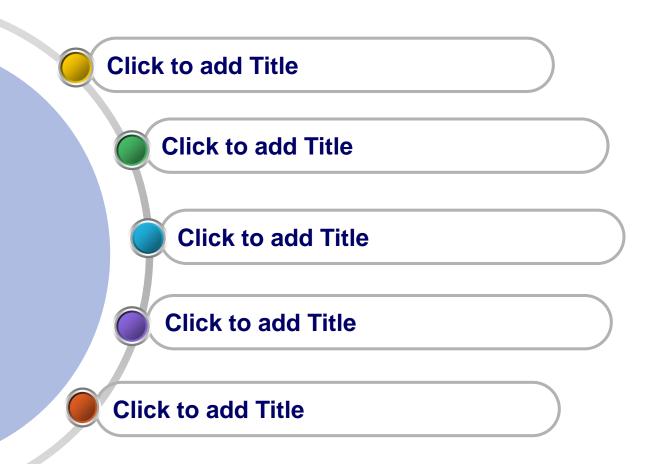


GV: ThS Phạm Thi Vương

Contents



Validation Controls

 Validation control kiểm tra tính đúng đắn của dữ liệu do client nhập vào trước khi trang được gửi về cho server

Validation control	Sử dụng khi
RequiredFieldValidator	Kiểm tra nếu dữ liệu trong control khác rỗng
CompareValidator	Kiểm tra nếu mục dữ liệu nhập trong control giống với control kh á c
RangeValidator	Kiểm tra nếu mục dữ liệu nhập trong control nằm trong khoảng 2 gi á trị
RegularExpressionValidator	Kiểm tra nếu mục dữ liệu nhập trong control thỏa 1 công thức định dạng chỉ định
CustomValidator	Kiểm tra tính đúng đắn của dữ liệu nhập vào control sử dụng client-side script hoặc a server-side code, hoặc cả 2
ValidationSummary	Hiển thị tất cả c á c lỗi kiểm tra xảy ra trong trang

Validation Controls

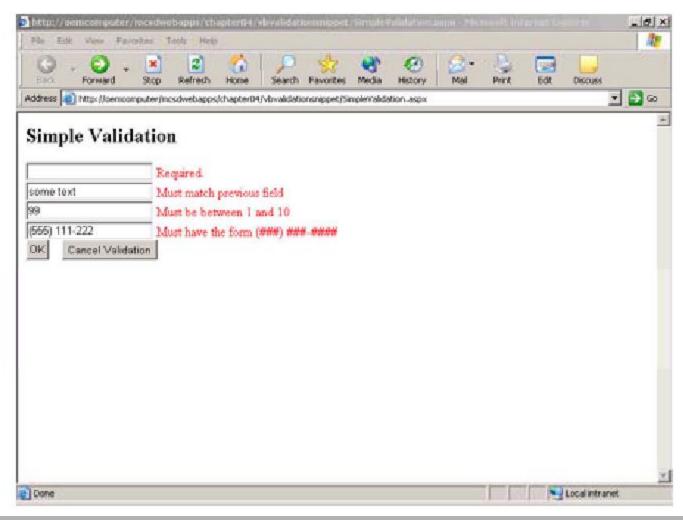
Sử dụng Validation Control

Các bước:

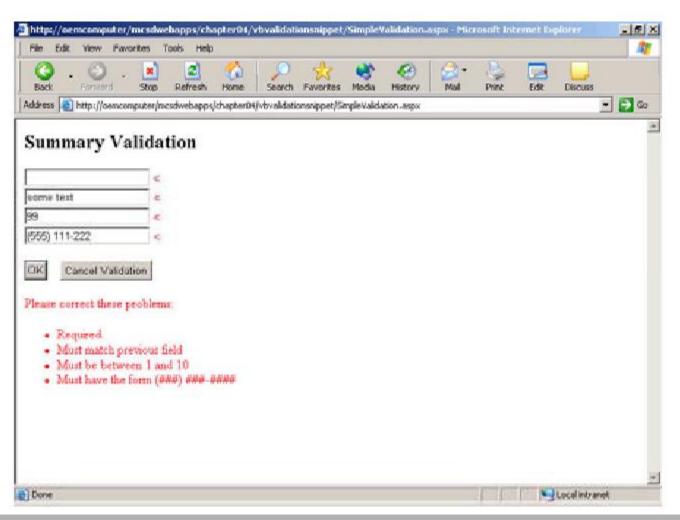
- Kéo thả 1 validate control vào Web form
- Thiết lập các thuộc tính cho validate control:
 - ControlToValidate là control ban muốn kiểm tra
 - ErrorMessage : Thông báo lỗi
 - Text : Hiển thị của validate control
- Sử dụng ValidationSummary control để hiền thị tất cả các lỗi xảy ra trong trang

Mặc dù việc kiểm tra xảy ra ở client, nhưng nó chỉ thực hiện khi có 1 sự kiện post-back xảy ra!

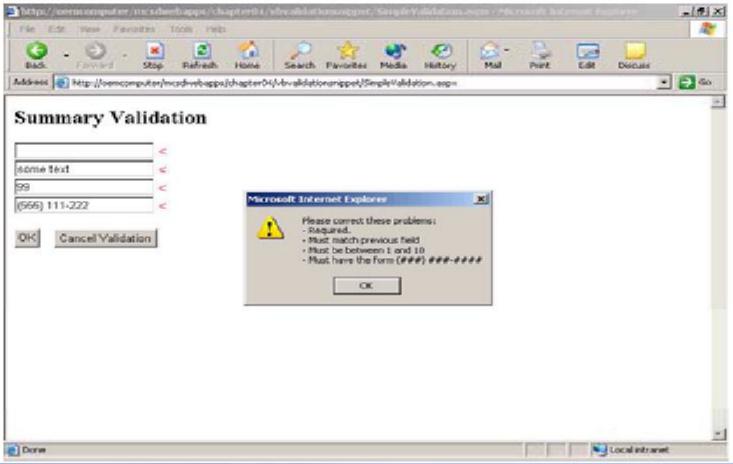
Đặt thuộc tính ErrorMessage cho Validate control



Sử dụng ValidationSummary



Sử dụng ValidationSummary control với ShowMessage=True



Company Logo

Required Field Validator

Thuộc tính	Ý nghĩa
ControlToValidate	Control dùng để thực hiện kiểm tra
InitialValue	Giá trị dùng để so sánh. Mặc định là rỗng.

