

Dezvoltarea Aplicatiilor Web-Anul 3

Laborator 1

Exercitiul 1

Page is not postback! HTML controls: <input type="text"/> <input type="button" value="Say Hello!"/>	Page is postback! HTML controls: <input type="text" value="Ana"/> <input type="button" value="Say Hello!"/> Hello Ana!
.NET Controls <input type="text"/> <input type="button" value="Say Hello!"/>	.NET Controls <input type="text" value="Andrei"/> <input type="button" value="Say Hello!"/> Hello, Andrei!

Partea de aspx:

1. Folosim HTML Controls

- folosim **input** pentru casuta de text in care dorim sa scriem si pentru buton si **label** pentru textul pe care dorim sa il afisam

2. Folosim .NET Controls

- in .NET inputul pentru casuta de text este **TextBox**, iar inputul pentru buton este **Button**. Pentru afisarea textului, in loc de label se foloseste **Literal**.

OBS!

- **Se poate observa ca in .NET controalele se scriu cu prima litera mare, la fel si denumirile proprietatilor.**
- **OnClick vs onclick** : OnClick se foloseste pentru controale ASP.NET. Acesta executa o metoda pe server si codul sau aferent. OnServerClick se foloseste pentru elemente de tip INPUT din

HTML (care nu sunt controale ASP.NET) pentru a executa cod pe server. Altfel, pe elementele de tip HTML, daca se foloseste **onclick**, acesta va executa codul aferent de javascript dat ca parametru.

Partea de CS:

Dupa cum se observa dorim ca atunci cand intram pentru prima data pe pagina sa fie afisat mesajul "**Page is not postback!**", iar dupa scrierea textului in casutele de text corespunzatoare, mesajul sa devina "**Page is postback!**"

OBS!

Page_Load – metoda care se executa in ciclul de viata al unei pagini, imediat dupa initializarea unei pagini Web. Aceasta este folosita fie pentru a initializa valorile default in pagina, fie pentru a face procesari inainte de afisarea paginii.

Page.IsPostBack – ne ajuta sa verificam daca a avut loc un postback pe pagina sau nu

De asemenea, valoarea se schimba utilizand id-ul elemntului, de aceea denumirile pe care le da id-urilor trebuie sa fie sugestive.

EX: daca id-ul literalului este **PageLiteral**, textul se schimba astfel:

PageLiteral.Text = "Page is postback!"

Pentru a prelua valoarea scrisa in casuta de text se foloseste si de aceasta data id-ul pe care l-am dat respectivului input, fiind si aici o mica diferenta intre HTML si .NET

Pentru HTML – folosim **.Value**

Pentru .NET – folosim **.Text**

Exemplu pentru partea de HTML:

```
protected void HtmlButtonClick(object sender, EventArgs e)
{
    HtmlHelloText.InnerText = "Hello " + HtmlTextBox.Value + "!";
    HtmlHelloText.Visible = true;
}
```

 **Implementati si pentru partea de .NET, rezolvand exercitiul.**

Exercitiul 2

Guess the number between 1 and 100?

Guess the number between 1 and 100?

Hidden number is lower

Guess the number between 1 and 100?

Hidden number is higher

Guess the number between 1 and 100?

You guessed right!

Indicatii:

- o sa folosim controale .NET pentru elementele de care avem nevoie
- butonul Try again nu o sa fie vizibil de la inceput. Acesta va fi vizibil doar in momentul in care este ghicit numarul corect
- o sa folosim si validare pentru a ne asigura ca numarul introdus este de tip intreg – o sa folosim proprietatile validatorului

Partea de CS:

1. Avem nevoie de o metoda cu ajutorul careia sa generam un numar random. Pentru acest caz putem salva numarul la nivel de sesiune, folosind un vector cheie-valoare. Se foloseste clasa Random din namespace-ul System.

```
Random r = new Random();  
Session["hiddenNumber"] = r.Next(100) + 1;
```

2. Cand trebuie generat acest numar random?
3. Functionalitatea care are loc in momentul apasarii butonului.

Atentie la tipurile de date!

- `int hiddenNumber = Session["hiddenNumber"];` - `hiddenNumber` este numarul generat automat, salvat in sesiune. **Ce trebuie sa avem in vedere in aceasta situatie?**
- `int guessedNumber = TBNumber.Text;` - `guessedNumber` este numarul pe care il scriem in TextBox. **Ce tip are numarul scris in TextBox? Este corecta aceasta scriere?**

La urmatorul pas verificam cele doua numere utilizand structura **if-else**.

De asemenea, putem utiliza **try - catch** avand structura:

```
try{  
  
    } catch (Exception) {  
        // mesajul pe care dorim sa il afisam  
    }
```

4. Functionalitatea care are loc in momentul apasarii butonului Try again.

 **Rezolvati intregul exercitiu!**

BE
POSITIVE
AND
HAVE A
NICE WEEK