



Insegnamento	Informatica
Livello e corso di studio	Laurea Triennale in Scienze e Tecniche Psicologiche (Classe L-24)
Settore scientifico disciplinare (SSD)	INF-01
Anno di corso	3°
Numero totale di crediti	5
Propedeuticità	
Docente	Luca Ducceschi https://ricerca.unicusano.it/author/luca-ducceschi/ Nickname: luca.ducceschi Email: luca.ducceschi@unitn.it Orario di ricevimento: mercoledì, 17-19
Presentazione	L'obiettivo del corso è quello di consegnare allo studente delle competenze informatiche e di programmazione di base e molto pratiche. Il linguaggio di programmazione utilizzato sarà Python, con alcuni cenni di programmazione da riga di comando (CLI). L'argomento utilizzato per fare pratica di programmazione sarà soprattutto il processamento del linguaggio naturale, e quindi la manipolazione di file di testo e la gestione di basi di dati.
Obiettivi formativi	<ul style="list-style-type: none">● Approfondire la comprensione del funzionamento del computer● Sviluppo di una competenza algoritmica● Sviluppo di un approccio problem-based all'apprendimento della programmazione● Approfondire la comprensione dei problemi informatici riguardanti il linguaggio umano e l'applicazione di soluzioni a temi vicini alla psicologia.● Sviluppo di una autonomia nella gestione delle risorse informatiche
Prerequisiti	Account Google per l'utilizzo di Colaboratory
Risultati di apprendimento attesi	<p>Lo studente al termine del Corso avrà dimostrato di aver acquisito la conoscenza e la capacità di comprensione di concetti di programmazione di base, quali i tipi di dato fondamentali e la gestione di costrutti di base (for, if, while, etc.).</p> <p>Lo studente sarà in grado di utilizzare la conoscenza acquisita nel settore della programmazione in Python per generare e gestire risorse informatiche utili alla ricerca in psicologia e utili, in generale, alla soluzione di problemi informatici, quali il processamento di una risorsa di testo, l'estrazione di conoscenza da risorse non annotate e l'eventuale utilizzo di modelli di machine learning.</p>



	<p>Lo studente sarà in grado di proporre soluzioni personali a problemi non affrontati prima, generalizzando le competenze acquisite durante il corso.</p> <p>Lo studente sarà in grado di descrivere e sostenere conversazioni in ambito informatico grazie alla maggiore dimestichezza con i concetti fondamentali sviluppata durante il corso.</p> <p>Lo studente al termine del Corso avrà sviluppato una generale attitudine all'apprendimento di nuovi concetti nel settore dell'informatica e sarà in grado di strutturare velocemente un proprio percorso di autoapprendimento basato su un approccio problem-based (ovvero imparando a partire da progetti concreti).</p>
Organizzazione dell'insegnamento	<p>Il corso è sviluppato attraverso le lezioni preregistrate audio-video che compongono, insieme a slide e dispense, i materiali di studio disponibili in piattaforma.</p> <p>Sono poi proposti dei test di autovalutazione, di tipo asincrono, che corredano le lezioni preregistrate e consentono agli studenti di accertare sia la comprensione, sia il grado di conoscenza acquisita dei contenuti di ognuna delle lezioni.</p> <p>La didattica interattiva è svolta nel forum della "classe virtuale" e comprende almeno 3 Etivity</p> <p>In particolare, il Corso di Informatica prevede 5 Crediti formativi. Il carico totale di studio per questo modulo di insegnamento è compreso tra 80 e le 100 ore così suddivise in:</p> <ul style="list-style-type: none">• circa 30 ore per la visualizzazione e lo studio del materiale videoregistrato (15 Ore videoregistrate di Teoria e 15 ore di esercitazioni);• circa 20 ore di Didattica Interattiva per l'elaborazione e la consegna delle Etivity;• circa 5 ore di Didattica Interattiva per l'esecuzione dei test di autovalutazione.• circa 40 ore per la messa a punto dei progetti.
Contenuti del corso	<p>Modulo 1</p> <ul style="list-style-type: none">• Cosa significa scienza dell'informazione o informatica e come si impara a programmare• Python e la programmazione ad oggetti: introduzione all'ambiente di sviluppo Colaboratory.• Tipi di dato (stringhe, liste, interi e decimali), metodi, variabili, debugging. Esercizio: calcolare la lunghezza di diversi tipi di dato dopo che li si è assegnati a variabili• Iterazione: i cicli e il loro utilizzo; condizionali if/else. Esercizio: eseguire due iterazioni su due oggetti diversi (stringa e lista) e verificare due diverse condizioni if/else.• I file di testo e la loro codifica: come importarli in python e come utilizzarli. Esercizio: calcolare la ricchezza lessicale di un testo• Etivity: Dati tre testi contenuti in diversi file, importarli in Colaboratory e calcolare la loro lunghezza e la loro ricchezza lessicale.