Compare Validator

Thuộc tính	Ý nghĩa
ControlToValidate	Control dùng để thực hiện kiểm tra
ControlToCompare	Control dùng để so sánh
Operator	Toán tử so sánh: Equal, GreaterThan, GreaterThanEqual, LessThan, LessThanEqual, NotEqual, hoặc DataTypeCheck
ValueToCompare	Giá trị hằng số để so sánh ControlToValidate với Operator
Туре	Kiểm tra kiểu dữ liệu của ControlToValidate: String, Integer, Double, Date, Currency

Ví dụ - Kiểm tra Mật khẩu và Xác nhận mật khẩu

```
Enter Password:
    Reenter Password: •
                                           Passwords must match
Enter Password: <asp:TextBox ID="txtPass1" runat="server"</pre>
TextMode="password"></asp:TextBox></br>
Reenter Password: <asp:TextBox ID="txtPass2" runat="server"
TextMode="password"></asp:TextBox>
<asp:CompareValidator ID="compPass" runat="server"
       ControlToValidate="txtPass2"
       Operator="Equal"
       ControlToCompare="txtPass1"
       Text="Passwords must match" />
```

Ví dụ - Kiểm tra dữ liệu là số nhỏ hơn hoặc bằng 18

Age: 35 You are too old to view this site

Regular Expression Validator

```
Email abc Enter a valid Email
```

```
Email <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>

<asp:RegularExpressionValidator ID="valEmail" runat="server"
ControlToValidate="txtEmail"

ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*"
ErrorMessage="Enter a valid Email"
/>
```

Regular Expression Validator

Ký hiệu	Ý nghĩa
^ \$	Dấu hiệu bắt đầu và kết thúc một Expression
\t	Có chứa Ký tự Tab
\n	Có chứa Ký tự xuống đòng
	Có chứa Ký tự bất kỳ khác ∖n
[qwerty]	Có chứa Ký tự bất kỳ trong ngoặc vuông
[^qwerty]	Không chứa ký tự nào trong ngoặc vuông
[a-z]	Có chứa ký tự trong khoảng từ a đến z
\w	Có chứa một từ bất kỳ (word). Tương tự [a-zA-Z0-0]
\W	Có chứa một chuỗi bất kỳ không phải là một từ (nonword)
	Ноặс

Hủy bỏ việc Kiểm tra

- Để cho người dùng tự hủy bỏ việc kiểm tra nếu muốn
- Các bước:
 - Tạo một HTML Button control

```
<INPUT id="butCancel" onclick="Page_ValidationActive=false;"

type="submit" value="Cancel">
```

- Hůy Sự kiện validation: onclick="Page_ValidationActive=false;
- Và gửi trang về cho server
- Kiểm tra lại dữ liệu ở trên server
 - Kiểm tra thuộc tính Page.IsValid trong hàm xử lý sự kiện Page_Load

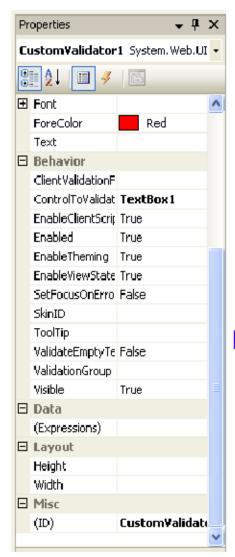
Page.IsValid

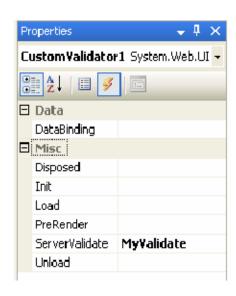
- Thuộc tính Page.IsValid kiểm tra xem các form đã thỏa các Validation Control hay không.
- Trả về true nếu tất cả đều được test thành công
- Trả về false, trong trường hợp ngược lại.

CustomValidator

- Sử dụng CustomValidator control
- Tự viết mã lệnh kiểm tra chạy trên server hoặc client
- Trên Server
 - Đặt mã lệnh kiểm tra trong hàm xử lý sự kiện ServerValidate
- Hoặc Trên Client
 - Chỉ định đoạn script kiểm tra cho thuộc tính
 ClientValidationFunction của CustomValidator

Ví dụ: Tự Kiểm tra trên Server

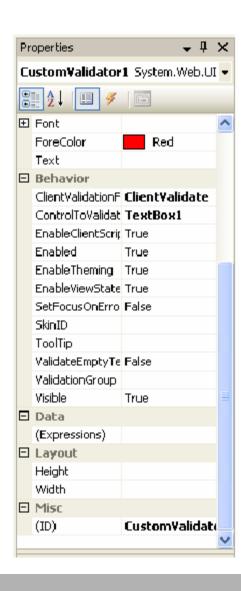




protected void MyValidate(object source, ServerValidateEventArgs args)

```
args.lsValid = false;
if (TextBox1.Text == "abc")
args.lsValid = true;
```

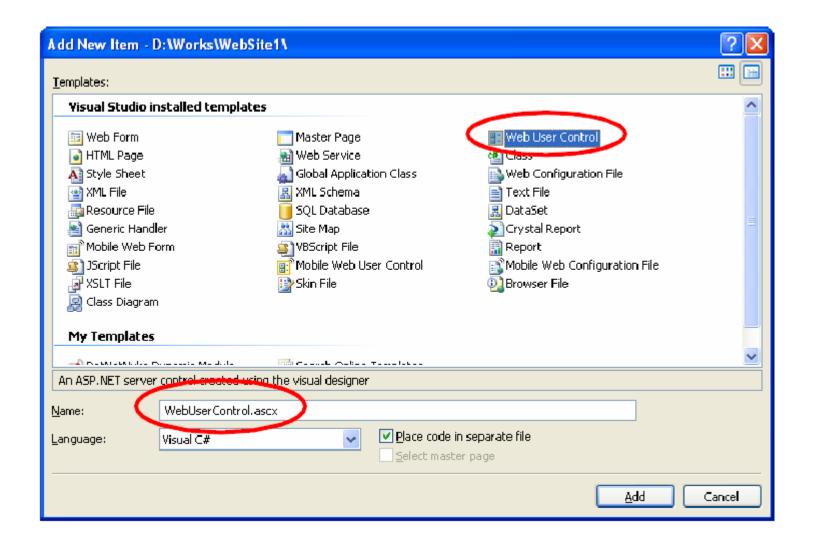
Ví dụ: Tự kiểm tra trên Client

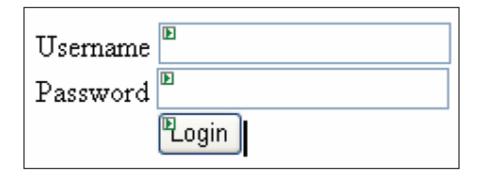


```
<script language="javascript">
    function ClientValidate(e, args)
    {
        args.lsValid = false;
        if (args.Value == "abc")
        args.lsValid = true;
    }
    </script>
```

