

## Parte 1

# Linguaggi dinamici: cenni storici

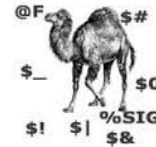
## Un po' di storia (1970-1980)

- L'invenzione dei linguaggi riflette il periodo storico e le necessità dei programmatori
- Sistemi di grandi dimensioni (mainframe):
  - Applicazioni: calcolo scientifico
  - Interfacce: testuali
  - Amministrazione: scripting
- Tale contesto ha portato a:
  - C (veloce, adatto al calcolo, poco portabile)
  - Assembly (molto veloce, non portabile, adatto all'interfacciamento con l'hardware)
  - Shell (lento, portabile, adatto per manutenzione)



## Un po' di storia (1980-1990)

- L'informatica prende piede su media e piccola scala
  - Singoli PC (Workstation)
  - Reti locali
- Sempre più persone hanno il compito di “amministrare” un calcolatore (*sistemisti*)
- Nasce il primo linguaggio di scripting “serio”: Perl (Larry Wall 1987) – nato per essere pratico
  - Portabile e veloce
  - Orientato a manipolazione di file di testo (utile per gestire configurazioni e analizzare log)



Linguaggi dinamici

3

## Un po' di storia (1990-2000)

- Invenzione (1991) e successivo boom (1995) del World Wide Web
- Nascono linguaggi nuovi per poter programmare applicazioni Web-based con maggiore semplicità rispetto al C
  - PHP (*PHP Hypertext Preprocessor*) nasce nel 1994 per creare pagine Web dinamiche
  - Javascript (Mocha) nasce nel 1995 per essere usato in Netscape (uno dei primi browser Web)
- I linguaggi esistenti si adattano per non perdere terreno
  - C e Perl vengono tipicamente usati per i CGI



Linguaggi dinamici

4

## Un po' di storia (1990-2000)

- I programmatori cominciano a comprendere 2 grandi verità sull'architettura dei calcolatori
  - L'hardware delle macchine diventa più potente: ritmo simile alla legge di Moore (1965!)
  - Scrivere codice a basso livello diventa sempre più difficile, per via dell'aumentata complessità delle architetture
- Si diffonde il concetto di “linguaggio ad alto livello general purpose”
  - Python (Guido Van Rossum, 1991)
  - Ruby (Yukihiro Matsumoto, 1993)
  - Java (James Gosling, 1995)



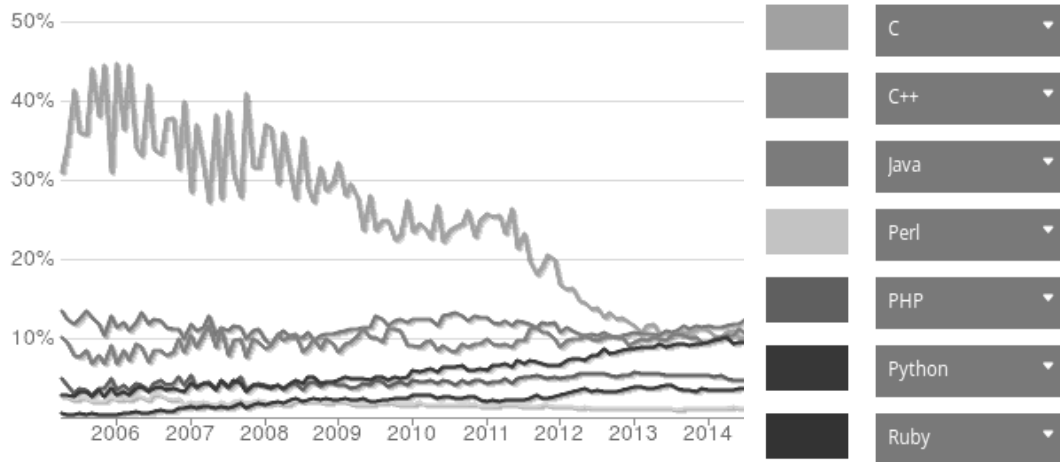
## Un po' di storia (2000-)

