

## Informazioni generali sull'insegnamento

# Didattica

- **4 ore settimanali**
  - mercoledì 9-11, laboratorio
  - giovedì 11-13, aula
- per 12 settimane
- **6 CFU (corrispondenti a 'circa' 150 ore di lavoro da parte dello studente)**
- **Suddivisione: 2CFU teoria, 4CFU laboratorio (Python)**
- **Esame: prova pratica di laboratorio + orale**
- **Ricevimento: mercoledì 14.30-16.00**

# Didattica

- **Materiale di studio**
  - **Slide proiettate a lezione**
  - **Materiale disponibile in Internet (segnalato di volta in volta)**
  - **Altro materiale messo a disposizione dal docente su GitHub**  
<https://github.com/leoncini/Linguaggi-Dinamici-2019>
- **Comunicazioni e avvisi sempre sulla stessa pagina**

# Argomenti

- **Teoria:**
  - Linguaggi di programmazione
  - Tipizzazione forte e debole
  - Architettura generale di un l.d.
  - Gestione della memoria
  - Metaprogramming
- **Labratorio di linguaggio Python 3**
  - Sintassi e primi semplici esempi 'imperativi'
  - OOP e il modello delle classi in Python
  - Programmazione funzionale
  - Metaprogramming

# Linguaggi dinamici: definizione e caratteristiche

# Definizioni

## Una possibile definizione (cfr. Wikipedia)

- **Linguaggio statico:** è un linguaggio *high-level* in cui le operazioni effettuate a *run time* sono legate *quasi* esclusivamente all'esecuzione del codice del programmatore (un'eccezione notevole è la gestione dello stack)
  - C, assembly

# Definizioni

## Una possibile definizione (cfr. Wikipedia)

- **Linguaggio statico:** è un linguaggio *high-level* in cui le operazioni effettuate a *run time* sono legate *quasi* esclusivamente all'esecuzione del codice del programmatore (un'eccezione notevole è la gestione dello stack)
  - C, assembly
- **Linguaggio dinamico:** è un linguaggio ad alto livello in cui le operazioni effettuate a tempo di esecuzione 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 specifiche**
  - *Tipizzazione dei dati*
  - *Metaprogramming*
  - *Gestione dinamica di memoria ed errori*
  - *Modello di generazione del codice (prodotto intermedio es. bytecode)*
- **Alcune caratteristiche si possono ritrovare in altri linguaggi considerati non propriamente dinamici**
  - **Es. Gestione della memoria in Java**

**Operazioni  
svolte a run-time**





# Linguaggio statico

- Esiste una fase di **compilazione**, in cui il codice sorgente viene tradotto in un formato 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, **quasi tutto è lasciato al programmatore (memoria, tipizzazione)**

# Linguaggio dinamico

- Nella fase di **compilazione** il codice sorgente viene tradotto in un **formato intermedio**, indipendente dall'architettura (es. **bytecode**)
  - Il formato intermedio è interpretato  
→ **linguaggio portabile**
- L'interprete si serve di funzioni interne per gestire **memoria ed errori in modo automatico a run-time**
  - **Assenza di dettagli ostici per il programmatore**
- Ha una **tipizzazione dinamica dei dati**
  - Il 'tipo' di una variabile (ma anche di una funzione) può mutare a run-time

# Linguaggio dinamico

- Include meccanismi che permettono al programma di “analizzarsi” e “modificarsi” durante l'esecuzione (**Meta-programming**)
- Meta-programming ha a che vedere con la possibilità di costruire funzioni e classi il cui obiettivo principale è la manipolazione di codice:
  - Generazione di “nuovo” codice
  - Modifica e wrapping di codice **esistente**
- Meccanismi fondamentali per il meta-programming sono
  - Decoratori
  - Meta-classi
  - Decoratori di meta-classi

# Altre caratteristiche

- Possibilità di creare strutture dati, anche eterogenee, variabili nel tempo
- Presenza massiccia di librerie esterne facilmente utilizzabili per diversi compiti
  - Servizi di calcolo scientifico
  - Interfacce grafiche complesse
  - Supporto per il Web

