

ASP.NET Web Forms Cheat Sheet

.NET Framework 4.8.1 · security-fixes-only · System.Web · quick reference for irurueta docs

.aspx page skeleton

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Default.aspx.cs"
Inherits="App.Default" %>
<!DOCTYPE html>
<html><body>
  <form id="form1" runat="server">
    <asp:Label ID="lbl" runat="server" />
  </form>
</body></html>

// Default.aspx.cs
public partial class Default : Page {
  protected void Page_Load(object s, EventArgs e) {
    if (!IsPostBack) lbl.Text = "Hello";
  }
}
```

Page life-cycle order

PreInit → Init → InitComplete → PreLoad → Load → postback event handlers → LoadComplete → PreRender → PreRenderComplete → SaveStateComplete → Render → Unload
Controls: Init/Unload bottom-up; Load top-down.
Check IsPostBack in PreInit/Load, not before.

Postback / ViewState

```
Client posts via hidden __doPostBack(target,arg) +
__VIEWSTATE/ __EVENTVALIDATION hidden fields.

EnableViewState="false" // per control
ViewStateMode="Disabled" // page/control
<pages enableViewStateMac="true"
viewStateEncryptionMode="Always" />
```

Server-control catalogue

Label TextBox Button LinkButton ImageButton
DropDownList ListBox CheckBox(List)
RadioButton(List) Panel PlaceHolder HiddenField
Literal Calendar AdRotator MultiView/View Wizard
FileUpload Table

Validation controls

```
<asp:RequiredFieldValidator
ControlToValidate="txtName"
ErrorMessage="Required" runat="server" />
<asp:RegularExpressionValidator
ValidationExpression="\S+@\S+"
ControlToValidate="txtEmail" runat="server" />
<asp:ValidationSummary runat="server" />

Server: check Page.IsValid before processing. Group
with ValidationGroup="g1".
```

Master page + ContentPlaceHolder

```
<!-- Site.Master -->
<asp:ContentPlaceHolder ID="cph"
runat="server" />

<!-- Page.aspx -->
<%@ Page MasterPageFile="~/Site.Master" %>
<asp:Content ContentPlaceHolderID="cph"
runat="server">
  ...
</asp:Content>
```

GridView + data source

```
<asp:SqlDataSource ID="ds" runat="server"
ConnectionString="<%$ ConnectionStrings:Db
%>"
SelectCommand="SELECT Id,Name FROM Product"
/>
<asp:GridView ID="gv" runat="server"
DataSourceID="ds"
AutoGenerateColumns="false">
  <Columns>
    <asp:BoundField DataField="Name" />
  </Columns>
</asp:GridView>

ObjectDataSource binds to a business/DAL class
instead of raw SQL.
```

Data-binding expressions

```
<%# Eval("Name") %> <!-- read-only -->
>
<%# Bind("Name") %> <!-- 2-way -->
<%# XPath("author/@name") %>
```

State management

Store	Scope	API
ViewState	page, per user	ViewState["k"]
Control state	page, persistent	SaveControlState
Session	user, server	Session["k"]
Application	app, server	Application["k"]
Cache	app, expiring	Cache.Insert(...)
Profile	user, persisted	Profile.Prop
Cookie / QS	client	Request.Cookies/QueryString

Output caching

```
<%@ OutputCache Duration="60"
VaryByParam="id" VaryByCustom="browser" %>
```

global.asax events

```
void Application_Start(object s, EventArgs e)
void Application_BeginRequest(...)
void Application_AuthenticateRequest(...)
void Application_Error(object s, EventArgs e)
void Session_Start(object s, EventArgs e)
void Application_End(object s, EventArgs e)
```

Custom handler / module

```
public class MyHandler : IHttpHandler {
  public bool IsReusable => false;
  public void ProcessRequest(HttpContext ctx)
    => ctx.Response.Write("hi");
}
// web.config
<add verb="*" path="*.hi"
type="App.MyHandler" />

public class MyModule : IHttpModule {
  public void Init(HttpApplication app) =>
    app.BeginRequest += (s,e) => { };
  public void Dispose() { }
}
```

Forms authentication (web.config)

```
<authentication mode="Forms">
  <forms loginUrl="~/Login.aspx"
timeout="30" slidingExpiration="true" />
</authentication>
<authorization>
  <deny users="?" />
</authorization>

Provider model: Membership, Roles, Profile --
SqlMembershipProvider,
ActiveDirectoryMembershipProvider.
```

AJAX: ScriptManager + UpdatePanel

```
<asp:ScriptManager runat="server" />
<asp:UpdatePanel runat="server">
  <ContentTemplate>
    <asp:Button runat="server" OnClick="Go" />
  </ContentTemplate>
  <Triggers>
    <asp:AsyncPostBackTrigger ControlID="btn"
/>
  </Triggers>
</asp:UpdatePanel>

The full page life cycle still runs on every partial postback.
```

MVP for testability

IView interface + Presenter holding logic, the Page implements IView and delegates -- unit-test the presenter without a running HttpContext.