- L'uso dei linguaggi di alto livello si estende a progetti su larga scala
- I programmatori si rendono conto che *il ciclo di sviluppo del software deve essere più semplice e breve*
- I linguaggi di alto livello sono corredati di una miriade di librerie per le funzioni più disparate
- Nasce il framework: linguaggio ad alto livello unito ad una serie di librerie general-purpose
  - *.NET (compilatore C#, CLR e librerie)*
  - *Mono e Portable.NET (implementazioni open source e portabili di .NET)*

# Un rapido sguardo ai giorni nostri

## Monthly Commits (Percent of Total)

From HopeHub, an online repository of Free and Open Source Software



Linguaggi dinamici

7

## Linguaggi dinamici: caratteristiche

Linguaggi dinamici

8

## Linguaggio statico

- Ha una fase di compilazione, in cui il codice sorgente viene tradotto in un formato a basso livello esclusivo per l'architettura considerata
  - La traduzione da codice sorgente a codice macchina è 1:1 (rappresentazione fedele)
- Permette l'esecuzione ad una velocità elevata
- I tipi di dati sono identificati a tempo di compilazione e non sono mutabili a tempo di esecuzione
- Non fornisce strumenti di controllo, né semplificazioni, è tutto lasciato al programmatore (memoria, tipizzazione)

## Linguaggio dinamico

- Ha una fase di compilazione, in cui il codice sorgente viene tradotto in un formato intermedio indipendente dall'architettura considerata (bytecode o AST)
- Il formato intermedio viene interpretato in maniera efficiente da un interprete, tenendo conto dell'architettura considerata
- L'esecuzione non è così veloce come in un linguaggio statico, ma è portabile (nel formato intermedio)

## Linguaggio dinamico

- L'interprete si serve di funzioni interne per allocare, deallocare, referenziare memoria in maniera automatica a run-time
- Ha una tipizzazione dinamica dei dati
  - I tipi di dato possono mutare a run-time
- Ha la caratteristica di sapersi “analizzare” e “modificare” durante l'esecuzione (Metaprogramming)
  - Eseguire funzioni diverse a seconda delle condizioni operative a run-time
  - Cambiare il codice stesso del programma
  - Creare strutture dati variabili nel tempo

## Linguaggi dinamici - definizione

Possibile definizione (da Wikipedia)

- **Linguaggio statico:** è un linguaggio ad alto livello in cui le operazioni effettuate a livello di esecuzione (run time) sono legate (quasi) esclusivamente all'esecuzione di codice (se si esclude la gestione dello stack)
  - C, assembly
- **Linguaggio dinamico:** è un linguaggio ad alto livello in cui le operazioni effettuate a livello di esecuzione (run time) non sono legate esclusivamente all'esecuzione di codice
  - Perl, Python, Ruby, PHP, Javascript

## Linguaggi dinamici - definizione

- Non esiste in realtà una definizione univoca, ma un insieme di caratteristiche tipiche
  - *Tipizzazione dei dati*
  - *Metaprogramming*
  - *Gestione dinamica della memoria*
  - *Modello di generazione del codice (prodotto intermedio es. bytecode)*
- Alcune caratteristiche si possono ritrovare in altri linguaggi non considerati propriamente dinamici
  - Es. Java

Operazioni  
svolte a run-time



## Linguaggi dinamici: assenza di dettagli ostici

- Nei linguaggi come il C, al programmatore è lasciato ogni dettaglio:
  - Gestione della memoria
  - Gestione dei puntatori
  - Rilascio delle risorse
- Tali task non sono funzionali al problema che si intende risolvere
  - Rallentano il ciclo di sviluppo del software
- Nei linguaggi dinamici tali aspetti sono gestiti automaticamente

## **Linguaggi dinamici: presenza massiccia di librerie**

- Nei linguaggi come il C, la vera libreria di riferimento è la libreria del C
  - Piuttosto complessa da usare
  - Fornisce essenzialmente servizi “di base”
- Parecchi dei compiti richiesti da un applicativo moderno non sono presi in considerazione
  - Servizi di calcolo scientifico, interfacce grafiche complesse, supporto per il Web
- Nei linguaggi dinamici, tali funzionalità sono fornite da librerie esterne facilmente usabili

