

# Γλώσσες Σεναρίων (Scripting Languages)



David Hockney, "Pacific Coast Highway and Santa Monica", 1990

# Γενικά Χαρακτηριστικά Γλωσσών Σεναρίων

- Οικονομία έκφρασης
- Μαζική (batch) ή διαδραστική (interactive) χρήση
- Απουσία δηλώσεων, απλοί κανόνες εμφάνισης
- Δυναμικό σύστημα τύπων
- Υψηλού επιπέδου τύποι δεδομένων
- Η εκτέλεση αρχίζει στην αρχή του αρχείου (δεν υπάρχει «main»)
- Χρήση **διερμηνέα**, όχι μεταγλωττιστή (συνήθως)
- Γενικά: *δυναμικές γλώσσες υψηλού επιπέδου*

# Τι είναι ένα «*scriptάκι*»;

Ένα μικρό πρόγραμμα (συνήθως), που χρησιμοποιείται για αυτοματοποίηση διαδικασιών (και όχι μόνο...)

```
for file in *_old.png; do mv "$file" "${file/_old.png/_new.png}"; done
```

Παραδείγματα:

- batch processing
- επεξεργασία αρχείων κειμένου
- «συγκόλληση» άλλων προγραμμάτων
- software development (build and CI scripts)

# Εφαρμογές (I)

- Γλώσσες εντολών φλοιού (shell command languages)
  - bash, zsh, csh,...
- Επεξεργασία κειμένου
  - sed, awk, Perl
- Μαθηματικά και στατιστική
  - Maple, Mathematica, Matlab, R, S
- Γλώσσες συγκόλλησης (glue) και σενάρια γενικού σκοπού
  - Python, Ruby, ...
- Γλώσσες επέκτασης (extension languages)
  - JavaScript, Visual Basic, Emacs Lisp

# Εφαρμογές (II)

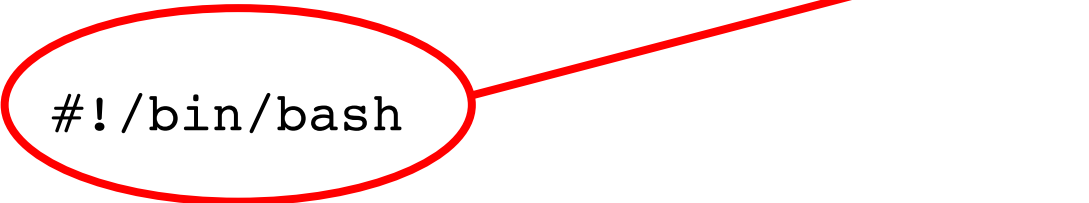
- WWW scripting
  - CGI (Perl)
  - Server-side (PHP, VisualBasic,...)
  - Client-side (JavaScript,..)
  - Applets (Java,...)
  - Άλλες τεχνολογίες (XSLT, XML)

# Shell Scripting

## Shebang

(or hashbang, or sha-bang, or ...)

`#!/bin/bash`



```
while [ "$1" != "" ]; do
  if [ -r "lecture-$1.pdf" ]; then
    pdfjam --nup 2x2 --paper a4paper --landscape --frame true \
      --outfile handout-$1.pdf lecture-$1.pdf
  else
    echo "failed to convert lecture-$1.pdf to handout-$1.pdf"
  fi
  shift # shifts arguments
done
```

Μετατροπή διαφανειών (του μαθήματος «Γλώσσες Προγραμματισμού») σε handouts

# Sed Script

```
# label (target for branch):
:top
/<[hH] [123]>.*<\/[hH] [123]>/ {
    h
    s/\(<\/[hH] [123]>\).*$/\1/
    s/^\.*\(<[hH] [123]>\)/\1/
    p
    g
    s/<\/[hH] [123]>//
    b top
}
/<[hH] [123]>/ {
    N
    b top
}
d
```

;