# Linguaggi dinamici vs statici

## ■ Pro:

- Curva di apprendimento
- Scrittura di codice
  - La scrittura di un software diventa la scrittura del suo **scheletro**
  - Presenza delle librerie esterne  
→ **forte riuso del codice**
  - Prototipazione veloce
- Flessibilità e portabilità

## ■ Contro:

- **Lentezza** (a causa delle molte operazioni a run-time)
- Può incoraggiare uno stile “**sciatto**” di programmazione (**sloppy coding**)

# Python è il linguaggio più popolare

**PYPL Index** - Popularity of Programming Language Index is created by analyzing how often language tutorials are searched on Google

Worldwide, Mar 2016 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Java	24.2 %	+0.3 %
2	↑	Python	11.9 %	+1.2 %
3	↓	PHP	10.7 %	-0.8 %
4		C#	8.9 %	+0.1 %
5		C++	7.6 %	-0.5 %
6		C	7.5 %	+0.1 %
7		Javascript	7.3 %	+0.3 %
8		Objective-C	5.0 %	-0.9 %
9	↑↑	Swift	3.0 %	+0.4 %
10		R	2.9 %	+0.3 %
11	↓↓	Matlab	2.8 %	-0.3 %
12		Ruby	2.3 %	-0.2 %
13		Visual Basic	1.8 %	-0.4 %
14		VBA	1.5 %	+0.1 %
15		Perl	1.1 %	-0.1 %

Worldwide, Aug 2019 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	28.73 %	+4.5 %
2		Java	20.0 %	-2.1 %
3		Javascript	8.35 %	-0.1 %
4		C#	7.43 %	-0.5 %
5		PHP	6.83 %	-1.0 %
6		C/C++	5.87 %	-0.3 %
7		R	3.92 %	-0.2 %
8		Objective-C	2.7 %	-0.6 %
9		Swift	2.41 %	-0.3 %
10		Matlab	1.87 %	-0.3 %
11	↑	TypeScript	1.76 %	+0.2 %
12	↓	Ruby	1.44 %	-0.2 %
13	↑↑↑	Kotlin	1.43 %	+0.4 %
14	↓	VBA	1.41 %	-0.0 %
15	↑↑	Go	1.21 %	+0.3 %

# Perchè ci focalizzeremo su Python

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

Programming languages used in most popular websites*					
Websites	Popularity (unique visitors per month) <sup>[1]</sup>	Front- end (Client- side)	Back-end (Server- side)	Database	Notes
Google.com <sup>[2]</sup>	1,600,000,000	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
Facebook.com	1,100,000,000	JavaScript	Hack, PHP (HHVM), Python, C++, Java, Erlang, D, <sup>[6]</sup> Xhp, <sup>[7]</sup> Haskell <sup>[8]</sup>	MariaDB, MySQL, <sup>[9]</sup> HBase Cassandra <sup>[10]</sup>	The most visited social networking site
YouTube.com	1,100,000,000	JavaScript	C, C++, Python, Java, <sup>[11]</sup> Go <sup>[12]</sup>	Vitess, BigTable, MariaDB <sup>[5][13]</sup>	The most visited video sharing site