Web UserControl

- Là các trang ASP.NET được sử dụng như là server control
- Có thể tự định nghĩa các thuộc tính & phương thức riêng
- Có khả năng tái sử dụng cho nhiều trang

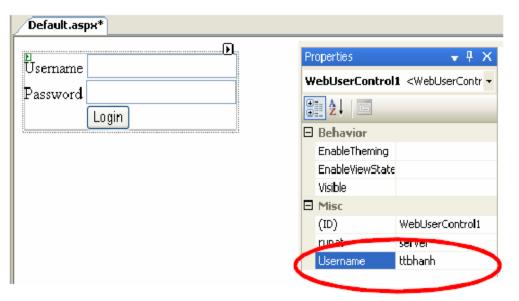




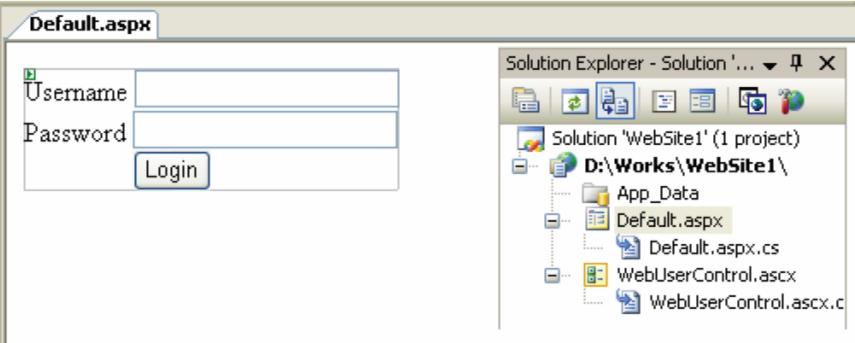
<%@ Control
Language="C#" AutoEventWireup="true"
CodeFile="WebUserControl.ascx.cs"</pre>

Inherits="WebUserControl" %>

```
public string Username {
    get {
       return txtUsername.Text;
    }
    set {
       txtUsername.Text = value;
    }
}
```



Sử dụng Web User Control



<@ Register Src="WebUserControl.ascx"</p>
TagName="WebUserControl" TagPrefix="uc1" %>

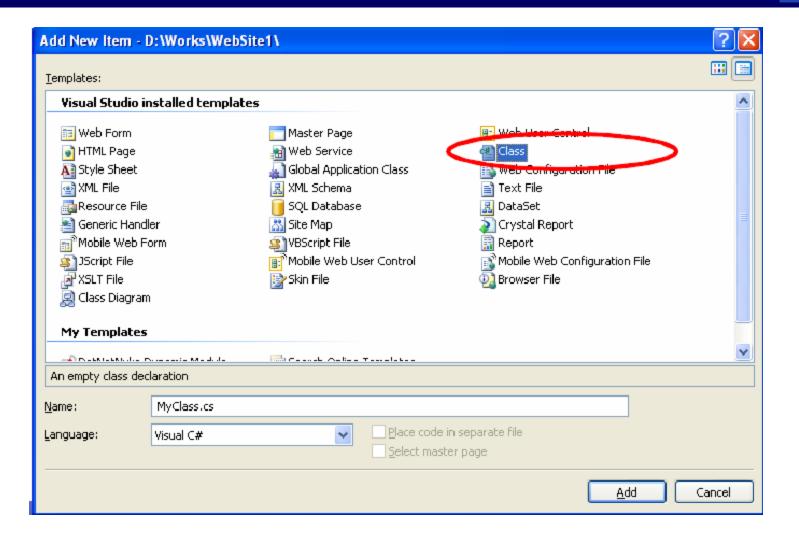
<uc1:WebUserControl ID="WebUserControl1" runat="server"/>

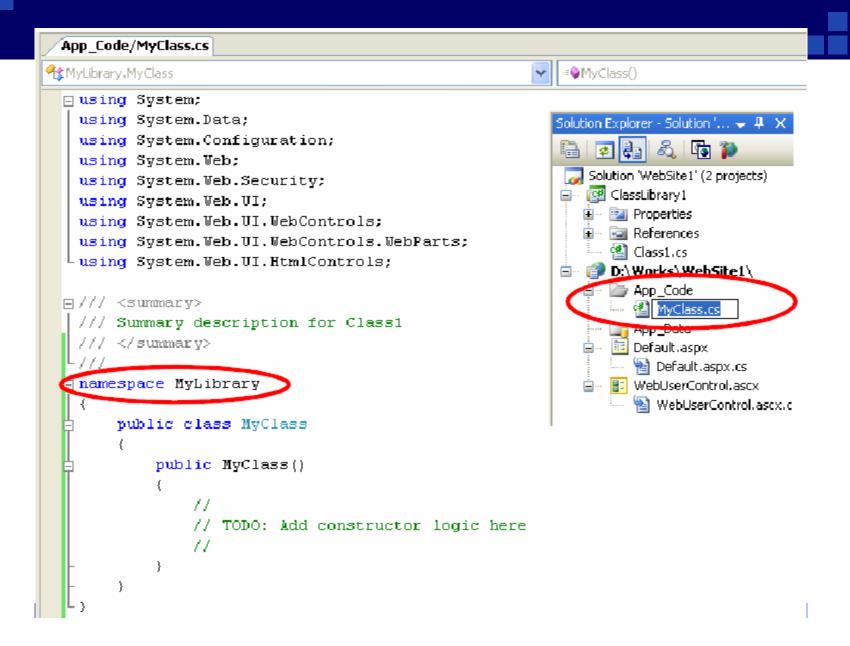
Component

- 1 Component cơ bản bao gồm 1 hoặc nhiều Class
- Được sử dụng như các lớp thư viện tự tạo
- Hỗ trợ đa ngôn ngữ
- Được biên dịch thành tập tin .DLL
- Có khả năng tái sử dụng cao trong nhiều ứng dụng khác nhau



