Slide 1 — Introduction to jQuery

What is jQuery?

- jQuery is a **JavaScript library**
- It simplifies **client-side scripting**
- It allows developers to write **less code** and achieve **more functionality**
- jQuery makes web pages **interactive and dynamic**

Purpose of jQuery

- To reduce complexity of JavaScript
- To provide ready-made functions
- To improve development speed

Key Point:

jQuery does not replace JavaScript; it makes JavaScript easier to use.

Slide 2 — Why jQuery is Used

Problems with Traditional JavaScript

- Requires more lines of code
- Complex syntax for beginners
- Different behavior in different browsers
- Difficult DOM manipulation

Advantages of jQuery

- Simple and readable syntax
- Faster development
- Cross-browser compatibility
- Easy event handling
- Smooth animations and effects

Real-World Usage

- Button interactions
- Form validation
- UI effects
- Bootstrap functionality

Slide 3 — jQuery and JavaScript Relationship

JavaScript

- A programming language
- Used to control browser behavior
- Powerful but complex for beginners

jQuery

- A library written in JavaScript
- Provides reusable functions
- Beginner-friendly

Important Concept:

jQuery works on top of JavaScript and cannot function without it.

Slide 4 — Adding jQuery to a Web Page

Using jQuery Library

- jQuery must be included before using it
- CDN (Content Delivery Network) is commonly used

Example:

```
<script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
```

Important Rules

- jQuery file must be loaded **before custom JavaScript**
- Internet connection is required when using CDN

Slide 5 — Document Ready Function

What is Document Ready?

- Ensures the HTML page is fully loaded
- Prevents errors caused by missing elements

Why It Is Important

- jQuery code may fail if elements are not loaded
- Ensures safe execution of scripts

Syntax:

```
$(document).ready(function(){
    // jQuery code
});
```

Simple Explanation:

Run jQuery only after the page is completely loaded.

Slide 6 — jQuery Basic Syntax

General Syntax

```
$(selector).action();
```

Explanation

- `$` represents jQuery
- `selector` selects an HTML element
- `action()` performs an operation

Example:

```
$("#btn").hide();
```

Meaning:

- Select the element with id `btn`
- Hide it

SLIDE 7 — jQuery Selectors (With Multiple Examples)

What is a Selector?

- A selector is used to **select one or more HTML elements**
- jQuery selectors work similarly to **CSS selectors**
- They tell jQuery **where to apply an action**

◆ Common jQuery Selectors

1. ID Selector
```
$("#title")
```

Selects the element with id `title`

2. Class Selector
```
$(".card")
```

Selects all elements with class `card`

3. Tag Selector
```
$("p")
```

Selects all `<p>` elements

◆ More Selector Examples (Explain One by One)

```
$("h1")                  // Select all h1 elements
$("input")               // Select all input fields
$("button")              // Select all buttons
$("div")                 // Select all divs
$(".box")                // Select all elements with class box
$("#loginBtn")           // Select element with specific id
$("ul li")               // Select all li inside ul
$("input[type='text']")  // Select text input fields
$("a")                   // Select all links
$("img")                 // Select all images
```

◆ Why Selectors Are Important?

- Without selectors, jQuery cannot find elements
- Every jQuery action starts with a selector

Rule to remember:
No selector → No action

◆ UPDATED SLIDE 8 — jQuery Events (With Multiple Examples)

What is an Event?

- An event occurs when a user **interacts with the web page**
- jQuery listens for events and performs actions

◆ Common jQuery Events

1. click

Triggered when an element is clicked

```
$("#btn").click(function(){
    alert("Button clicked");
});
```

◆ More Event Examples (Explain Slowly)

```
$(".btn").click(function(){
    $(this).hide();
});
```

➡ Hides the clicked button

```
$("#box").mouseenter(function(){
    $(this).css("background-color", "red");
});
```

➡ Mouse enters the box

```
$("#box").mouseleave(function(){
    $(this).css("background-color", "blue");
});
```

➡ Mouse leaves the box

```
$("#input").focus(function(){
    $(this).css("border", "2px solid green");
});
```

➡ Input field gets focus

```
$("#input").blur(function(){
    $(this).css("border", "2px solid gray");
});
```

➡ Input field loses focus

```
$("#form").submit(function(){
    alert("Form submitted");
});
```

➡ Form submission event

```
$(window).resize(function(){
    alert("Window resized");
});
```

➡ Browser window resized

```
$(document).scroll(function(){
    console.log("Page scrolled");
});
```

➡ User scrolls the page

```
$("input").change(function(){
    alert("Input value changed");
});
```

➡ Input value changes

```
$("#btn").dblclick(function(){
    alert("Double clicked");
});
```

➡ Double-click event

◆ **Key Teaching Point:**

Events allow websites to **respond to user actions**

◆ UPDATED SLIDE 9 — Basic jQuery Functions (More Examples)

Show / Hide

```
$("#box").hide();
$("#box").show();
```

Toggle

```
$("#box").toggle();
```

Change Text

```
$("#text").text("Hello jQuery");
```

Change HTML Content

```
$("#box").html("<b>Bold Text</b>");
```

Change CSS

```
$("#box").css("color", "red");
$("#box").css("font-size", "20px");
```

Add / Remove Class

```
$("#box").addClass("active");
$("#box").removeClass("active");
```

Fade Effects

```
$("#box").fadeIn();
$("#box").fadeOut();
```

Slide Effects

```
$("#box").slideUp();
$("#box").slideDown();
```

Get Value from Input

```
let value = $("#name").val();
```

Set Value to Input

```
$("#name").val("Sarim");
```