

Programmazione CGI Facilitata per WEB AS/400

<http://www.easy400.ibm.it>

Giovanni B. Perotti IBM Italia



Per far funzionare il programma di esempio:

<http://www.easy400.ibm.it/cgidevoit/opinion.mbr>

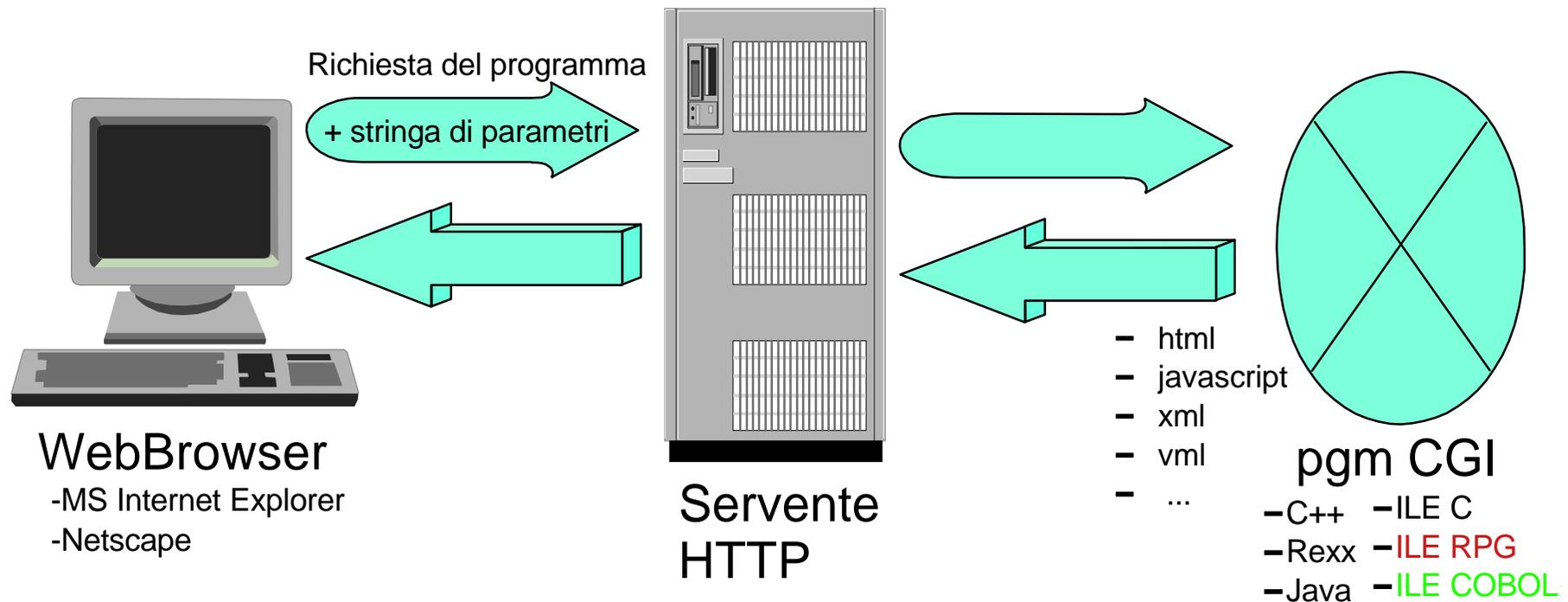
Per leggere le annotazioni dello speaker:

Page-Open speaker note

Che cosa è CGI Common Gateway Interface

supportato da tutte le piattaforme

- Uno **standard** di scambio informazioni tra web browser e programma server
 - forma dell'input
 - forma dell'output
- Programmi **transaction oriented**
 - Una sola richiesta dal browser
 - Una sola risposta dal programma



Programma interattivo tradizionale per 5250

invocato da un programma precedente

```
CALL MYLIBLIB/MYPGM PARM('2')
```

inizializzazione

PUT screen
WAIT
GET screen

?

