

Llenguatges de Programació

Albert Rubio

Llenguatges de Programació, FIB, UPC

Primavera 2016

Llenguatges de Scripting

- ① Història
- ② Característiques
- ③ Dominis d'aplicació
- ④ Exemples

Continguts

① Història

② Característiques

③ Dominis d'aplicació

④ Exemples

Història

Ancestres:

- “shells”
- processadors de textos o generadors d'informes. Exemples: RPG (IBM, 1959), sed o awk (Unix, 1974 i 1977).

Evolucions:

- Rexx (IBM, 1979) o Perl (1987).
- Altres de propòsit general: Tcl, Python, Ruby, VBScript (Windows) i AppleScript (Mac).

Història

World Wide Web (Anys 90):

- Perl més usat per scripting de servidors web.
- Scripts en Perl van evolucionar a PHP (1995).
- Perl també va influir el desenvolupament de Python (1991). També influït per molts llenguatges de programació.
- Altres competidors de Perl, PHP i Python (codi obert) són JSP (Java Server Pages; per HTML o XML; SUN) i VBScript (Microsoft, 1996)

Continguts

① Història

② Característiques

③ Dominis d'aplicació

④ Exemples

Característiques

Llenguatges de scripting vs llenguatges tradicionals

- Els LS estan pensats per crear aplicacions combinant components.
- Amb LS desenvolupar és de 5 a 10 vegades més ràpid
- Amb LS l'execució és de 10 a 20 vegades més lenta.
- Els LS són normalment interpretats o amb compilació *just-in-time* o les dues.
- Molts LS són dèbilment tipats (no e.g. Python).
- Molts LS tenen tipat dinàmic (no e.g. RPG).

Alguns llenguatges de scripting han evolucionat i poden ser usats com a llenguatges tradicionals (amb - eficiència).

Característiques

Compilació *just-in-time* (JIT):

- Es compila el codi en temps d'execució (a codi màquina o a algun tipus de bytecode)
- Pot aplicar optimitzacions que depenen del context d'execució.
- Positiu: el codi resultant pot ser molt més eficient.
- Negatiu: la compilació en execució és ineficient.
- Es pot aplicar parcialment: només en certes construccions
Per exemple: expressions regulars (matching).
- Pot combinar-se amb la compilació (estàtica) a bytecode:
Primer es compila el llenguatge a bytecode.
S'aplica JIT al bytecode per obtenir codi màquina més eficient.

Característiques

Característiques principals dels llenguatges de scripting:

- Permeten tant ús en batch com interactiu.
 - La majoria tracten l'entrada línia a línia.
Perl és dels pocs que usen un compilador just-in-time que requereix llegir tota l'entrada abans de tractar-la.
 - Altres accepten instruccions per línia de comandes (p.e. REXX, Python, Tcl i Ruby).
Han de poder ser interpretats (sense compilació just-in-time).
Python, Tcl i Ruby admeten les dues possibilitats.

Característiques

- Economia d'expressions.

Afavorir el desenvolupament ràpid i l'ús interactiu.

- Fort ús de símbols de puntuació i identificadors molt curts (Perl).
- “+” paraules en anglès i “-” puntuació (Rexx, Tcl,...)

Eviten les declaracions extenses i les estructures de nivell superior

```
class Hello {  
    public static void main(String[] args){  
        System.out.println("Hello, world");  
    }  
}
```

```
print "Hello, world!\n"
```

Característiques

- Absència de declaracions. Regles simples d'establiment d'àmbit (scoping)

Declaracions:

- En molts llenguatges no hi ha declaracions.
- Per exemple l'assignació dona el tipus.

Àmbit:

- En alguns tot és global per defecte.
- En alguns tot és local per defecte.
- Pot haver regles com que l'assignació defineix la localitat.

Característiques

- Tipat dinàmic flexible

Relacionat amb l'absència de declaracions, la majoria dels LS són tipats dinàmicament.

