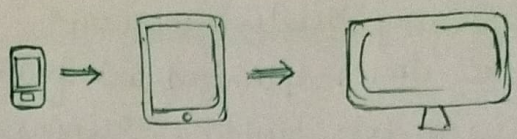
MOBILE FIRST



What is Mobile First?

- It is a web and application design strategy that prioritizes designing for mobile devices (smartphones, tablets) before scaling upto larger screens (like desktops and laptops).
- In this approach, the design process starts with the constraint and requirements of small screens, ensuring that essential features and content are accessible and optimized for mobile users.

Why "Mobile First"? : 2012-16, sales increased & ppl spend more time on internet from mobile ends.

Terms to know : RWD (Responsive Web Design) → web to fit automatically acc. to screens

Progressive Advancement → build from basic version (mobile) to advance version (laptop).

Graceful Degradation → build from advance version (laptop) to basic version (mobile).

What is Mobile Web?

- Refers to website and internet services accessed through mobile devices such as smartphones and tablets using mobile browsers.
- These applications and websites are specifically designed and optimized for the unique characteristics (smaller screen size, touch based navigation).

Understanding Mobile Devices and Desktop.

Aspects	Mobile Device	Desktop Device / Computers
Screen size	Smaller (3-6 inches)	Larger (13-30 inches).
Resolution	Lower	Higher.
Interaction	Touchscreen, gesture (tap, swipe, pinch)	Mouse keyboard, shortcuts.
Portability	Highly portable, used on-the-go	Stationary, requires dedicated workspace.
Connectivity	WiFi & cellular	Wired / Wireless broadband.
Performances	Optimized for efficiency, less processing power	High processing power, better for heavy tasks
Examples	Smartphones, Tablets, e-Readers, Smart watches, Fitness bands, etc.	Laptops, Desktop PC's, All-in-one Computers, etc.

Introduction to JQuery Mobile framework

- JQuery Mobile is a touch-optimized web framework built on top of jQuery library. It is used for building responsive & accessible mobile web applications that work across all smartphones, tablets and desktop devices.
- JQuery Mobile is a HTML5 CSS3 based UI system.

Features:

→ Write less, do more
→ Open source & cross platform compatible framework
→ Support user friendly I/P
→ Responsive
→ Ajax Navigation

How to use? ① Downloading package files ② Using CDN link.

Advantages:

→ Ease of Learning & Development
→ Responsive & Scalable design
→ Touch friendly UI Widgets
→ Open Source
→ Cross-platform & Cross Browser compatibility.
→ Minimal Scripting Required

Disadvantages:

→ Limited Customisation
→ Not suitable for complex Designs
→ Page load Overhead.
→ Appl^n build on jQuery mobiles are slower on mobiles.
→ Performance limitations.
→ Non-Native look and feel

Use Cases:

- Prototyping and building simple to moderately complex mobile web apps quickly
- Projects where wide device and browser support is essential.
- Apps that require touch-friendly, accessible interfaces without deep native integration.

What is CDN (Content Delivery Network)?

- CDN is a geographically distributed network of servers that work together to deliver web content - such as images, videos, stylesheets, JS files and entire web page quickly and efficiently to users based on their geographic locations.
- Instead of all users accessing your website's files from single, central server; a CND caches and serves those files from multiple servers located closer to users around the world.
- Its powerful because:

→ Faster Website loading
→ Reduce Server load
→ Improve availability
→ Global Reach
→ Easy Integration
→ Scalable (Handles high traffic and spikes with ease)

load jquery from CDN

```
<script src = "https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

Pages

- Fundamental building blocks for structuring mobile web applications.
- Each "page" is distinct view, defined within HTML using specific attributes and markup.
- Page is divided into 3 parts: header, content, and footer.
- eg.

```
<div data-role = "page" id = "home">                          ] Header
    <div data-role = "header"><h1> Home </h1> </div>
    <div role = "main" class = "ui-content">                  ] Content
        <p> Welcome to the Home Page ! </p>
    </div>
    <div data-role = "footer"><h4> Footer </h4></div>         ] footer
</div>
```

data-role = "page" → the page displayed on browser.
data-role = "header" → creates toolbar at top of the page.
data-role = "main" → defines content of the page (text, image, buttons, forms, etc).
"ui-content" class → adds extra margin & padding inside page content.
data-role = "footer" → creates toolbar at bottom of the page.

Types of pages:

1) Single Page – created in HTML document using a standard way of writing a template
2) Multipage Template – can be included in the single HTML document, which loads together by adding multiple divs with data-role = "page"
3) Dialogs Page – Modal dialogs open content in interactive overlay above the page.

Icons

jQuery Mobile provides set of built-in icons, which can be used with buttons, listview buttons which will make the button more attractive.

Icon set → sets the icon in the button. (data-icon)
Positioning Icons → specifies the position of icon in the button. (data-iconpos)
Icon-only → displays only an icon in the button (data-iconpos = "notext")
Icon-shadow → adds an icon shadow in your button. (data-iconshadow = "true")
Removing Circle → removes grey circle around the icon (ui-nodisc-icon)
Black or white icons → changes color of icons to black or white. (ui-alt-icon)
Combining alt & nodisc → combines alt & nodisc classes to the icon.

Transitions

- It allows to change property which occurs over specific duration and alters behaviour of each element from one state to another state by applying different styles for each state.

fade	slidefade	`<a href="#" data-transition="slide">`
flip	slideup	Slide to Page 2
pop	slidedown	`</a>`
slide	Turn	

Layouts

- Grid systems are used to create page layouts through a series of rows and columns.

Types of Grid

1) Grid - jQuery mobile grid system creates page layout through a series of rows and columns.
2) Buttons in Grid - Collection of buttons in grid format in jQuery mobile.
3) Custom responsive grid - The basic grid ~~exted~~ extended using media queries in CSS.

eg. of Basic Grid —

