autocomplete

Enable to provides the suggestions while you type into the field.

Menu

Menu shows list of items.

Datepicker

It is to open an interactive calendar in a small overlay.

Select menu

Enable a style able select element/elements.

Tooltip

Its provides the tips for the users.

Tabs

It is used to swap between content that is broken into logical sections.

Thank You