Cách 1

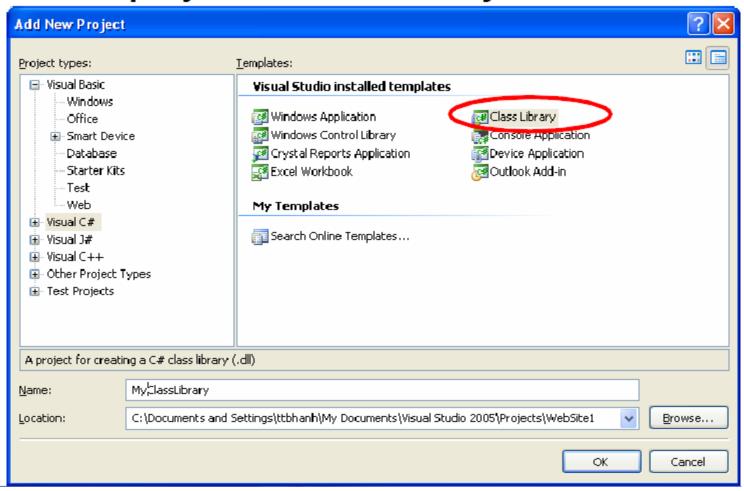


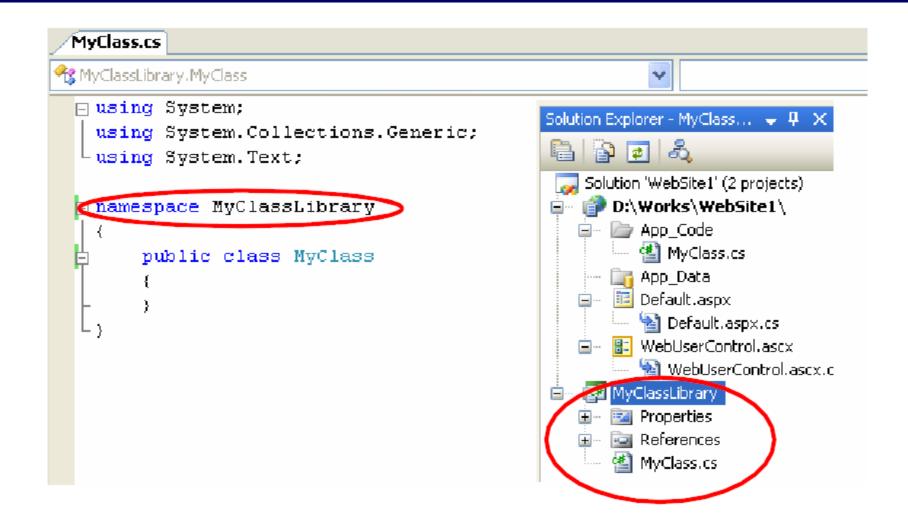


```
namespace MyLibrary
                                                  blic class MyClass
                                                     public MyClass()
using MyLibrary;
public partial class _Default : System.Web.UI.Pag
  protected void Page_Load(object sender, EventArgs e)
    MyClass x = new MyClass();
```

Cách 2

Add new project > Class Library





Add references > Projects

MyClassLibrary.DLL

```
namespace MyClassLibrary
{
    public class MyClass
    {
        }
     }
```

using MyClassLibrary;

```
public partial class _Default : System.Web_Lf.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        MyClass x = new MyClass();
    }
}
```

Các đối tượng của ASP.NET

- Response
- Request
- Server
- Application
- Session

Response Object

- Response là đối tượng được tạo ra tương ứng với mỗi yêu cầu của client
- Đối tượng Response thông thường dùng để xử lý các HTTP Request, và trả kết quả về cho client

Thuộc tính	Mô tả
Charset	xác lập bộ charset sẽ truyền
IsClientConnected	cho biết hiện thời client có đang kết nối hay không
Cookies	các thông tin về Cookies sẽ được ghi xuống client

Phương thức	Mô tả
Write	ghi dữ liệu text
WriteFile	ghi dữ liệu từ file
BinaryWrite	ghi dữ liệu nhị phân
Close	đóng kết nối tới client
End	Kết thúc thi hành một trang
Redirect	chuyển client sang một URL khác

```
protected void btnClick_Click(object sender, EventArgs e)
{
    Response.Redirect(txtLink.Text);
}
```

Request Object

- Cung cấp các thông tin về yêu cầu của client
- Được tạo ra tương ứng với các Http Request
- Dùng đối tượng này để đọc những thông tin client đã gởi (header, query string, cookies, ...)

Thuộc tính	Mô tả
Browser	Cung cấp thông tin liên quan đến trình duyệt của client
Url	Địa chỉ trang gửi request
Cookies	các thông tin về Cookies ở client sẽ được đọc lên
QueryString	Lấy tham số truyền từ client theo phương thức GET
Form	Lấy tham số truyền từ client theo phương thức POST

Ví dụ - Truyền tham số theo phương thức GET

- Cách tạo HTTP QueryString
 - Gõ trực tiếp vào chuỗi URL

 string sample

Cookies

- Cookies là một mẫu thông tin nhỏ dùng để lưu trữ thông tin của người dùng trên máy tính.
- Cookies có thể được lưu trữ tạm thời hoặc lưu trữ lâu dài.
- Cookie lưu trữ tạm thời : sẽ không còn giá trị ngay khi người dùng rời khỏi web site.
- Cookie lưu trữ lâu dài : vẫn còn được lưu trữ trên máy của người dùng,và Web Server có thể đọc những thông tin này vào những lần kế tiếp người dùng vào web site.

- Thể hiện trong ASP.NET là lớp HttpCookie
- Các thuộc tính của HttpCookie
 - Name : tên của Cookie
 - Domain : domain cookie này thuộc về.
 - Expires : xác định thời gian có hiệu lực của Cookie
 - Value : Giá trị của Cookie
 - HasKeys : Cookie có tập giá trị con hay không
 - Values : tập các giá trị của Cookie

Sử dụng cookies

Lệnh ghi cookies

- Response.Cookies[cookie][(key)|.attribute] = value;
 - cookie: tên biến
 - key : tham số tùy chọn, có thể đặt nhiều giá trị cho 1 cookie
 - attribute: thuộc tính (domain, path,....)