```
<div class="ui-grid-a">
    <div class="ui-block-a"><div class="ui-bar ui-bar-a">
     ColumnA </div></div>
    <div class="ui-block-b"><div class="ui-bar ui-bar-b">
     Column B </div></div>
</div>
```

eg. of Buttons in Grids

```
<div class="ui-grid-b">
  ~~<div class="ui-grid-a">~~
  <div class="ui block-a"><a href="#" class="ui-btn ui-corner-all">Yes
  </a></div>
  <div class="ui-block-b"><a href="#" class="ui-btn ui-corner-all">NO
  </a></div>
  <div class="ui-block-c"><a href="#" class="ui-btn ui-corner-all">Maybe
  </a></div>
</div>
```

eg. of custom grid

```
<div style = "display : flex ; flex-wrap : wrap;">
    <div style = "flex : 1 0 50% ; text-align : center ;"> A </div>
    <div style = "flex : 1 0 50% ; text-align : center ;"> B </div>
</div>
```

OR

```
<style>
  .custom-grid {
    display : flex ;
    flex-wrap : wrap ;
  }
  .custom-grid > div {
    flex : 1 0 50% ;
    padding : 10px ;
  }
  @media (min-width : 768px) {
    .custom-grid > div {
      flex : 1 0 25% ;
    }
  }
</style>
<div class = "custom-grid">
    <div> Block 1 </div>
    <div> Block 2 </div>
    <div> Block 3 </div>
    <div> Block 4 </div>
</div>
```

Widgets

- A small gadget or control of jQuery mobile application.
- They are pre-built, touch-friendly UI components that enhance functionality and appearance of mobile web applications.

Types of Widgets:

Buttons - clickable elements that can include text or images.

Support mini buttons with data-mini = "true"

inline buttons with data-inline = "true"

theming via data-theme

eg:
```
<a href = "#" data-role = "button">
    Click Me </a>
```

Checkbox - allows selection of multiple options.

eg:
```
<input type = "checkbox" name = "checkbox-1"
    id = "checkbox-1" />
<label for = "checkbox-1"> I agree </label>
```

Collapsible - Expandable / collapsible content blocks for organizing information.

use data-role = "collapsible" on a container.

The header becomes clickable element.

Content is hidden by default and expands when header is tapped.

eg.
```
<div data-role = "collapsible">
    <h3> Click to Expand </h3>
    <p> This content is hidden until you click the header </p>
</div>
```

Listview - Renders complex, customizable lists with support for icons, thumbnails and search filters.

Use data-role = "listview" on <ul> or

Supports split buttons, count bubbles and filterable lists

eg:
```
<ul data-role = "listview">
    <li><a href = "#"> Home </a></li>
    <li><a href = "#"> About </a></li>
</ul>
```

Slider – Lets user choose a value by sliding handle along a track.
Use <input type = "range"> with data-role = "slider"
Supports custom min, max and step values.

eg. <label for = "slider"> Volume : </label>
<input type = "range" name = "slider" id = "slider" min = "0" max = "100" value = "50" >

Popup – small overlay windows for messages, forms or custom content.
Use data-role = "popup" on a container.
Trigger with a link using data-rel = "popup"

```
<a href = "#" data-rel = "popup" class = "ui-btn" > Show Popup </a>
<div data-role = "popup" id = "myPopup">
    <p> This is popup dialog </p>
</div>
```

Events

The actions that happen on a web page (clicks, hover or keypress). jQuery mobile provides rich set of custom events that extend native browser events.

Events in jQuery mobile

1) Touch Events – these are triggered when user taps on an element or swipes over an element. jQuery Mobile provides several custom touch events :

tap

```
$("p").on("tap", function(){
    $(this).hide();
});
```

taphold

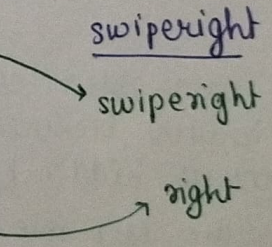
```
$("p").on("taphold ,function(){
    $(this).hide();
});
```

swipe

