Click Here

Classic asp tutorial

ASP supports various development models including Classic ASP, ASP.NET Web Forms, ASP.NET MVC, ASP.NET Web Pages, ASP.NET API, and ASP.NET Core. Both ASP and ASP.NET are server-side technologies that enable code execution by an internet server when a browser requests a file. Classic ASP, introduced in 1998, uses the .asp extension and is typically written in VBScript. Given article text here Looking for a comprehensive guide to learn ASP.NET Core from scratch? This tutorial covers everything you need to know, including the differences between various ASP models and how to get started with coding. ASP.NET MVC is being merged into the new ASP.NET Core, so it's not covered in this tutorial. However, you can still explore other ASP models like Web API, Web Forms, and ASP.NET Core itself. To learn ASP, you don't have to pay hundreds or thousands of dollars for courses. This free online course will teach you the basics of ASP programming, including setting up a webserver, writing output, defining variables, and using arrays. The tutorial is divided into 12 lessons, covering topics such as IF-THEN statements, CASE statements, FOR-NEXT loops, DO-WHILE loops, and more. You'll also learn how to use the REQUEST object, send mail with ASP, keep your code modular, and optimize for search engines. Additionally, this site offers a complete database of functions, sample code, and other resources to help you build your ASP knowledge. Whether you're new to programming or looking to upgrade your skills, this tutorial is an excellent starting point. ASP pages are typically written in VBScript and have the .asp file extension, whereas ASP.NET pages use the .aspx extension and are usually developed in C#. The latest official version of ASP.NET is 4.6, while ASP.NET 5 was planned to be a major overhaul but has been replaced by ASP.NET Core. ASP.NET Web Pages follows the Single Page Application model, similar to PHP and Classic ASP, and is being merged into ASP.NET Core. ASP.NET MVC also uses the Model-View-Controller approach and is being integrated with ASP.NET Core as well. The platform includes features such as tracking progress, setting goals, creating learning paths, and even hosting personal websites for free. However, creating an account is optional. ASP.NET Web API handles Application Programming Interfaces, while ASP.NET Web Forms operates on an event-driven framework not part of the new ASP.NET Core framework. ASP.NET Core was introduced in 2016, integrating MVC, API, and Pages into a single application framework. using the ASP 'Response.Write' method, we can embed a variable in the HTML by placing its contents into the HTTP stream. Response.Write (strMessage) Next, we'll use the 'Response.Write' method to create a line break and display server time using the VBScript function 'Time()'. Response.Write ("The time on the server is: " & Time()) We'll close the server-side script tag after displaying server time. Finally, we need to finish the HTML web page by closing the body and HTML tags. We also need to call a file named 'my_first-asp-page.asp' and save it in an accessible directory through your web server, such as 'c:\inetpub\public_html'. To display the page, type ' in your web browser, replacing 'my_computer' with your actual computer name. After creating the first dynamic ASP web page, you can try some other Classic Tutorials. Private mstrDelimiter As String 'Delimiter between multipart/form-data (43 chars) Private CR As String ' ANSI Carriage Return Private LF As String ' ANSI Line Feed Private CRLF As String ' ANSI Carriage Return & Line Feed Private mobjFieldAry() As Object ' Array to hold field objects Private mlngCount As Long ' Number of fields parsed Private Sub RequestData Dim llngLength As Long ' Number of bytes received mlngBytesReceived = Request.TotalBytes mbinData = Request.BinaryRead(mlngBytesReceived) End Sub Private Sub ParseDelimiter() mstrDelimiter = MidB(mbinData, 1, InStrB(1, mbinData, CRLF) - 1) End Sub Private Sub ParseData() llngStart = 1 While Not llngStart = 0 llngEnd = InStrB(llngStart + 1, mbinData, mstrDelimiter) - 2 llngLength = llngEnd - llngStart lbinChunk = MidB(mbinData, llngStart, llngLength) Call