Lấy giá trị cookies

– value = Request.Cookies[cookie][(key)|.attribute]

Cookies don

Ghi

```
Response.Cookies["userName"].Value = "mike";

Response.Cookies["userName"].Expires = DateTime.Now.AddDays(1);

- Đọc

if (Request.Cookies["userName"] != null)

Label1.Text = Request.Cookies["userName"].Value;
```

Cookies có các subkey

Ghi

```
Response.Cookies["userInfo"]["userName"] = "mike";
Response.Cookies["userInfo"]["lastVisit"] = DateTime.Now.ToString();
Response.Cookies["userInfo"].Expires = DateTime.Now.AddDays(1);
— Đọc
if (Request.Cookies["userInfo"] != null)
{
    Label1.Text = Request.Cookies["userInfo"]["userName"];
    Label2.text = Request.Cookies["userInfo"]["lastVisit"];
}
```

Xóa Cookies

Response.Cookies["userName"].Expires = DateTime.Now.AddDays(-1);

Response.Cookies["userInfo"].Expires = DateTime.Now.AddDays(-1);

Server Object

Cung cấp các phương thức giúp

- Chuyển điều khiển giữa các trang với nhau
- Lấy các thông tin về mã lỗi, encode, ...

Các thuộc tính

- MachineName : tên server
- ScriptTimeout : thời gian time-out của request

Phương thức	Mô tả
MapPath	Ánh xạ địa chỉ tương đối thành địa chỉ tuyệt đối trên server
HtmlEncode	Giữ nguyên tag HTML
HtmlDecode	Định dạng nội dung theo tag HTML
UrlEncode	Mã hóa Url theo ASCII
UrlDecode	Giải mã ASCII từ Url
Transfer	Lấy kết xuất từ trang khác rồi quay lại
Execute	Lấy kết xuất từ trang khác

Application Object

- Một ASP.NET application bao gồm tất cả các file, trang web, sự kiện, module và code trong phạm vi một thư mục web ảo (virtual directory) và các thư mục con của nó
- Đối với mỗi ASP.NET application, một Application
 Object được tạo ra để thể hiện tình trạng của
 ASP.NET application này
- Application Object được tạo khi client yêu cầu bất kỳ trang nào trong application này

Application Object bị hủy khi

- Stop Web Server
- Server bị sập
- Huy ASP.NET Application

Sự kiện

- Application_OnStart
- Application_OnEnd

(global.asax)

- Application Object chứa một danh sách các biến trạng thái dùng chung của application. Ta có thể sử dụng chúng để lưu trữ các thông tin xuyên suốt ứng dụng
- Các biến trạng thái này được lưu trữ thành từng cặp key-value.
 - Key : tên trạng thái
 - Value : giá trị trạng thái

```
Application["SoLan"] = 0;
```

```
//global.asax
<script language="C#" runat="server">
 void Application_OnStart(Object sender, EventArgs E) {
       Application ["SoLan"] = 0;
</script>
//Myform.aspx.cs
void Page_Load(Object Src, EventArgs E){
 Application["SoLan"] = (Int32) Application ["SoLan"] + 1;
 Response.Write("Số lần vào trang này:" + Application ["SoLan"]);
```

Ví dụ – Đếm số lần duyệt 1 trang web (Cải tiến)

```
void Page_Load(Object Src, EventArgs E){
   Application.Lock();
   Application["SoLan"] = (Int32) Application ["SoLan"] + 1;
   Application.UnLock();
   Response.Write("Số lần vào trang này :" + Application ["SoLan"]);
}
```

Session Object

- Một session (phiên làm việc) là một chuỗi các thao tác của người dùng trên cùng một web application
- Với mỗi phiên làm việc của client, sẽ có một Session Object được tạo ra
- Session Object sẽ cung cấp cho ta những thông tin về phiên làm việc hiện hành này của client

Session được tạo ra khi client bắt đầu phiên làm việc của mình

- Session sẽ được hủy khi
 - Client tường minh thoát khỏi session
 - Sau một khoảng thời gian (time-out), client không có hành động làm việc nào
- Sự kiện
 - Session OnStart
 - Session_OnEnd(global.asax)

	Ý nghĩa
Session.Timeout = minutes	Đ ặ t th ờ i gian c ủ a phiên làm vi ệ c
Session.SessionID	ID của phiên lầm việc
Session.Abandon()	Hủy phiên lầm việc (hủy cả biến dữ liệu)

- Cung cấp cho ta một danh sách các trạng thái xuyên suốt session này
- Các trạng thái của Session Object cũng được tổ chức trong một Collections dưới dạng key-value

Session["DaDangNhap"] = 0;

Ví dụ - Kiểm tra Quyền đăng nhập

 Làm thế nào để ngăn không cho người dùng truy cập vào các trang web nếu chưa đăng nhập?

Ý tưởng

 Dùng các biến Session để lưu trạng thái đăng nhập của người dùng:

```
Session["IsLogin"] = 0/1 : Lưu trạng thái đăng nhập
```

Session["Username"] : Lưu Tên đăng nhập

Session["Authentication"]: Lưu Loại quyền đăng nhập

. . .

 Khởi tạo giá trị mặc định cho biến Session["IsLogin"] = 0 (chưa đăng nhập)

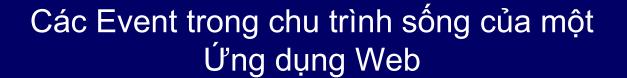
Tạo trang Login.aspx cho phép người dùng đăng nhập

- Nếu kiểm tra thông tin đăng nhập không đúng thì hiển thị thông báo yêu cầu đăng nhập lại.
- Ngược lại, nếu ĐÚNG thì dùng một (hoặc nhiều) biến Session để lưu trạng thái login thành công lại.

```
//Login.aspx.cs
void btnLogin_Click(Object Src, EventArgs E){
    if (Thong tin dang nhap dung)
        Session ["IsLogin"] = 1;
    else
        Response.Write("Vui lòng nhập lại!");
}
```

 Trong tất cả các trang muốn bảo mật, phải thêm đoạn mã sau để kiểm tra người dùng đã login hay chưa, nếu chưa thì redirect lại trang login.aspx