Modulo 2

- Funzioni: definizione e utilizzo. **Esercizio:** definire una funzione che, data una stringa in input, restituisca la ricchezza lessicale in output.
- *NLTK*, tokenizzazione e come utilizzare alcuni metodi del modulo Natural Language Toolkit. Tipo di dato dizionario e calcolo della lista di frequenza di un testo. **Esercizio:** estrarre la lista di frequenza di diversi testi utilizzando una funzione da definirsi.
- Il modulo *requests* e l'accesso alla rete da Colab. Importare testo e codice html. **Esercizio:** data una lista di urls provenienti dal dominio Gutenberg.org definire una funzione che restituisca diverse informazioni sui testi (romanzi o saggi) che queste contengono
- File strutturati csv/tsv: come importarli e come utilizzarli. [Morph-it](#) per la creazione di un lemmatizzatore in Italiano. **Esercizio:** definire una funzione che utilizzando una risorsa lessicale come *Morph-it* restituisca un testo lemmatizzato.
- File strutturati csv/tsv: la risorsa *NRC lexicon* e come utilizzarla per calcolare il profilo emotivo di un testo. **Esercizio:** definire una funzione che dato in input un testo in formato stringa e il nome di un'emozione lo divide in 10 porzioni e calcola la quantità di emozione specificata in input per ogni porzione.
- **Etivity:** da definire

Modulo 3

- La programmazione ad oggetti, parte 1: cos'è una classe, come definirla e come usarla
- La programmazione ad oggetti, parte 2: una classe per il testo e il modulo *requests*. **Esercizio:** eseguire alcune modifiche alla classe appena definita durante la lezione.
- Scraping, parte 1. Il codice sorgente html, come funziona un browser, cosa sono gli elementi html e i loro tags.
- Scraping, parte 2: Il modulo BeautifulSoup e il parsing di una sorgente html. Estrazione di testo da una pagina html. **Esercizio:** modificare la funzione di scraping definita durante la lezione per ottenere informazioni aggiuntive.
- Scraping, parte 3: gestione dell'errore *try/except* e messa a punto di uno scraper per la ricerca di informazioni. **Esercizio:** implementare una nuova funzione per eseguire raccogliere dati di valenza e polarità (sentiment analysis) su recensioni di tripadvisor.
- Scraping, parte 4: realizzazione di un nuovo scraper per la collezione di informazioni strutturate [XML](#). **Esercizio:** estrarre dati clinici da questo file [xml](#)



	<ul style="list-style-type: none">● Etivity: implementare uno scraper su due funzioni: la prima per ottenere una lista di url da cui estrarre informazioni testuali, la seconda per estrarre le informazioni desiderate. Creare un corpus di testi su file. <p>Modulo 4</p> <ul style="list-style-type: none">● Utilizzo di ELAN, software per la trascrizione e l'annotazione del parlato.● Schemi di annotazione e trascrizione: quali aspetti dell'interazione annotare.● Utilizzo di Python per l'estrazione di dati testuali da ELAN con BeautifulSoup.● Regex e creazione di regole per processare il testo.● Il modulo Stanza e il processamento del testo.● Processare un file .eaf (elan) con stanza e BeautifulSoup.● Etivity: scrivere un programma in grado di prendere in input un file .eaf e processarlo usando le risorse NLP viste fino ad ora. <p>Modulo 5</p> <ul style="list-style-type: none">● Altre applicazioni della programmazione e di altre competenze informatiche a problemi relativi a dati di linguaggio e sperimentali.
Materiali di studio	La natura del corso è estremamente pratica. I principali materiali di studio saranno gli script di Python messi a punto durante la lezione e alcuni manuali sulla programmazione di base. Le slides e le dispense sono da intendersi come una sorta di riassunto della lezione, ma per la natura stessa del corso non sono in grado di sostituire l'attività di programmazione vera e propria.
Modalità di verifica dell'apprendimento	<p>L'esame consiste nello svolgimento di un piccolo progetto in Python, che sarà a scelta tra diversi titoli o concordabile con il docente. Il progetto verrà discusso durante l'esame, che sarà accompagnato da una piccola prova scritta di programmazione.</p> <p>I risultati di apprendimento attesi circa le conoscenze della materia e la capacità di applicarle sono valutate dalla prova scritta, dal colloquio orale e in itinere attraverso le Etivity.</p> <p>La prova orale consiste in un colloquio teso ad accertare il livello di preparazione dello studente e la sua competenza rispetto a quanto presentato nel progetto. La prova orale prevede una presentazione di 5 minuti del progetto, seguita da un colloquio di circa 10 minuti.</p> <p>In ambedue le modalità d'esame, particolare attenzione nella valutazione delle risposte viene data alla capacità dello studente di rielaborare, applicare e presentare con proprietà di linguaggio il materiale presente in piattaforma.</p> <p>In sede di valutazione finale, si terrà conto anche della proficua partecipazione ai forum (aule virtuali) e al corretto svolgimento delle e-tivity proposte.</p>



CUNIVERSITÀ CUSANO

Criteria per l'assegnazione dell'elaborato finale	L'assegnazione dell'elaborato finale avverrà sulla base di un colloquio con il docente in cui lo studente manifesterà i propri specifici interessi in relazione all'argomento che intende approfondire.
---	---