## **Linguaggi dinamici: portabilità**

- Nei linguaggi come il C, la portabilità si ottiene tramite strumenti a dir poco complessi:
  - Codice condizionato - richiede parecchio lavoro allo sviluppatore (scrittura delle macro AC\_XYZ)
  - Utilizzo non immediato per l'utente inesperto (`./configure --prefix=/usr/local ...`)
  - Autotools (autoconf, automake, ...)
- Nei linguaggi dinamici, il codice scritto è perfettamente portabile
  - Funziona in diversi ambienti operativi senza alcuna modifica → *codice adattabile*



## Linguaggi dinamici vs statici: pro e contro

- **Pro:**
  - La scrittura di codice è estremamente semplificata rispetto ai linguaggi statici
  - La scrittura di un software diventa la scrittura del suo scheletro, grazie alla presenza delle librerie esterne → forte riuso del codice
  - Funziona su diverse piattaforme
- **Contro:**
  - Più lento (fino a 10 volte) del codice macchina nativo
  - Può incoraggiare la programmazione “sloppy”

Sloppy...



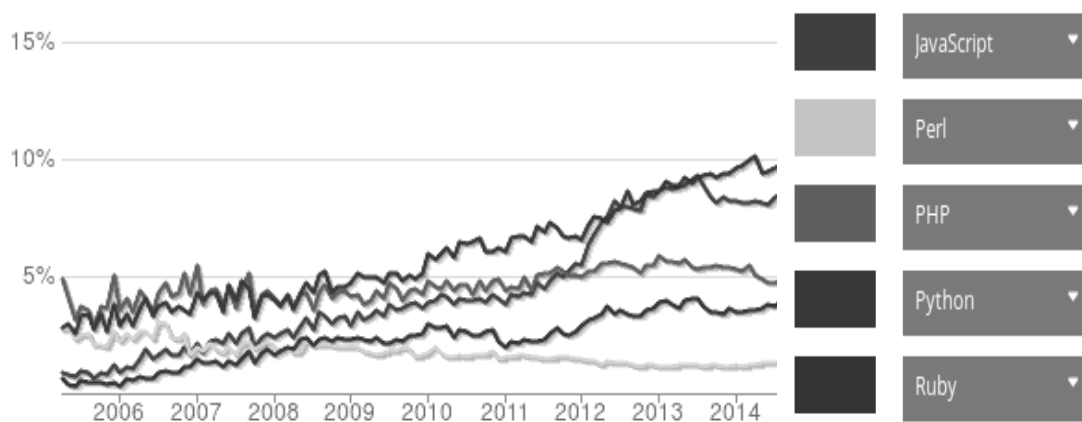
Linguaggi dinamici

17

## Perchè focalizzarci su Python?

### Monthly Commits (Percent of Total)

From OpenHub, an online repository of Free and Open Source Software



Python è il linguaggio dinamico più usato attualmente

Linguaggi dinamici

18

# Perchè focalizzarci su Python?

- Usato nei più popolari siti Web (server-side)

Programming languages used in most popular websites\*

Websites ↕	Popularity (unique visitors) <sup>[1]</sup> ↕	Front-end (Client-side) ↕	Back-end (Server-side) ↕	Database ↕	Notes
Google.com <sup>[2]</sup>	1,100,000,000	JavaScript	C, C++, Go, <sup>[3]</sup> Java, Python	BigTable, <sup>[4]</sup> MariaDB <sup>[5]</sup>	The most used search engine in the world
YouTube.com	1,000,000,000	Flash,HTML5, JavaScript	C/C++, Python, Java, <sup>[6]</sup> Go <sup>[7]</sup>	MySQL, BigTable	The most visited video sharing site
Facebook.com	900,000,000	JavaScript	Hack, PHP, C++, Java, Python, Erlang, D, <sup>[8]</sup> Xhp <sup>[9]</sup>	MySQL, <sup>[10]</sup> HBase	The most visited social networking site