- ταίριαξε όλη την επικεφαλίδα
- αντίγραψε το χώρο προτύπων
- σβήσε το κείμενο μετά το κλείσιμο
- σβήσε το κείμενο πριν το άνοιγμα
- τύπωσε ό,τι απέμεινε
- επανάφερε το χώρο προτύπων
- σβήσε το κλείσιμο

;

- και πήγαινε στην αρχή του σεναρίου
- ταίριαξε το άνοιγμα (μόνο)
- επέκτεινε την αναζήτηση
- στην επόμενη γραμμή
- και πήγαινε στην αρχή του σεναρίου
- αν δεν ταιριάζει, σβήσε

Εξαγωγή επικεφαλίδων HTML. Υποθέτει ότι οι επικεφαλίδες είναι ταιριασμένες και όχι ένθετες.

# Awk Script

```
/<[hH] [123]>/ {
  # εκτέλεσε αυτό το μπλοκ αν η γραμμή περιέχει ετικέτα ανοίγματος
  do {
    open_tag = match($0, /<[hH] [123]>/)
    $0 = substr($0, open_tag)      # σβήσε το κείμενο πριν από αυτή
                                   # $0 είναι η τρέχουσα γραμμή εισόδου

    while (!/<\/[hH] [123]>/) {    # τύπωσε τις εσωτερικές γραμμές
      print                       # ολόκληρες
      if (getline != 1) exit
    }
    close_tag = match($0, /<\/[hH] [123]>/) + 4

    print substr($0, 0, close_tag) # τύπωσε μέχρι το κλείσιμο
    $0 = substr($0, close_tag + 1) # σβήσε μέχρι το κλείσιμο
  } while (/<[hH] [123]>/)
}
```

Εξαγωγή επικεφαλίδων HTML. Υποθέτει ότι οι επικεφαλίδες είναι ταιριασμένες και όχι ένθετες.



# Perl Script

```
while (<>) {                                     # επανάλαβε για κάθε γραμμή εισόδου
    next if !/<[hH][123]>/;                       # άλμα στην επόμενη επανάληψη
    while (!/<\/[hH][123]>/) { $_ .= <>; }        # πρόσθεσε επόμενη γραμμή στο $_
    s/.*?(<[hH][123]>.*?<\/[hH][123]>)//s;
    # ελάχιστο ταίριασμα; η έκφραση με παρενθέσεις μπαίνει στο $1
    print $1, "\n";
    redo unless eof;                             # συνέχισε χωρίς να διαβάσεις επόμενη γραμμή
}
```

Εξαγωγή επικεφαλίδων HTML. Υποθέτει ότι οι επικεφαλίδες είναι ταιριασμένες και όχι ένθετες.

# Python

- Δυναμική γλώσσα γενικής χρήσης
- Multi-paradigm (imperative, object oriented, functional στοιχεία)
- Garbage collected
- Python 2 (deprecated) vs **Python 3**
  - Not backwards compatible...

# The Python Top-Level

```
→ ~ python3
```

```
Python 3.12.3 (main, Apr 9 2024, 16:03:47) [Clang 14.0.0 (clang-1400.0.29.202)] on darwin  
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> print("hello world")
```

```
hello world
```

```
>>> █
```

# Anaconda



- Διανομή Python και R
- Ευρεία χρήση σε scientific computing (data science, machine learning, data processing)
- Πάνω από 250 πακέτα
- Package manager (conda)
- GUI alternative to CLI
- ....

# Run a Python Program

```
→ ~ cat hello.py  
print("hello world!")  
→ ~ python3 hello.py  
hello world!  
→ ~
```

# Run a Python Program as Executable (I)

```
→ ~ cat hello.py
#!/usr/bin/python3

print("hello world!")
→ ~ chmod +x hello.py
→ ~ ./hello.py
hello world!
→ ~ █
```

# Run a Python Program as Executable (II)

```
→ ~ cat hello.py
#!/usr/bin/env python3

print("hello world!")
→ ~ ./hello.py
hello world!
→ ~ █
```

Για τις διαλέξεις θα χρησιμοποιήσουμε  
Jupyter Notebook

<https://jupyter.org/install>