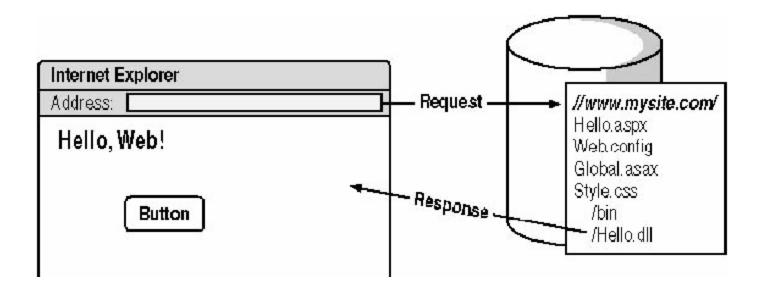
- 4. Tạo xử lý khi người dùng logout
 - Reset trạng thái login là chưa đăng nhập





Các Tập tin trong một Ứng dụng Web

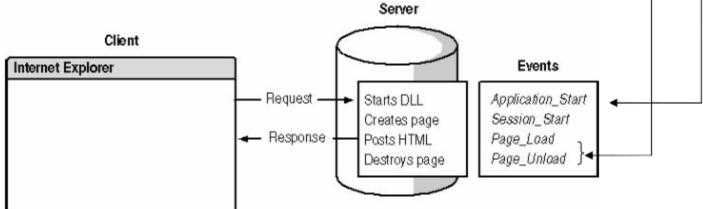
- Khi xây dựng một Ứng dụng Web:
 - Visual Studio .NET biên dịch tất cả mã nguồn vào một file .DLL lưu trong thư mục /bin
 - Phần giao diện của ứng dụng nằm ở các file .aspx và .html



Chu trình sống của một ứng dụng web

- Bắt đầu khi một trình duyệt yêu cầu 1 trang web từ ứng dụng, gọi là Session
- Ứng dụng web vẫn chạy nếu như nó vẫn còn Session đang hoạt động

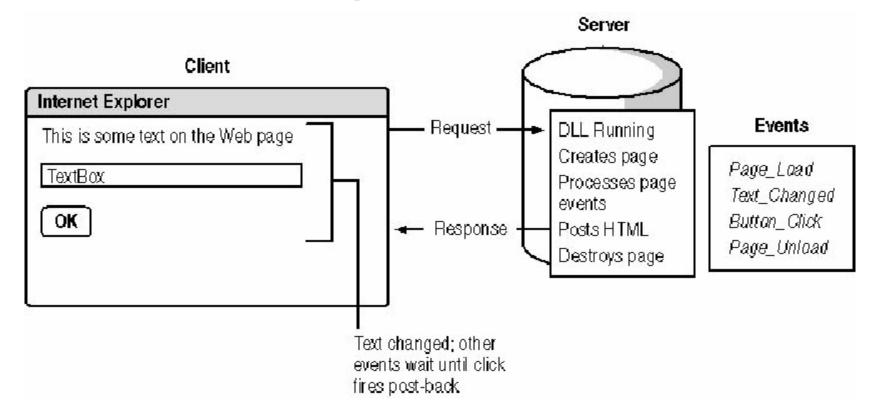
 Chu trình sống của 1 Web Form chỉ tồn tại trong 1 khoảng thời gian ngắn



Chu trình sống tiếp tục

- Người dùng tương tác với giao diện web (gõ vào text box, đánh dấu chọn các check box...) cho đến khi kích hoạt một sự kiện post-back (nhấn button ...)
- Dữ liệu của trang (view state) được gửi về cho server
- Khi server nhận được view state
 - ➤ Nó **tạo** ra thể hiện mới của Web Form
 - ➤ Điền dữ liệu vào view state
 - ➤ Xử lý các sự kiện xảy ra
 - ➤ Trả kết quả HTML về cho trình duyệt và hủy thể hiện của Web Form

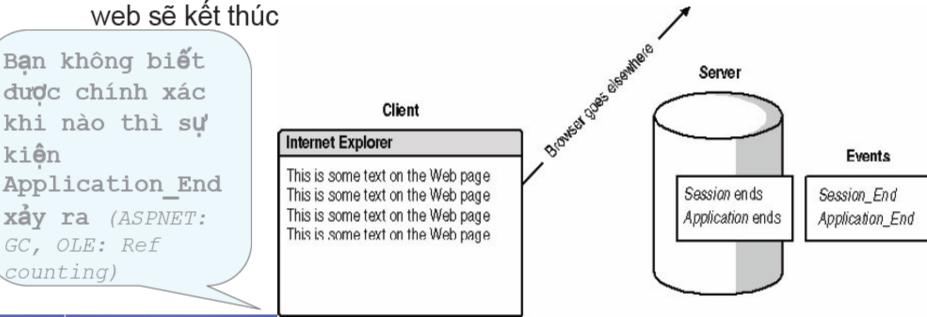
Chu trình sống tiếp tục



Chu trình sống kết thúc

 Khi người dùng tắt trình duyệt, hoặc sau một khoảng thời gian không refesh lại trang web thì Session của người dùng kết thúc

Nếu không còn Session nào từ người dùng thì ứng dụng





Hot Tip

- How do I incorporate my logo to a slide that will apply to all the other slides?
 - On the [View] menu, point to [Master], and then click [Slide Master] or [Notes Master]. Change images to the one you like, then it will apply to all the other slides.