Fine

processo

Interattivo

Programma CGI per Web

invocato da una pagina html precedente

```
http://.../QSYS.LIB/MYLIB.LIB/  
mypgm.pgm?custno=2
```

GET stringa
parametri

Inizializzazione
&
processo

PUT **html**

no **WAIT**

Fine

Transazionale



Le "debolezze" del CGI

Data la risposta, il CGI *deve finire*:

HTTP non può attendere
Accende LR e fa RETURN

- richiamato, inizia dal principio
- non ha memoria

CGI non persistente

Classico rimedio per la memoria

Adottare la tecnica dello *smemorato*

- ★ mandare un promemoria al client
- ★ prepararsi a riceverlo

CGI persistenti

- RETURN con LR spento
- Tecnica degli scontrini (*handle*)

- richiamato, inizia dal principio
- necessario per COMMIT
- stesse prestazioni
- richiede esperienza

INPUT al CGI

Si tratta di una **stringa** inviata dal browser unitamente alla *invocazione del programma*

```
http://.../path/mypgm.pgm?codcli=000817&namcli=Rossi
```

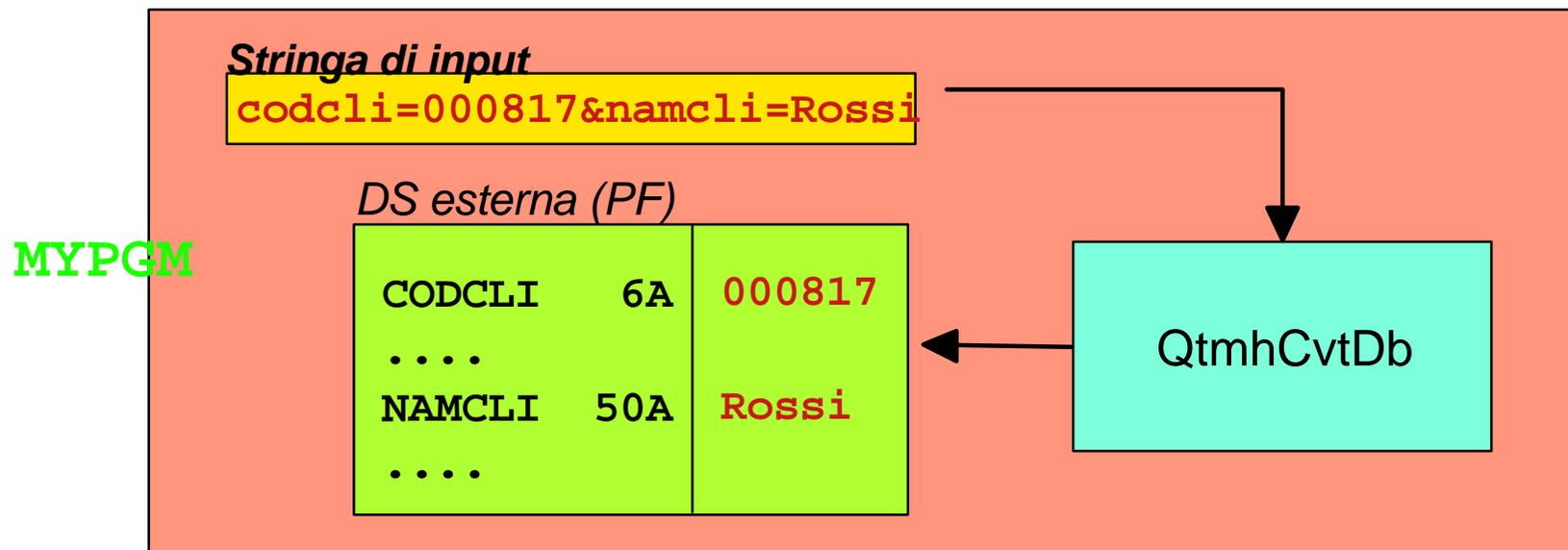
La stringa e' assimilabile ad un elenco di parametri.

Serve perche' il programma capisca la richiesta e possa servirla.

Il programma deve:

1-riceverla tramite apposita **API**, `QtmhRdStdIn` o `QtmhGetEnv`

2-farne il *parsing*, spezzandola in variabili del programma; questo si puo' fare con apposita **API** `QtmhCvtDb` che lavora su una DS esterna



Output dal CGI

- Una stringa, contenente uno script in un linguaggio interpretabile dal Web browser, solitamente HTML
- Spedizione del buffer di output a carico dell'API `QtmhWrStOut`

```
Content-Type: text/html
```

```
<HTML>  
<HEAD>  
<TITLE>Gestione Ordini</TITLE>  
</HEAD>  
<BODY><H1>Conferma Ordine</H1>  
.....  
</BODY></HTML>
```