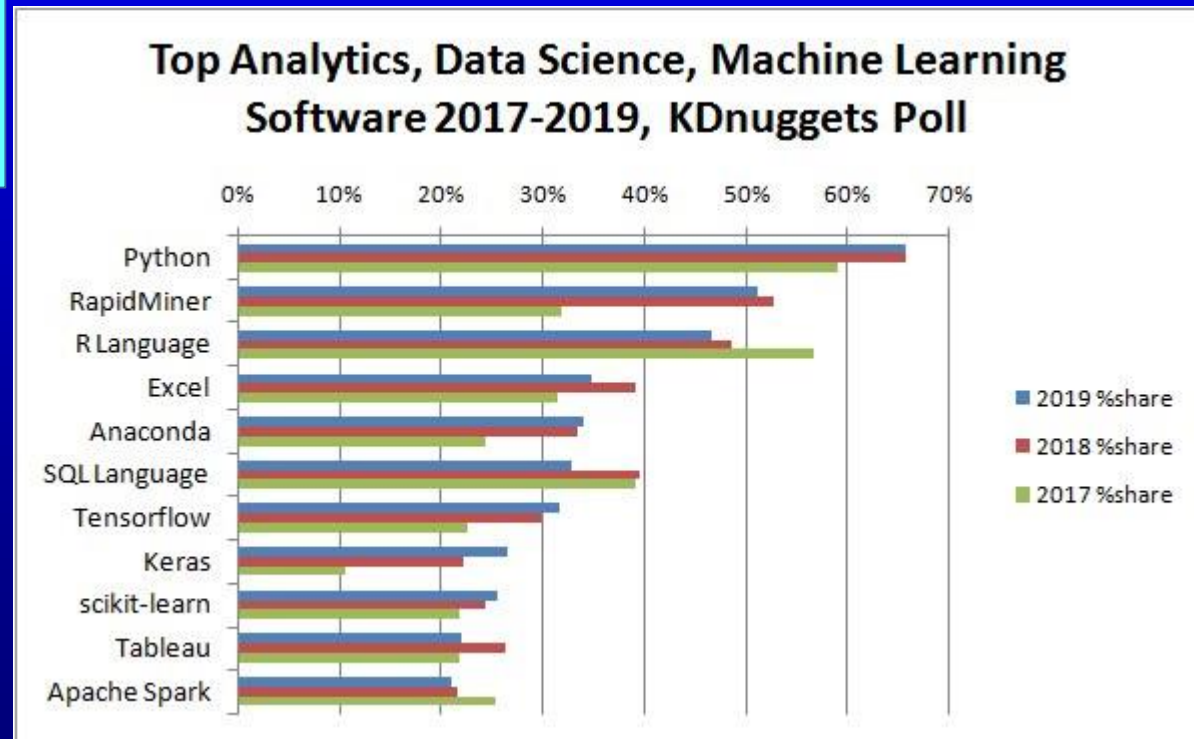
GO

\*dati 2018

# Perchè ci focalizzeremo su Python

- Popolarità in continua ascesa anche nel campo della Data Science e del Machine Learning

