



# PiLLoW

**Programming in (Constraint)  
Logic Languages on the Web**

# Történet

- Technical University of Madrid
  - Manuel Hermenegildo
  - Daniel Cabeza
- Ciao Prologhoz készült
- Más Prologokhoz is illesztették
  - SICStus
  - SWI
  - YAP
  - ECLiPSe



# Témák

- HTML
- HTML sablonok
- XML
- Formok (CGI)
- HTTP

# Betöltés

- **Ciao Prologban**

```
use_package(pillow).
```

- **SICStus Prologban**

```
use_module(library(pillow)).
```

- **Esetleg külön modulokban**

```
html, http, pillow_types
```

# HTML

## ■ html2terms(String, Terms)

main:-

```
    html2terms(S, [
        env(html, [], [
            env(body, [], cica)
        ])
    ]),
    write_string(S).
```

?- main.

```
<html><body>cica</body></html>
```

yes

?-

# HTML visszafelé is

```
main:-
```

```
    html2terms("<html><body>macko</body></html>", T),  
    write(T).
```

```
?- main.
```

```
[env(html, [], [env(body, [], [macko])])]
```

```
yes
```

```
?-
```

# HTML elemek

- `image(address)`
- `\\`
- `ref(addr, text)`
- `heading(n, text)`
- `itemize([also, masodik])`
- ``
- `<br>`
- `<a href="addr">`  
`text`  
`</a>`
- `<hn> text </hn>`
- `<ul>`  
`<li>also</li>`  
`<li>masodik</li>`  
`</ul>`

# HTML sablonok

- Ha az oldalon vannak változó elemek, mit tegyünk?
- A HTML oldal elkészítése így nehézkes lehet
- Megoldás: `html_template/3`



# html\_template

- `html_template(Chars, Terms, Dict)`
- Html váz
- Változó elemek:
  - `<v>valami</v>`
  - `_`-al kezdődő attribútumok

# html\_template példa

```
?- html_template(`
<html>
<body bgcolor=_szin>
<v>valami</v>
</body>
</html>`,
T,
[szin="white", valami="2003.11.24"]).
```

```
T=[env(html, [], [
    env(body, [bgcolor="white"], [
        "2003.11.24"])
    ])
]
```

# XML

## ■ xml\_terms(String, Terms)

```
xml2terms("<types>  
  <type name=\"egyik\">  
    <size>2</size>  
  </type>  
  <type name=\"masik\">  
    <size>8</size>  
  </type>  
</types>", T).
```

```
T = [env(types, "", [],  
env(type, [name="egyik"], [env(size, [], ["2"]), ""]),  
env(type, [name="masik"], [env(size, [], ["8"]), ""]))  
]]
```

# XML parser sebességteszt

	<b>220k</b>	<b>670k</b>
<b>Visual C + libxml2</b>	67 ms	207 ms
<b>Ciao + Pillow</b>	240 ms	750 ms
<b>SICStus + Pillow</b>	381 ms	1170 ms
<b>Python + PyXML</b>	8616 ms	30692 ms

# CGI alapok

- Futtatható program
- Paraméterek (általában form-ból)
- HTML (vagy más) kimenet
- Content type: text/html\n\n
- Bármilyen nyelven íródhat

# Form eszközök (HTML)

- `start_form(addr, attr)`
- `end_form`
- `checkbox(name, state)`
- `radio(name, value, selected)`
- `input(type, atts)`
- `textinput(name, atts, text)`
- Stb..

# CGI script eszközök

- `cgi_reply` (html-ben)
  - Kiírja a szükséges fejléctet
- `get_form_input(Dict)`
  - Paramétereket Dict-be helyezi
- `get_form_value(Dict, Var, Val)`
  - Dict-ből a Var nevű paraméter értékét Val-ba helyezi (“”, ha nincs ilyen)
- `form_default(Val, Default, NewVal)`
  - Ha Val üres, akkor Default a kimenet
- Stb..

# Cookie alapok

- Website által
  - Beállítható
  - Később lekérdezhető
- Név – érték pár
- Böngésző tárolja
- Pl. megjegyzi, hogy ki vagy, és legközelebb megismer



# Cookie kezelés

- `set_cookie(Name, Value)`
  - Cookie beállítása
- `get_cookies(Dict)`
  - Dict-be helyezi a cookie-kat

# CGI példa: köszöntő

```
<html><body>
<form method="post" action="cgi-bin/hello.pl">
  <input type="text" name="nev">
  <input type="submit" value="Koszonj!">
</form>
</body></html>
```

```
=====
#!/bin/sh
exec /usr/bin/ciao-shell $0 $*
:-use_package(pillow)
main(_):-
  get_form_input(Info),
  get_form_value(Info, nev, Nev),
  output_html([
    cgi_reply, start, title('Koszones'),
    ['Szia ', Nev, '!'],
    end]).
```

# CGI példa: telefonkönyv

```
:- use_package(pillow).
main(_):-
    get_form_input(Input),
    get_form_value(Input, nev, Nev),
    response(Nev, Valasz),
    output_html([
        cgi_reply, start, heading(2, 'Telefonkonyv'),
        Valasz, start_form, 'Nev:', '\\',
        input(text, [name=nev, size=20]),
        end_form, end]).

response(Nev, Valasz):-
    form_empty_value(Nev) -> Valasz = [] ;
    telefonszam(Nev, Szam) ->
        Valasz = [Nev, 'szama: ', Szam] ;
    Valasz = ['Nincs ', Nev, 'a telefonkonyvben'].
```

```
telefonszam ('Peti', '123-4567').
telefonszam ('Balazs', '234-5678').
```

# HTTP

- `fetch_url(Url, Request, Response)`
  - Az URL által mutatott oldalt Request paraméterekkel letölti, és Response-al egyesíti

```
main:-
```

```
    url(URL),  
    url_info(URL, URLStruct),  
    fetch_url(URLStruct, [], Valasz),  
    write(Valasz).
```

```
url('http://www.index.hu/').
```

# Összefoglalás

- Prolog programok webhez illesztéséhez ideális eszköz
- HTML struktúra magában nehézkes lenne
- Template-ek segítségével változó tartalmú oldalak előállítása könnyű
- Minden, amit egy CGI scriptnek tudnia kell, egyszerűen elérhető