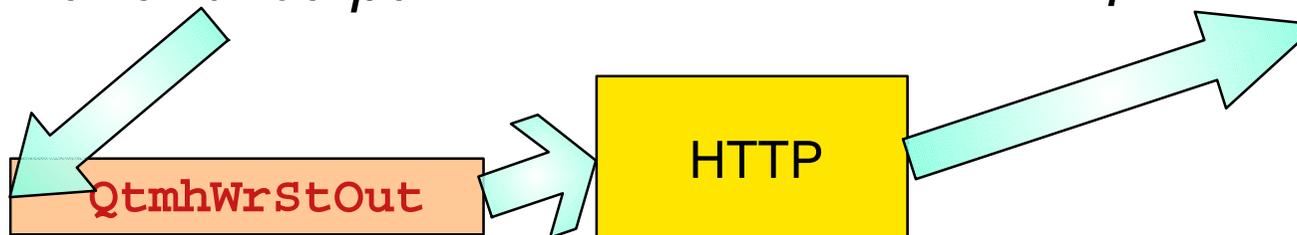
Buffer di output

Gestione Ordini -

Conferma Ordine

.....

Interpretazione del browser



I grossi problemi del CGI

manca un ...Web device file esterno !

- ▶ Input/output gestito non tramite istruzioni (EXFMT) ma tramite API
- ▶ input non preformattato (stringa di input)
- ▶ output non preformattato (stringa di output): gravissimo

Mancata separazione tra logica di processo e presentazione del dialogo conduce a difficoltà di sviluppo e impossibilità pratica di modifica

Troppo difficile da scrivere

Troppo rigido alle modifiche

Troppo costoso

... sono risolti così

un **Service Program** sviluppato nel Lab di Rochester in **RPG ILE**
(sorgenti disponibili)

- ✓ consente definizione esterna degli script (HTML, ...)
- ✓ gestisce gli script come tracciati record
- ✓ gestisce variabili nei tracciati record
- ✓ gestisce tutte le API HTTP

- ★ Gli script si scrivono come testo in membri di file sorgente
- ★ Estremamente piu' facili delle DDS
- ★ Esenti da level-check

Script HTML esterno

```
crtpf mylib/htmlsrc rcdlen(132)
```

```
/$testata
```

```
Content-type: text/html
```

```
<HTML><BODY>
```

```
<HEAD><TITLE>Script HTML  
esterno</TITLE></HEAD>
```

```
/$sezioni
```

<P>Un membro sorgente HTML può essere suddiviso in sezioni (tracciati record), sino ad un massimo di 50. Ogni sezione può essere emessa dal CGI indipendentemente e nella sequenza voluta.

```
/$variabili
```

<p>Ogni sezione può contenere sino ad un massimo di 50 `/%variabili%/`. Ogni variabile può essere caricata dal CGI con un valore lungo sino a 500 caratteri.

```
/$fine
```

```
</BODY></HTML>
```

50

/\$sezioni

per membro

50

/%variabili%/

per sezione

<http://www.easy400/cgi devoit/opinion.mbr>

Una opinione

Mi piacerebbe sapere la tua opinione su

 (1)

Programma CGI EXERCISE

- 👉 Legge l'input
- 👉 Carica l'HTML esterno
- Elabora la richiesta
- 👉 Assegna valori alle `/%variabili%/` nel tracciato record `/$risposta`
- 👉 Spedisce nel buffer di output il tracciato record `/$risposta`
- 👉 Invia il buffer al client
- LR & RETURN

Source file HTMLSRC, membro EXERCISE (2)

```
/$risposta
<HTML>
<h1>E' solo
un'opinione</h1>
Alle /%time%/ la mia
opinione su
/%subject%/ è:<i>
/%questa%/</i><br>
<u>Riprova</u>
```

E' solo un'opinione

Alle 12:56 la mia opinione su questo caffè è: *eccellente*

[Riprova](#)

👉 = operazioni assistite dal service program

Le funzioni del CGI SERVICE PROGRAM

★ Funzioni base

1. Ricevi l'input dal browser client
2. Mappa la stringa nelle variabili del programma
3. Carica l'HTML esterno
4. Valorizza le variabili HTML ed emetti tracciati record (*sezioni*) HTML

★ Funzioni d'uso frequente

1. Assegna i file database
2. Edita campi numerici
3. Ricevi le variabili di ambiente
4. HTML trace & debug

★ Altre funzioni

1. Gestisci i contatori di pagina
2. Genera un numero a caso
3. Esegui un comando CL
4. Crea o modifica una variabile di ambiente
5. Gestisci piu' presenze di una stessa variabile di input

<http://www.easy400.ibm.it>

E' il sito da cui si scarica gratuitamente tutto il materiale necessario e sussidiario

❖ cgi service program

❖ dimostrazioni

❖ didattica, esempi, esercizi

❖ programmi di utilita'

❖ applicazioni

... tutto corredato dai relativi sorgenti



Realizzazione di G.B.
Perotti, IBM Italia