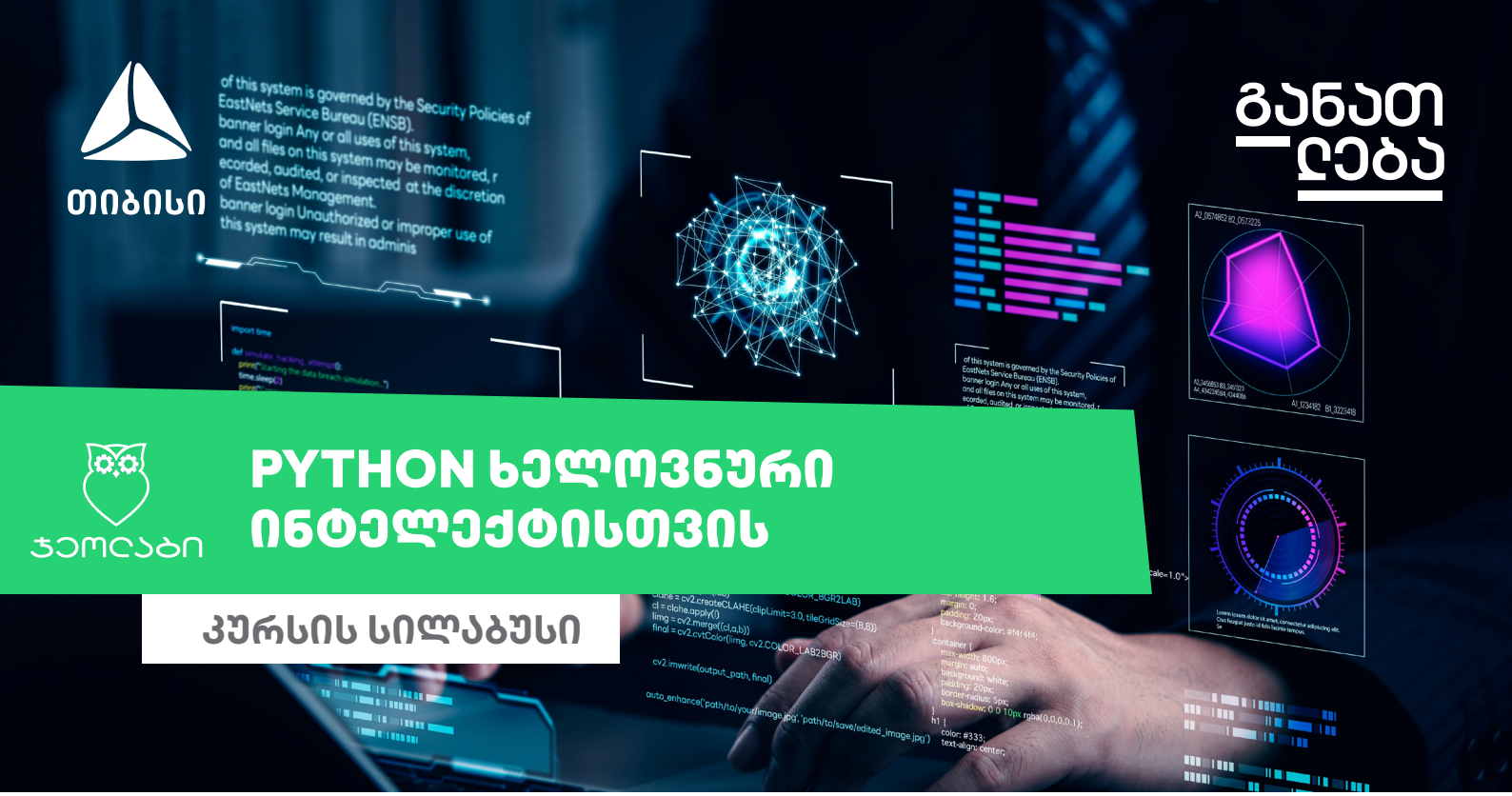
თიბისი

განათლება

ჯეოლაბი

PYTHON ხელოვნური ინტელექტისთვის

კურსის სილაბუსი



სასწავლო კურსის სტატუსი	სასერტიფიკატო კურსი ADVCTBC01
შეხვედრებისა და საათების რაოდენობა	ხანგრძლივობა: 3 თვე შეხვედრების რაოდენობა: 20 ლექცია სალექციო საათები: 40
ლექტორი	ნათია ფორჩხიძე ელფოსტა: natia.porchkhidze@geolab.edu.ge
სასწავლო კურსის ფორმატი	თეორიული და პრაქტიკული ლექციები მოიცავს ინტერაქციულ ლექციებს, ინდივიდუალურ და ჯგუფურ მუშაობას, დისკუსიებსა და პრაქტიკულ სამუშაოებს. კურსი დაყოფილია 4 ბლოკად. თითოეული ბლოკი მოიცავს ერთ შუალედურ პროექტს. კურსის ბოლოს იქნება დამოუკიდებელი სამუშაო პერიოდი ფინალური პროექტისთვის მოსამზადებლად.

#	სასწავლო კურსის შინაარსი	ძირითადი მიგნებები
ბლოკი 1	**PYTHON-ის ძირითადი სინტაქსი და პროგრამირების საფუძვლები**	
ლექცია 1 (2 საათი)	**შესავალი პროგრამირებაში** • პროგრამირების ენის ცნება; Python-ის ადგილი თანამედროვე პროგრამირების ეკოსისტემაში; • განვითარების გარემოს მომზადება: Python-ის ინსტალაცია, IDE/რედაქტორის (მაგ. VS Code, Jupyter) გაცნობა; • ინტერაქტიული რეჟიმი vs სკრიპტის გაშვება • პირველი კოდი: print() ფუნქცია, სინტაქსური წესები, კომენტარების საწყისი ცნება.	
ლექცია 2 (2 საათი)	**ცვლადები და ოპერატორები** • ცვლადის დეკლარირება და მინიჭება, სახელდების კონვენციები; • არითმეტიკული, შედარების, მინიჭების და ლოგიკური ოპერატორები; • ოპერატორთა პრიორიტეტი და გამოსახულებათა შედგენა; • კომენტარები და კოდის დოკუმენტირების საწყისი პრაქტიკა.	