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

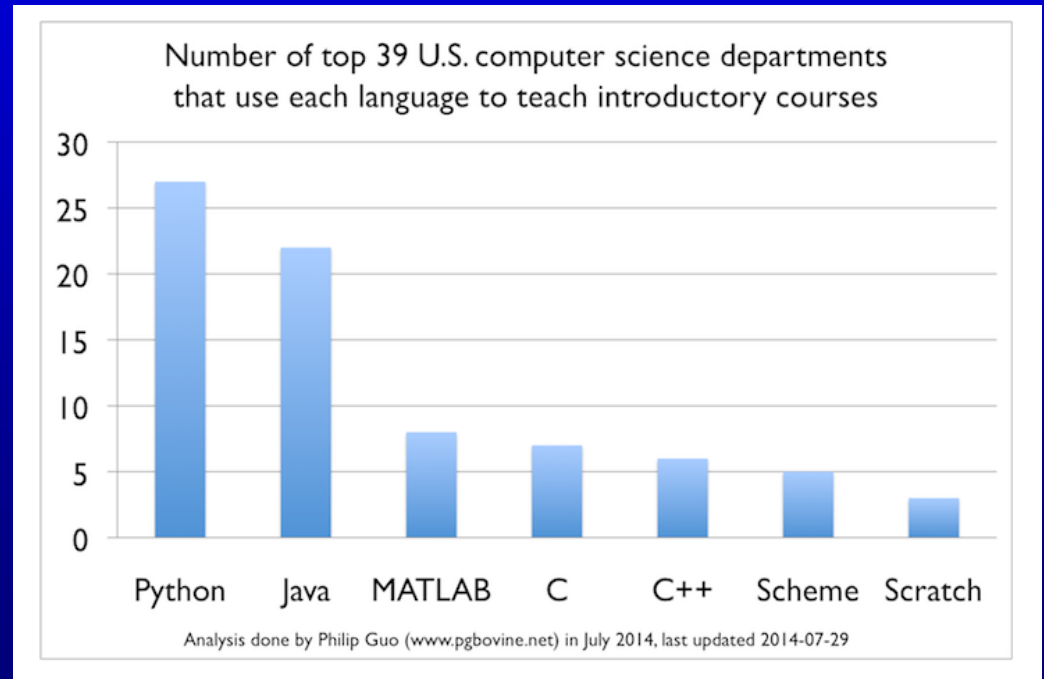




# Trend delle Università Americane

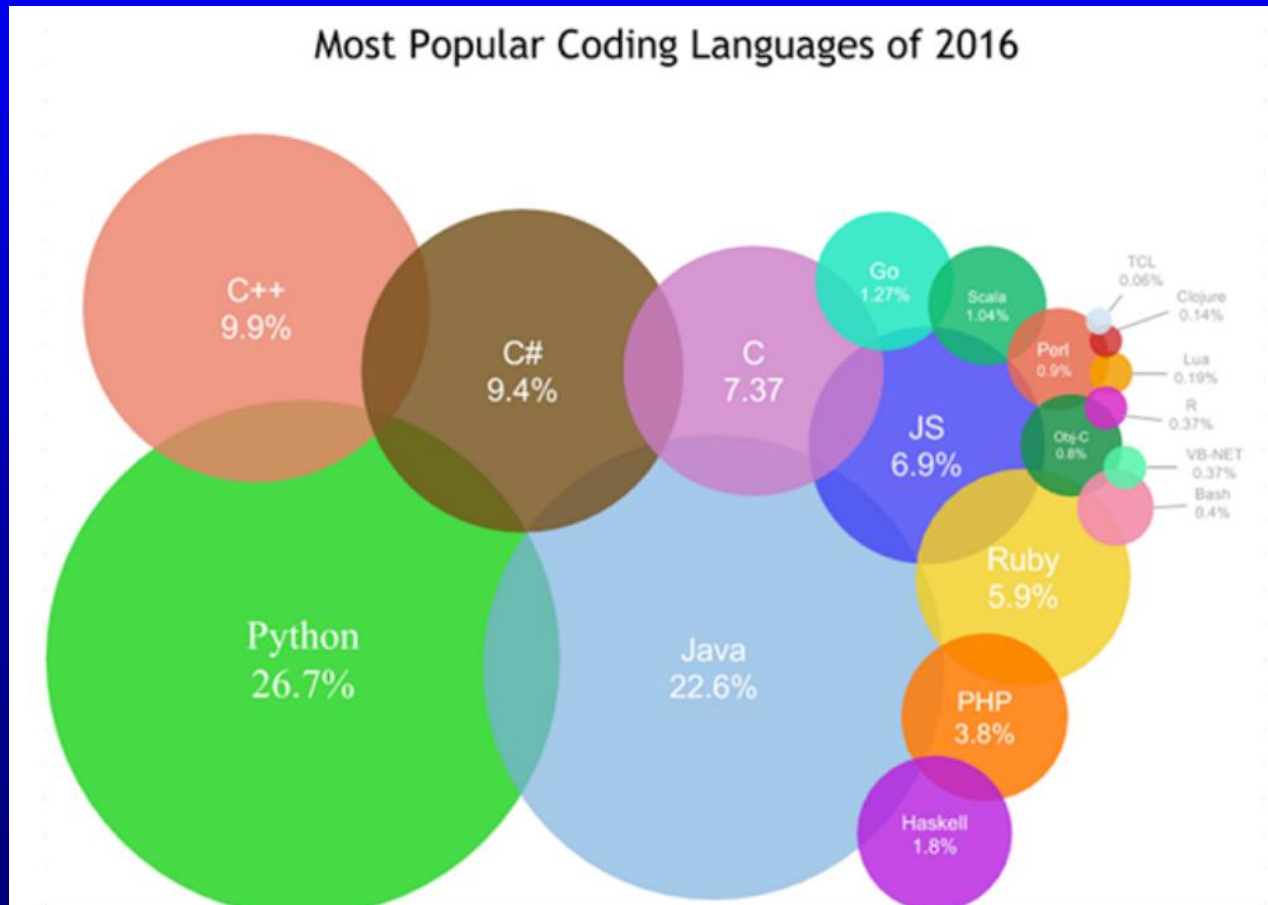
- Python Is Now the Most Popular Introductory Teaching Language at Top U.S. Universities
  - Communications of the ACM (July 7, 2014)
- Universities finally realize that Java is a bad introductory programming language
  - thenextweb.com (Apr 24, 2017)

**Python ha superato  
Java (linguaggio  
predominante per  
molti anni)**



# Rilevazioni CodeEval

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



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

# Rilevazioni 2019

