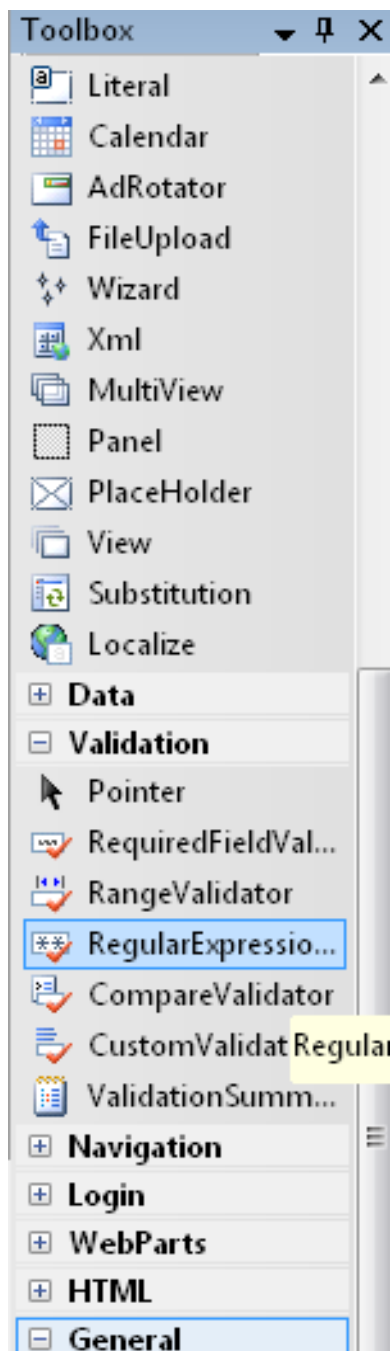


## RegEx'y



Regular Expressions - czyli wyrażenia opisujące wzorec, do którego mają pasować dane. Dzięki nimi sprawdzimy zgodność wpisywanych przez użytkownika strony informacji z intencjami, jakie mieliśmy, podając mu pola formularza ;)

W wyrażeniach regularnych możemy używać zarówno znaków zwykłych, które będą interpretowane dosłownie (na przykład słowo poniedziałek będzie wzorcem dla słowa poniedziałek), ale też znaków specjalnych, których wybór mamy u dołu tej broszurki; dla żądnych wiedzy jest też lista źródeł, z których korzystałam.

**ĆWICZENIE: tworzymy wyrażenie regularne, opisujące e-mail uniwersytecki na serwerze UMK w formacie „twójloign@stud.umk.pl” (z wariacjami):**

Wersja 1 - login tylko w postaci numeru indeksu:

```
([d]{6})@stud\.umk\.pl
```

Nie wpisze więc adresu ktoś, kto zmienił swój domyślny login, którym jest numer indeksu, na inny, na przykład na literowy.

Wersja 2 - z numerem indeksu lub zmienionym loginem:

```
([d\w]*)@stud\.umk\.pl
```

Teraz zaloguje się każdy, ale ciągle można wtedy wpisać @stud.umk.pl i będzie działać! Błąd.

Wersja 3 - w której trzeba coś podać przed @stud.umk.pl:

```
([d\w]+)@stud\.umk\.pl
```

Na deser: Wersja 4, zarówno dla studentów, jak i wykładowców :) (z dodatkową kontrolą początku i końca wyrażenia)

```
^([d\w]+)@(stud\.)?umk\.pl$
```

```
^([d\w]+)@((stud\.)?(fizyka\.)?)umk\.pl$
```

RegularExpressionValidator znajdziemy w części walidacyjnej.

Properties

**RegularExpressionValidator1** System.Web.UI.Wel ▾

Accessibility

AccessKey	
TabIndex	0

Appearance

BackColor	<input type="text"/>
BorderColor	<input type="text"/>
BorderStyle	NotSet
BorderWidth	
CssClass	
Display	Static
ErrorMessage	<b>RegularExpressionValidator</b>

Font

ForeColor	<input type="color" value="Red"/> Red
Text	

Behavior

ControlToValidate	<b>TextBox1</b>
EnableClientScript	True
Enabled	True
EnableTheming	True
EnableViewState	True
SetFocusOnError	False
SkinID	
ToolTip	
ValidationExpressi	<input type="text" value="\w+{[-+.']\w+}@\w+{[-.]\w+..."/>
ValidationGroup	
Visible	True

Data

(Expressions)	
---------------	--

Layout

Height	
Width	

Misc

(ID)	<b>RegularExpressionValidator1</b>
------	------------------------------------

**ValidationExpression**  
Regular expression to determine validity.

**Display** - Static oznacza, że jeśli pojawi się błąd, pojawi się też tekst o nim informujący, a na stronie zostanie zarezerwowane na to miejsce; Dynamic pozwala na to, by miejsce na komunikat kontrolki nie było „rezerwowane”

**ErrorMessage** - to, co wyświetli się, jeśli wprowadzone dane będą niezgodne z wzorcem

**ControlToValidate** - pole, którego zawartość chcemy poddać kontroli

**ValidationExpression** - nasze wyrażenie regularne - wybrane z listy lub wpisane samodzielnie

### Ściaga:

- \* - wystąpienie wielokrotne lub niewystąpienie danego znaku/ciągu
- + - wystąpienie raz lub więcej razy danego znaku/ciągu
- ? - wystąpienie pojedyncze lub - niewystąpienie
- () - nawias działa jak w matematyce
- [] - nawias kwadratowy pozwala podać zakres
- {2} - nawias klamrowy pozwala podać częstość - tu: dwa wystąpienia,
- {2,3} - od dwóch do trzech wystąpień
- {2,} - dwa lub więcej wystąpień
- \w - dowolny znak literowy lub ciąg liter - uwaga na wielkość liter!
- \d - dowolna cyfra lub ich ciąg - uwaga na wielkość liter!
- .
- .\ - kropka (żeby znaki specjalne pozbawić specjalności, dodajemy backslash
- ^ - ciąg rozpocznie się dokładnie tym, co nastąpi po znaku ^
- \$ - ciąg skończy się dokładnie na tym, co jest przed znakiem \$
- [^] - ciąg NIE BĘDZIE tym, co w nawiasie (zaprzeczenie)
- | - alternatywa

### Źródła:

<http://www.if.pw.edu.pl/~ludwik/regex.html>

<http://codeguru.pl/Articles/14413/Comments.aspx>

[http://4programmers.net/PHP/Wyrazenia Regularne \(Regular Expression\)](http://4programmers.net/PHP/Wyrazenia_Regularne_(Regular_Expression))

Paweł Chłosta, „Ćwiczenia z ASP.NET i kolekcje C#”, Wydawnictwo MIKOM, Warszawa 2004, s.160-163.