- En alguns, el tipus d'una variable es comprova just abans de ser usada.
- En altres es pot interpretar diferent en diferents àmbits.

```
$a = "4";  
print $a . 3 . "\n"; # '.' es la concatenació  
print $a + 3 . "\n";  
escriu  
43  
7
```

Característiques

- Fàcil accés a altres programes

Donen moltes opcions predefinides per executar programes o operacions directament sobre el SO.

- Entrada/sortida,
- manipulació de fitxer i directoris,
- manipulació de processos,
- accés a bases de dades,
- sockets (APIs) per comunicacions entre processos
- sincronització i comunicació entre processos,
- Protecció i autorització.
- comunicació en xarxa.

Característiques

- Pattern matching i manipulació de strings sofisticada

És una de les aplicacions més antigues.

- facilita la manipulació de l'input i l'output textual de programes externs.
- Tendeixen a tenir facilitats molt riques per fer pattern matching, cerca i manipulació de strings.
- Normalment és basen en formes esteses de les expressions regulars.

Característiques

- Tipus de dades d'alt nivell
 - S'inclouen com predefinitos tipus d'alt nivell com ara: sets, bags, maps, lists, tuples,...
que no són llibreries, sinó part del llenguatge.
 - Per exemple és habitual tenir arrays indexats per strings com part del llenguatge (que s'implementen amb taules de hash).
 - S'utilitzen “garbage collectors” per gestionar l'espai.

Continguts

① Història

② Característiques

③ Dominis d'aplicació

④ Exemples

Dominis d'aplicació

- Llenguatges de comandes shell
JCL (IBM), csh, tcsh, ksh, bash,...
- Us interactiu.
- Processament batch (no molt sofisticats).
- manipulació de noms de fitxers, arguments i comandes.
- enganxar (*glue*) diversos programes.

```
for fig in *.eps
do
    ps2pdf $fig
done
```

Dominis d'aplicació

- Processament de textos i generació d'informes
sed, awk, Perl
 - Els LS estan fortament orientats al tractament de strings.
 - Les comandes són strings que es divideixen en paraules.
 - Els valors de les variables són strings.
 - se'ns permet extreure substrings.
 - concatenar i moltes més opcions...

Els exemples amb awk il·lustren aquestes possibilitats.

Dominis d'aplicació

- Matemàtiques i estadístiques
 - Llenguatges com Maple, Mathematica i Matlab.
 - gran suport pels mètodes numèrics
 - manipulació simbòlica de formules
 - visualització de dades
 - modelat matemàtic.

orientats a aplicacions científiques i en l'enginyeria

- Llenguatges com S i R per computació estadística
 - Inclou arrays i llistes multidimensionals
 - funcions de primera classe
 - laziness (call-by-need)
 - operacions de selecció (slice) sobre arrays
 - extensió il·limitada (noms, objectes,...)

Dominis d'aplicació

Altres aplicacions (les veurem a les presentacions)

- Scripting de propòsit general: Perl, Tcl, Python, Ruby
- Llenguatges d'extensió

Són llenguatges de scripting que permeten fer scripts per una determinada aplicació. Exemples:

- Per gràfics d'Adobe (p.e. Photoshop) es poden fer scripts JavaScript, Visual Basic (a Windows) o AppleScript (a un Mac).
- Per GIMP es poden fer scripts en Scheme, Tcl, Python i Perl.
- Per emacs hi ha un dialecte de Lisp anomenat Emacs Lisp (veure exemple)

Dominis d'aplicació

- Web scripting: Perl, PHP, Python, JavaScript, Cold Fusion, ...
- XSL (extensible stylesheet language)
 - De HTML a XML, per distingir els continguts i l'aparença
 - Els documents XML són ben format (`<q>...</q>`)
 - XSL: família de llenguatges per especificar que fer amb l'XML.
 - XSL: XSLT , XPath (accedir a parts) i XSL-FO (format visual, a PDF)
 - XSLT (XML Transformations) permet seleccionar, reorganitzar i modificar *tags* i els elements que delimiten.
Permet fer scripts per processar les dades representades en XML. Per exemple, de XML a HTML.

Continguts

① Història

② Característiques

③ Dominis d'aplicació

④ Exemples

Exemples

- Shell
- Awk
- XML
- ...