Alapozás

- Scriptnyelvek jellemzői és összevetése statikus típusrendszerű nyelvekkel
- Példák scriptnyelvekre
- Adatszerkezetek
- Reguláris kifejezések alapjai

Scriptek, scriptnyelvek

- Mit nevezhetünk scriptnek?
  - Egy egyszerű szövegfile-t, melynek tartalma egy program, mely valamely értelmezett nyelven íródott, a program terjesztése a forrás terjesztését jelenti.
- Sok értelmezett nyelvet szokás scriptnyelvnek nevezni.
- Akár funkcionális nyelveket is, mint pl. Scheme, LISP.
- Mi imperatív nyelvekkel foglalkozunk.

Miért érdemes használni?

- gyors alkalmazásfejlesztés
- kritikus részek megírhatók C/C++ nyelven
- rövidebb és érthetőbb kód
- élmény a programozás :-)

Dinamikus és statikus típusrendszerek

- értelmezett nyelvek
- korlátozott deklaráció
- gyors indulás
- nem, vagy gyengén típusos nyelvek
- gyorsabb alkalmazásfejlesztés
- automatikus memóriakezelés

- fordított nyelvek
- deklaráció fontos
- lassú fordítás
- általában erősen típusos nyelvek
- biztonságosság a típusellenőrzés által
- manuális vagy automatikus memóriakezelés

Dinamikus és statikus típusrendszerek

- alapfilozófia: bízzunk a programozóban, tudja, hogy mit akar
- az erőforrások jelentősége kisebb (memória, CPU-idő)
- Perl, Python, Ruby, stb.

- alapfilozófia: a programozó hibát hibára halmozhat
- esetleg fontos az optimális erőforráshasználat
- C++, Ada, Java, Eiffel, stb.

hello_name.pl

wc.pl

Perl

- Larry Wall 1987-ben adta ki az első verziót
- Nagy felhasználói és fejlesztői tábora van
- Rengeteg modul egy központi szerveren
- Flexibilis nyelv
- Fejlődésének története nyomonkövethető a nyelv tulajdonságain
- TIMTOWTDI (There is more than one way to do it)
- Érdekes az objektum-orientáltság a nyelvben

comment_logging.rb

Ruby

- Japán eredetű objektum-orientált scriptnyelv
- Megfontoltan tervezett nyelv
- Érdekes nyelvi konstrukciók
- TIMTOWTDI
- Perl-programozók találhatnak bizonyos hasonlóságokat
- Következetesen objektum-orientált

Dog.py

game.py

Python

- Nagyon elterjedt és kedvelt nyelv előnyös tulajdonságokkal
- Objektum-orientált
- Egyszerű, TIOOWTDI
- Tisztább nyelv, mint a Perl
- Érdekessége, hogy a blokkokat nem kell semmivel elhatárolni ({}, begin..end), hanem egy blokkot az indentáció mélysége határoz meg

example.php

PHP

- PHP: Hypertext Preprocessor
- nagyon elterjedt webes programozási nyelv
- Perl-hez nagyon hasonló
- HTML és forráskód keverése

example.js

JavaScript

- böngészőkben való turkálásra használható
- platform: a böngésző
- nehéz cross-platform kódot írni vele

Icon, Unicon

- Célvezérelt kifejezések
- Nincsenek utasítások, minden kifejezés
- Egy kifejezés visszaadhat 0 vagy több eredményt
  - every e1 do e2;
  - pl. másodfokú egyenlet valós megoldásai (0,1,2)
- Unicon: az Icon objektum-orientált kiterjesztése

TCL

- Tool Command Language
- John Ousterhout fejlesztette ki
- Népszerű, egyszerű fogalmakból építkező nyelv
- Tcl/Tk: egy grafikus könyvtár, amelyhez nagyon sok nyelvben van valamiféle wrapper
- Különös szintaxis
  - command arg1 arg2 arg3

TCL példák

- set hello “Hello World!”
- set sum [expr 1 + 2]
- if {$sum < 10} {
  puts “sum too low”
  }
- for {set i 0} {$i < 5} {incr i} {
  puts $i
  }

Egyéb nyelvek

- REXX: nagygépes scriptnyelv
  - http://www-306.ibm.com/software/awdtools/rexx/language/
- Lua: egy egyszerű ragasztónyelv
- elastiC: C-szerű szintaxissal rendelkező nyelv
- És még sokan mások...

