

Linguaggi dinamici

Claudia Canali
claudia.canali@unimore.it

Obiettivi formativi del corso

- **Competenze di base**
 - Capire *cosa sono* i linguaggi dinamici, caratteristiche salienti, *differenze* rispetto ad altri
 - *Comprendere frammenti di codice* di diversi linguaggi dinamici
 - Imparare a *programmare in Python*
 - Imparare a realizzare un semplice *applicativo Web* basato sul paradigma *Model-View-Controller (MVC)*

Obiettivi formativi del corso

- **Competenze avanzate**
 - Esser in grado di assimilare le caratteristiche di *nuovi* linguaggi dinamici in tempi brevi
 - Saper sfruttare le caratteristiche tipiche dei linguaggi dinamici per la prototipazione rapida e il testing del software
 - Saper scegliere il linguaggio adatto allo scopo!
 - Acquisire la padronanza di linguaggi utili per lo sviluppo di funzionalità avanzate (scripting avanzato, applicativi Web-based, calcolo scientifico, utilizzo API)

Programma del corso

Prima parte: presentazione linguaggi dinamici principali (*Python, Perl, Ruby, PHP*) e loro differenze con linguaggi già noti (*C, C++, Java*)

- Architettura di un linguaggio dinamico
- Tipizzazione dei dati
- Gestione run-time della memoria, del codice, degli errori
- Strumenti di supporto per:
 - modularizzazione del software
 - ricerca di errori (debugging)
 - verifica funzionale (unit testing)

Programma del corso

Seconda parte: *Il linguaggio Python*

- Costrutti di base
- Strutture dati avanzate
- Caratteristiche dinamiche
- Organizzazione ad oggetti
- Strutturazione in moduli e package
- Configurazione
- Parallelizzazione e gestione dei Thread
- Esempi di utilizzo di librerie esterne per funzioni specifiche

Programma del corso

Terza parte: *Framework Web Django*

- Caratteristiche del Framework
- Paradigma Model-View-Controller
- Interazione con database
- Guida allo sviluppo di un'applicazione di prova
- Analisi di un'applicazione reale complessa
- Esempi di funzionalità avanzate

Prerequisiti e materiale didattico

- **Prerequisiti obbligatori:**
 - Programmazione I
 - Programmazione II
 - Programmazione a oggetti
- **Prerequisito fortemente consigliato:**
 - Sistemi operativi
- **Materiale didattico**
 - Sul sito del corso
 - Slide protette
 - Username: LD1415 - Pwd: chime77spes

Modalità di comunicazione

Studente → docente

- **Per chiarimenti o per fissare ricevimenti/esami**
 - Via email

Docente → studenti

- **Per comunicazioni standard**
 - Sito Web del corso
- **Per comunicazioni veloci e urgenti**
 - Vostra mailing list (in formato elettronico!)
 - Altri strumenti ? (Google group?)

Modalità di esame

L'esame consiste di 2 parti:

1) Sviluppo di un progetto software

2) Colloquio orale

- **Discussione approfondita del progetto**
- **Domande sul contenuto del corso**

- **Il progetto va consegnato almeno 7 giorni prima dell'esame orale**
- **Per fissare l'appello contattare il docente via email**

Progetto d'esame

- **Le tracce saranno comunicate durante il corso**
- **Progetto (1 o 2 studenti a seconda della traccia):**
 - **Progetto ed implementazione del software come da traccia assegnata**
 - **Produzione di un file README con istruzioni**
 - **Stesura di una tesina (max. 10 pagine)**
 - **Capitolo 1: Descrizione del progetto**
 - **Capitolo 2: Modularizzazione e organizzazione**
 - **Capitolo 3: Scelte implementative**

Software da installare

Interprete Python (v. 2.7)

- **Disponibile nella maggior parte delle distribuzioni Linux e delle versioni MacOS**
- **Da installare in Windows**

IDE

- **PyCharm (fortemente consigliato)**
 - **Community Edition (solo Python!) - gratuita**
 - **Professional Edition: Python, Django, Javascript, e molto altro - licenza studenti**
- **Plugin per Eclipse: PyDev (sconsigliato)**

Vari pacchetti per applicazione avanzate Python