```
$("p").on("swipe", function(){
    $("span").text("Swip detected");
});
```

swipeleft

```
$("p").on("swipeleft", function(){
    alert("You swiped left!");
});
```



2) Orientation Events – triggered when the user changes orientation from portrait to landscape or vice versa.

- Portrait → vertical position landscape → horizontal position.

value of orientation property = 0 :: portrait mode
value of orientation property = 90° or -90° :: landscape mode.

- This event is attached to the window object.

eg. ~~$(window).on("orientationchange", function(event){ if(event.orienta~~
~~alert("Orientation is: " + event.orientation);~~
~~});~~

```
$(window).on("orientationchange", function(event){
  if(event.orientation == 0){
    alert("Portrait : Device is placed vertically");
  } else {
    alert("Landscape : Device is placed horizontally");
  }
});
```

3) Page events - triggered when a page is shown, hidden, created, loaded.
Common page events include:
pagebeforecreate, pagecreate, pageinit, pagebeforeshow, pageshow, pagebeforehide, pagehide, pageremove.

eg.
```
$(document).on("pagebeforecreate", function(event){
  alert("Page beforecreate event triggered!");
});
```

Forms

- jQuery Mobile enhances native HTML form elements to make them more touch-friendly and visually consistent across devices.
- Forms are used to collect data from end users and send or process in background for any functionality. Forms are made up with multiple inputs & button widgets
- Some form controls are:

- Text Inputs : <input type="text"> /<input type="email">
- Buttons : <button>, <input type="submit">
- Checkbox & Radio Buttons
- Select menus.
- Sliders : <input type="range">
- Switches

eg:
```
<form>
  <label for="username">Username:</label>
  <input type="text" name="username" id="username" placeholder="Enter username"/>
  <label for="remember">Remember Me</label>
  <input type="checkbox" name="remember" id="remember"/>
  <button type="submit" class="ui-btn ui-btn-b">Login</button>
</form>
```

Themes

- Themes control the look & feel of UI widgets, including colors, fonts and background
- jQuery Mobile provides default swatches named from a to e. Each swatch has a different color scheme.
- eg:

```
<div data-role="header" data-theme="b">
    <h1> My Header </h1>
</div>
```

data-theme = a means light theme
b means dark theme
c means gray
d & e means custom.

Header and Footer

- Headers & footers are implemented as toolbars using and .
- They can be positioned as:

i) Inline (default) - sits in the natural document flow.
ii) Fixed - remains positioned at the top & bottom of the page.
iii) Fullscreen - Hidden by default and shown on tap, useful for images, videos.

eg.

```
<div data-role="header" data-theme="b">
    <h1> Page Title </h1>
</div>

<div data-role="footer" data-theme="b" data-position="fixed">
    <h4> Footer Text </h4>
</div>
```

CSS Classes

- CSS is essential for designing mobile websites because it controls how your website looks and behaves across different screen sizes and devices.
Reasons css is imp: Responsive Design, Touch friendly UI, Performance Optimization, UX, Consistency Across Devices.

1) Global Class - apply styling globally (all)

```
eg. <div class="ui-content"> → adds padding & standard formatting to content area
      <p> This is global class applied to content area </p>
    </div>
```

2) Grid Class - creates responsive, column-based layouts.

```
eg. <div class="ui-grid-a"> creates 2 column layout
      <div class="ui-block-a"><div class="ui-bar ui-bar-a"> Col A </div></div>
      <div class="ui-block-b"><div class="ui-bar ui-bar-a"> Col B </div></div>
    </div>
```

define individual blocks inside the grid.

3) ~~Block Class~~ Button Class - used to style buttons

```
<a href="#" class="ui-btn ui-btn-b"> Click Me </a>
```

4) Theme class - defines look & feel of elements

```
<div class="ui-bar ui-bar-a">        → creates block with background & padding
                                     → applies theme "a" to the bar
    <h3> This is themed bar using theme A </h3>
</div>
```

Data Attributes

- These are HTML5 data-* attributes that jQueryMobile uses extensively to initialize and control widget behaviour.

- data-role - defines structural role eg. page, header, footer, listview (replace by)
- data-theme - selects theme swatch
- data-icon - assigns an icon
- data-position - for header & footer
- data-transition - slide, fade, pop

Building a simple Mobile Webpage

```
<!DOCTYPE html>
<html>
<head>
    <title> Mini Mobile Page </title>
    <link rel="stylesheet" href="link of jquery mobile.css">
    <script src="link of jquery.js"></script>
    <script src="link of jquery mobile.js"></script>
</head>
<body>
    <div data-role="page">
    <div data-role="header">
        <h1> Mini Page </h1>
    </div>
    <div data-role="content">
        <p> Hello Web ! </p>
        <a href="#" class="ui-btn"> Click Me </a>
    </div>
    <div data-role="footer">
        <h4> 2025 </h4>
    </div>
</div>
</body>
</html>
```