Típusok

Scriptnyelvekben általában előforduló típusok általános leírása

Sztringek

- Automatikusan méretváltoztató sztringek
- Bizonyos nyelvekben ,,interpoláció''
  Például Perlben, TCL-ben, stb.
  ($s, $n) = ('alma', 2);
  print qq{Ez $n láda $s\n};

Numerikus típusok

- Gyakran beépített típussal tetszőlegesen nagy egész számok is kezelhetők
- Esetleg lehet automatikus konverzió típusok között
- Általában egyfajta valós típus van

Listák

- Scriptnyelvekben általában tetszőlegesen hosszúak lehetnek
- A tömb és a lista fogalma összemosódik
- Végig-iterálhatunk rajtuk
- Meghatározhatjuk az elemszámukat
- Vehetünk egy részlistát
- Kicserélhetünk egy részlistát
- Beszúrhatunk listát
- Összefűzhetünk listákat
- Erejük megmutatkozik szövegfeldolgozási feladatoknál
- Elemeik sokfélék lehetnek

Hash

- Modern nyelvek tartalmazzák ezt az adattípust
- Szótár, asszociatív tömb, hasítótáblák
- Kulcs-érték párokat tartalmaz
- Egy érték lehet egy bonyolultabb adatszerkezet is
- Végig-iterálhatunk az egész adatszerkezeten, a sorrend ilyenkor nemdefiniált

Reguláris kifejezések

^regexp\.org$

Reguláris kifejezések

- Reguláris kifejezések mondatok egy halmazának tömör leírását teszik lehetővé
- Hasonló a célja a shell wildcardoknak is (*, ?), de ez annál jóval rugalmasabb
- A Perl egyfajta mérce lett a témában
- POSIX 1003.2, regex(7)
  - Basic (ed): nincs |, +, ?, stb.
  - Extended (egrep)
- Perl

Regex: alapfogalmak

- Reguláris kifejezés
  - Regular Expression, re, RE, regexp, regex
  - Mondatok egy halmazát írja le
  - Pl. /körte/
- Mintára illeszkedés
  - Egy sztring illeszkedik, ha eleme a fenti halmaznak.
  - /körte/-re illeszkedik minden olyan mondat (sztring), ami tartalmazza a *körte* szót (pl. *körtepálinka, vilmoskörte, villanykörte*)

Regex: metakarakterek

- '.': pontosan egy tetszőleges karakterre illeszkedik
- '^': sor eleje
- '$': sor vége
- '|': alternatíva
- '{}', '*', '+', '?': ismétlések
- '[]': karakter osztályok
- '()': csoportosítás
- '\': escape

Regex: karakterek illesztése

- /.ör.e/
  - körte, törpe, görbe, gödörbe, körtelé
- /^.ör.e/
  körte
  törpe
  görbe
  körtelé
- /^.ör.e$/
  körte
  törpe
  görbe

Regex: alternatívák

- /alfa|béta/
  - alfa, béta, analfabéta
- '()' használható csoportosításra
- /Ott (van|volt) egy alma/
  - Ott van egy alma
  - Ott volt egy alma

Regex: karakterosztályok

- /Ka[rj]ak/
  - Karak, Kajak
- /[0123456789ABCDEFabcdef]/
  - Hexadecimális számjegy
- /[0-9A-Fa-f]/ és /[0-9a-f]/i ekvivalens a fentivel
- ^ karakterrel illeszkedő karakterek halmazának komplementerét kell megadni
- /[^c]ici/
  - pici illeszkedik
  - A női testrész nem illeszkedik :)

Regex: előre definiált karakterosztályok

- POSIX: [:class:]
  - [:digit:]: számjegyek
  - [:alnum:]: alfanumerikus karakterek
  - [:alpha:]: szókarakterek
  - [:space:]: fehér szóközök
  - stb.
- Perl: \<char>
  - \d: számjegyek
  - \w: alfanumerikus karakterek + _
  - \s: fehér szóközök
  - stb.