Title

Add your text

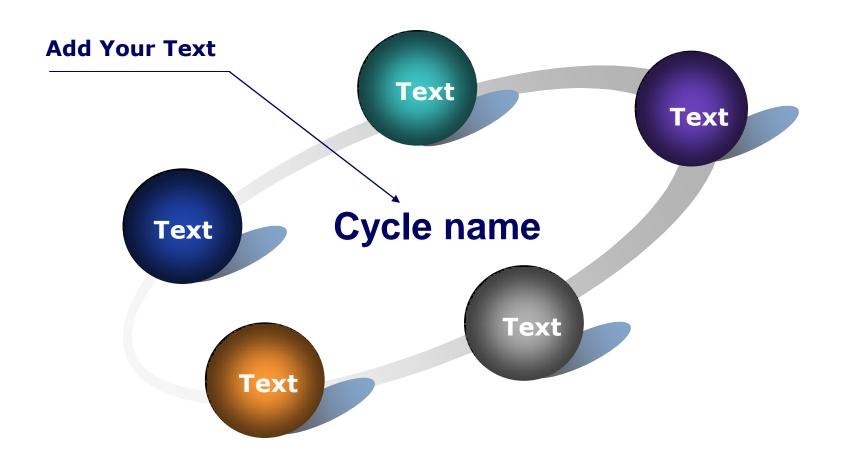
ThemeGallery

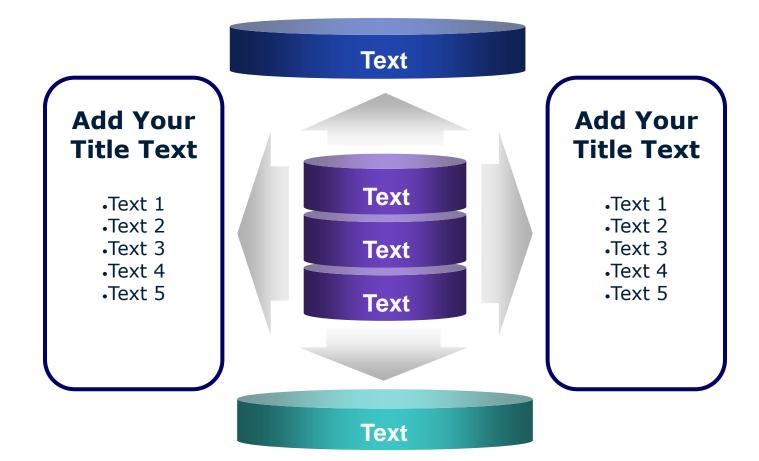
is a Design Digital Content & Contents mall developed by Guild Design Inc.

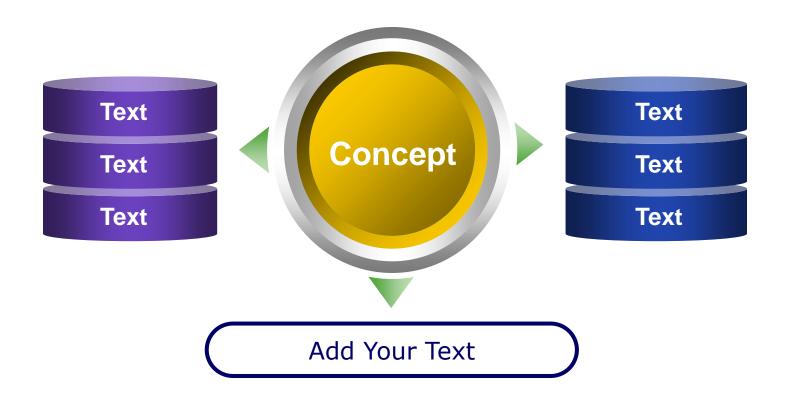
ThemeGallery

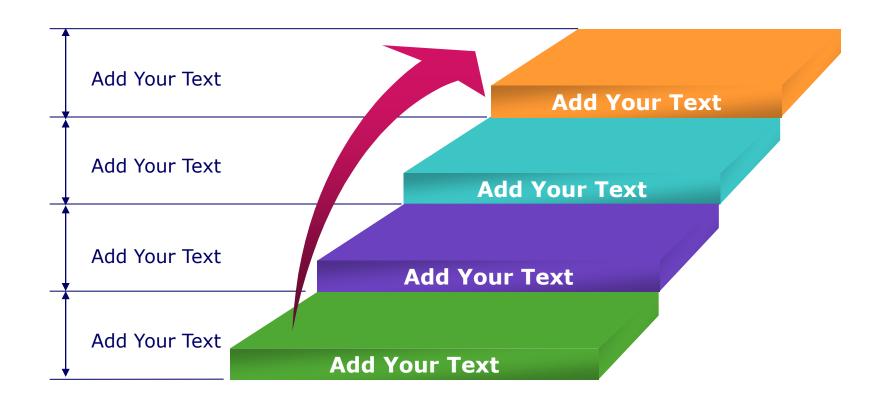
is a Design Digital Content & Contents mall developed by Guild Design Inc.