\*dati gennaio 2015

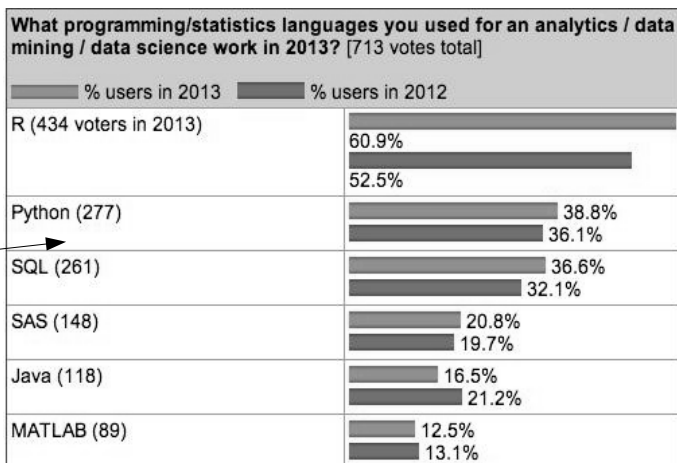
Linguaggi dinamici

19

# Perchè focalizzarci su Python?

- Popolarità in ascesa
- E' il linguaggio dinamico di riferimento per la piattaforma dei servizi offerti da Google
- Curva di apprendimento ripida
- Flessibilità
- Semplicità

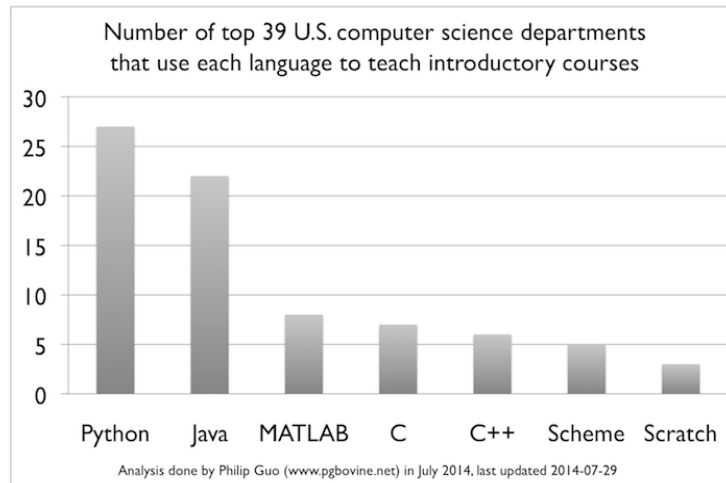
Popolare nel contesto della **analisi dati**  
Interfacciabile con software scientifico R



Linguaggi dinamici

20

## Trend delle Università Americane



**Python ha superato Java (linguaggio predominante per gli ultimi 10 anni)**

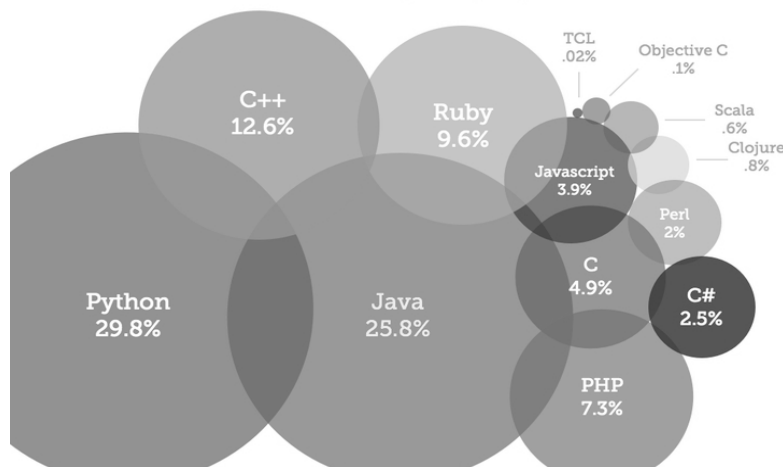
Linguaggi dinamici

21

## Rilevazioni CodeEval

**CodeEval = comunità di sviluppatori interessati a risolvere sfide di programmazione (contest)**

Most Popular Coding Languages of 2013



**Nota:** le sfide possono essere risolte in un linguaggio a piacere

## **Ibrido: Google Go**

- **Google ha da pochi anni (fine 2009) introdotto il linguaggio Go**
- **Alcune caratteristiche dinamiche → semplicità**
- **Alcune caratteristiche statiche → particolare attenzione alla velocità di esecuzione**
- **Mira a fornire la velocità e l'efficienza della tipizzazione statiche ma con la facilità di programmazione dei linguaggi dinamici**
- **<http://www.golang.org>**