

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24



Python Web Server Gateway Interface

[[Alex Morega](http://alexmorega.com)] (<http://grep.ro>)

ROSEdu TechTalks, 2009/11/10


```
1
2 ###          Cum funcționează          ###
3
4
5 * `environ`:
6 + toată informația despre request
7   - variabile CGI
8 + informație de la alte layere (middleware)
9   - chei cu litere mici, namespaced
10
11
12 * `start_response`:
13 + înseamnă "sunt gata să trimit răspunsul"
14 + status code, response headers
15
16
17 * `return`:
18 + iterator cu response body (ex. o listă)
19 + trimite răspunsul în bucăți
20
21
22
23 -----
24 ROSEdu TechTalks, 2009/11/10
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

###

Middleware

###

- * Scopul: modificarea request-ului și/sau response-ului
- * Funcționare: un middleware "îmbracă" aplicația
 - + aplicația este WSGI
 - + din afară, middleware-ul + aplicația sunt o nouă aplicație WSGI
 - + turtles all the way down



UA: user agent (ex. browser) == (HTTP)
PX: HTTP proxy --- (WSGI)
S: origin server
M*: middleware
A: aplicație

Biblioteci

###

* Generic WSGI:

+ wsgiref: ``simple_server`, `validate``

+ WebOb:

- obiecte prietenoase de request, response, erori

- diverse utilitare (decoratori, middleware, escaping, diverse headere)

+ Routes: URL dispatch

+ Werkzeug: asemănător cu WebOb, + URL dispatch

* Framework-uri: Pylons/TurboGears, BFG, Django, Zope3

* Servere: ``wsgiref.simple_server`, `mod_wsgi`, flup, twisted, magnum`

* Middleware: weberror, toscawidgets, deliverance

* Testing: twill, zope.testbrowser