Regex: illesztés szóhatárra

- 0 hosszúságú illeszkedés!
- POSIX 1003.2 szabvány szerint:
  - [:<:] : szó eleje
  - [:>:] : szó vége
- Egrep:
  - \<, \> : szó eleje, vége
  - \b: szóhatár
  - \B: nem szóhatár
  - /óra\>/
    - karóra, tanóra illeszkedik
    - óraszám nem illeszkedik
  - /óra\b/ ugyanígy viselkedik

Regex: ismétlések

- '*': tetszőleges számú ismétlődés
  - /a*/
  - '', a, aa, aaa, aaaa, ...
- '+': legalább egy előfordulás
  - /b+/
  - b, bb, bbb, bbbb, bbbbb, ...
- '?': 0 vagy 1 előfordulás
  - /\.html?/
  - .htm, .html

Regex: ismétlések

- '{n,m}': legalább n-szer és legfeljebb m-szer illeszkedik
  - /[[:digit:]]{2,4}/: '04', '2004'
- '{n}': pontosan n-szer illeszkedik
  - /[[:digit:]]{4}/: '2004'
- '{n,}': legalább n-szer illeszkedik
  - /[1-9][[:digit:]]{3,}/: legalább négyjegyű (decimális) számok

Regex: csoportosítások

- '()' között csoportosított karakterek visszahivatkozhatók a \1, \2, stb. hivatkozásokkal
- A '()' karakterek gyakran \ karaktert kívánnak, ha metakarakterként akarjuk használni őket
- Csere során nagyon hasznos (sed példa):
  - s/^\([[:digit:]]*\)-\(.*\)$/Track \1 has title “\2”/
  - '*01-Title of the first track*'
    ->
    'Track 01 has title “Title of the first track”'

Regex: mp3 tagelés példa

- Az előző példában megadott formátumú file.
- A következő shell paranccsal regexpek segítségével könnyedén megtaggelhetjük az mp3 fileokat:

```
IFS=''
for line in `sed 's/^\([[:digit:]]*\)-\(.*\)
$/mp3info -a “Artist” -l “Album” -t “\2” -n “\1”
Track-\1.mp3/' <mp3info.txt`; do
   eval $line;
done
```


Regex: alkalmazások

- grep, egrep: megfelelő sorok kiszűrése
- Szövegszerkesztés közben keresés
- Szövegszerkesztés közben csere
- Az előző példához hasonló shell kifejezések egyszeri alkalmakra
- Programozás során a szövegfeldolgozásban nagy segítséget nyújt
- Nem való XML illetve HTML források elemzésére

Regex: információforrások

- info sed
- grep(1), regex(7)
- Vim: help pattern
- http://www.perldoc.com/perl5.8.0/pod/perlre.html
- http://www.python.org/doc/howto/
- Programming Ruby
- ...

Regex: példák

- SELECT
  id, date, name, code, amount, color,
  weight, width, height
  FROM
  name_code_mappings m
- s/\</m./g
- s/\s*\(\w\+\)\(,\?\)\n\?/\tSUM(\1) AS \1\2\r/g
  - a g kapcsoló globális cserét eredményez, azaz minden előfordulást cserélni fog

Regex: példák

- természetes számok:
  - /^[[:digit:]]+$/
- egész számok
  - /^-?[[:digit:]]+$/ # +3-ra nem jó
  - /^[+-]?[[:digit:]]+$/