ParseChunk(lbinChunk) llngStart = InStrB(llngStart + 1, mbinData, mstrDelimiter & CRLF) Wend End Sub Private Sub ParseChunk(ByRef pbinChunk As Binary) lstrName = ParseDisposition(pbinChunk) lstrFileName = ParseName(lstrName) lstrContentType = ParseContentType(pbinChunk) If lstrContentType = "" Then lstrValue = CStrU(ParseBinaryData(pbinChunk)) Else ' assume the field is binary End Sub Given article text here Private Function ParseDisposition(ByRef pbinChunk) Dim llngStart, llngEnd As Long llngStart = InStrB(llngStart + 22, pbinChunk, Chr(13) & Chr(10)) If llngStart = 0 Then Exit Function llngEnd = llngStart + 22 If llngStart >= llngEnd Then Exit Function llngLength = llngEnd - llngStart ParseDisposition = CStrU(MidB(pbinChunk, llngStart, llngLength)) End Function Private Function ParseName(ByRef pstrDisposition) Dim llngStart, llngEnd As Long llngStart = InStr(1, pstrDisposition, "name=""", vbTextCompare) If llngStart = 0 Then Exit Function llngEnd = InStr(llngStart + 6, pstrDisposition, """") If llngEnd = 0 Then Exit Function llngStart = llngStart + 6 If llngStart >= llngEnd Then Exit Function ParseName = Mid(pstrDisposition, llngStart, llngEnd - llngStart) End Function Private Function ParseFileName(ByRef pstrDisposition) Dim llngStart, llngEnd As Long llngStart = InStr(1, pstrDisposition, "filename=""", vbTextCompare) If llngStart = 0 Then Exit Function llngEnd = InStr(llngStart + 10, pstrDisposition, """") If llngEnd = 0 Then Exit Function llngStart = llngStart + 10 If llngStart >= llngEnd Then Exit Function ParseFileName = Mid(pstrDisposition, llngStart, llngEnd - llngStart) End Function Public Property Get Count() As Long Count = mlngCount End Property Public Default Property Get Fields(ByVal pstrName) As Object Dim llngIndex As Long If IsNumeric(pstrName) Then llngIndex = CLng(pstrName) If llngIndex > mlngCount - 1 Or llngIndex < 0 Then Err.Raise vbObjectError + 1, "clsUpload.asp", "Object does not exist within the ordinal reference." Exit Property End If Else Set Fields = mobjFieldAry(pstrName) End If End Property Given article text here Public Class clsField Public Name As String ' Field name in form Private mstrPath As String ' Full path to file on visitor's computer Public FileDir As String ' Directory where file was located on visitor's computer Public FileExt As String ' File extension (e.g. GIF) Public FileName As String ' File name Public ContentType As String ' Content/Mime type of file (e.g. image/gif) Public Value As String ' Unicode value of field (for non-file fields) Public BinaryData As Byte() ' Binary data for files Public Length As Integer ' Byte size of value or binary data Private mstrText As String ' Text buffer for binary data Public Property Get BLOB() As Byte() BLOB = BinaryData End Property Public Function BinaryAsText() As String ' Convert binary data to text Dim lbinBytes As Byte() If Length = 0 Or LenB(BinaryData) = 0 Then Exit Function If Not Len(mstrText) = 0 Then Return mstrText lbinBytes = ASCII2Bytes(BinaryData) mstrText = Bytes2Unicode(lbinBytes) BinaryAsText = mstrText End Function Public Sub SaveAs(pstrFileName As String) ' Save binary data to file Dim lobjStream As Object, lbinBytes As Byte() If Length = 0 Or LenB(BinaryData) = 0 Then Exit Sub Set lobjStream = Server.CreateObject("ADODB.Stream") lobjStream.Type = adTypeBinary lobjStream.Open() lbinBytes = ASCII2Bytes(BinaryData) lobjStream.Write(lbinBytes) lobjStream.SaveToFile(pstrFileName, adSaveCreateOverWrite) lobjStream.Close() Set lobjStream = Nothing End Sub Public Property Let FilePath(ByRef pstrPath As String) mstrPath = pstrPath FileExt = UCase(Mid(pstrPath, InStrRev(pstrPath, ".") + 1)) FileName = Mid(pstrPath, InStrRev(pstrPath, "\") + 1) FileDir = Mid(pstrPath, 1, InStrRev(pstrPath, "\") - 1) End Property End Class Looking forward to seeing everyone at the meeting tomorow and discussin our strategies. I'm developing a web application that requires users to upload photos or documents, which should be accessible by all users, similar to Facebook's upload feature. Can anyone suggest the most effective way to approach this task? Should I use JQuery or pure Javascript? Any recommendations or assistance would be greatly appreciated.