Cycle Diagram







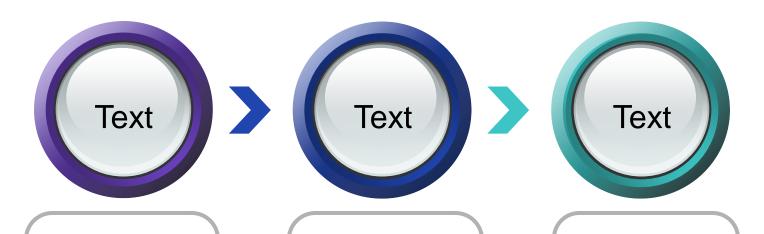


Add Your Text

Add Your Text

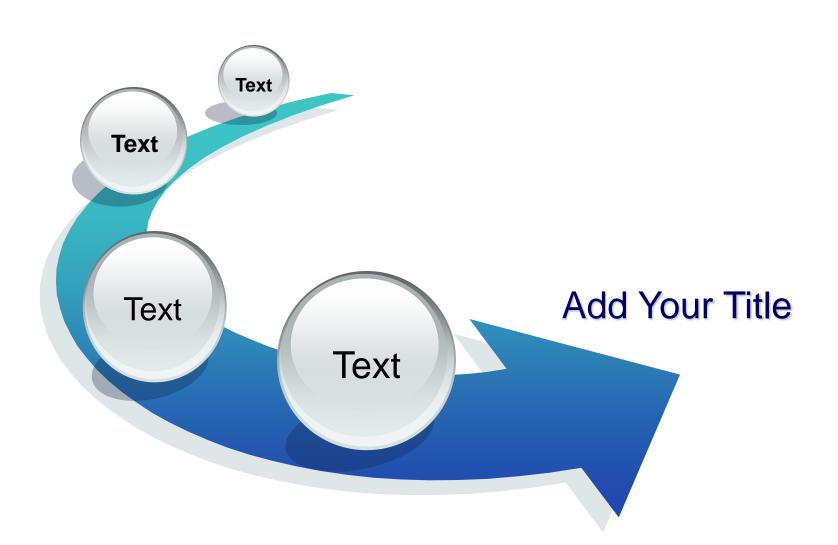
Add Your Text

Add Your Title

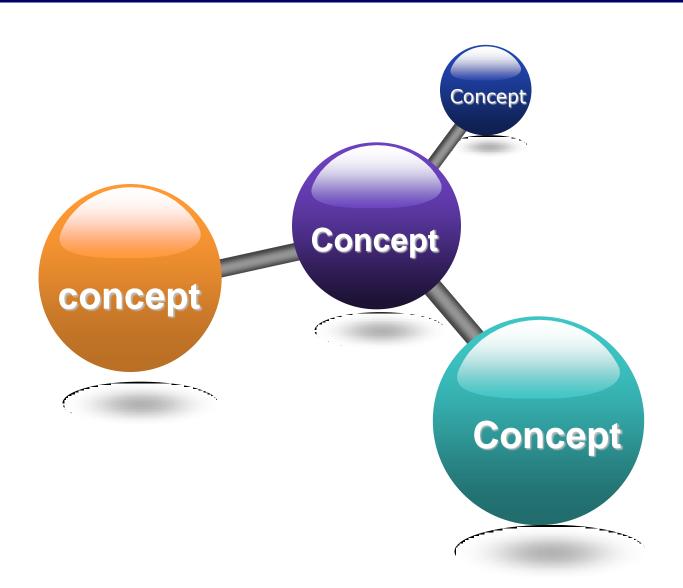


ThemeGallery is a Design Digital Content & Contents mall developed by Guild Design Inc. ThemeGallery is a Design Digital Content & Contents mall developed by Guild Design Inc.

ThemeGallery is a Design Digital Content & Contents mall developed by Guild Design Inc.



Marketing Diagram



1

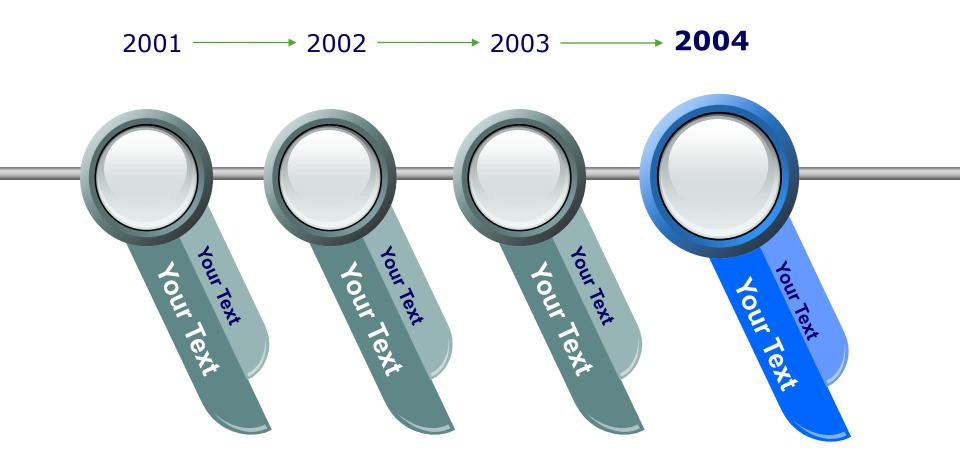
ThemeGallery is a Design Digital Content & Contents mall developed by Guild Design Inc.

2

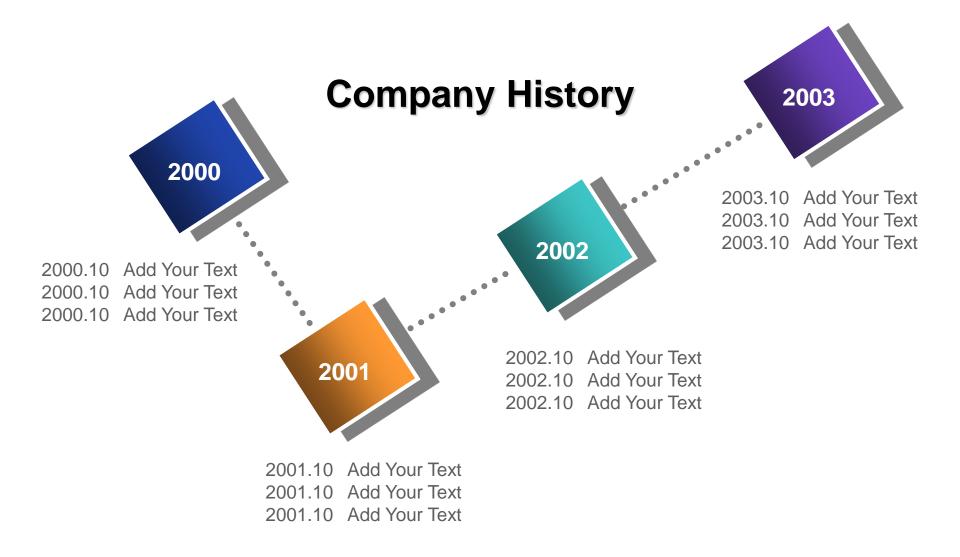
ThemeGallery is a Design Digital Content & Contents mall developed by Guild Design Inc.

3

ThemeGallery is a Design Digital Content & Contents mall developed by Guild Design Inc.



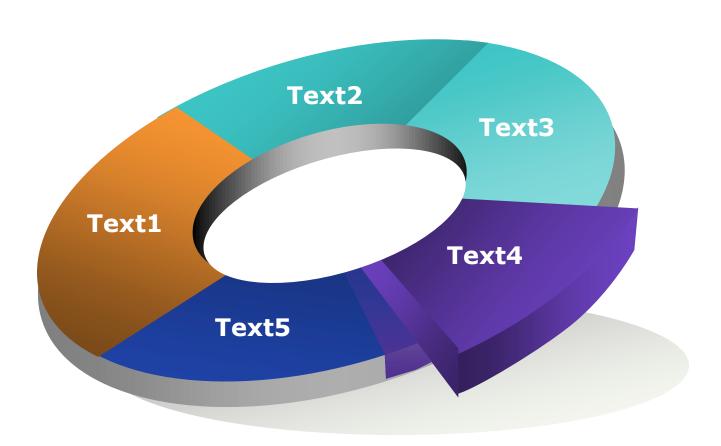
Add Your Title **ThemeGallery** is a Add Your Title **Design Digital** Add Your Title **ThemeGallery** is a Content & **Design Digital** Contents mall ThemeGallery is a Content & developed by **Design Digital** Contents mall Guild Design Inc. Content & developed by Contents mall Guild Design Inc. developed by Guild Design Inc.



Table

	Title	Title	Title	Title	Title
Title	0	0	0	0	0
Title	0	0	0	0	0
Title	0	0	0	0	0
Title	0	0	0	0	0
Title	0	0	0	0	0
Title	0	Х	0	Х	0

3-D Pie Chart



Marketing Diagram