ლექცია 3 (2 საათი)	**მონაცემთა ტიპები I: სტრიქონები და რიცხვითი მონაცემები** • int, float, str — ტიპების თავისებურებები და ურთიერთგადაქცევა (type casting); • სტრიქონული მეთოდები და ინდექსირება/დაჭრა (slicing); • ფორმატირებული სტრიქონები (f-strings).	
ლექცია 4 (2 საათი)	**მონაცემთა ტიპები II: მიმდევრობითი ტიპები (Sequence Types)** • სიები (list): შექმნა, მეთოდები, ცვალებადობის (მუტაბელურობა); • ტუპლები (tuple): უცვლელობის (იმუტაბელურობის) კონცეფცია და გამოყენების შემთხვევები; • დიაპაზონი (range); • List comprehension-ის საწყისი შესავალი.	
ლექცია 5 (2 საათი)	**მონაცემთა ტიპები III: სიმრავლეები, ლექსიკონები და ლოგიკური მონაცემები** • სიმრავლეები (set) და მათი ოპერაციები; • ლექსიკონები (dict): key-value სტრუქტურა, მეთოდები, dict comprehension; • ბულის ტიპი (bool) და ლოგიკური ოპერატორები (and, or, not).	

ლექცია 6 (2 საათი)	**მართვის სტრუქტურები: პირობითი ოპერატორები და ციკლები** • if / elif / else კონსტრუქციები; • for და while ციკლები, break, continue, pass; • ერთდაერთი ხაზის ციკლური კონსტრუქციები (comprehension-ების გაფართოებული გამოყენება); **ქვიზი N1:** Python-ის მონაცემთა ტიპები და მართვის სტრუქტურები.	
ლექცია 7 (2 საათი)	**ფუნქციები, შეცდომების დამუშავება და ფაილებთან მუშაობა** • ფუნქციის განსაზღვრა, პარამეტრები, დაბრუნებული მნიშვნელობები, ნაგულისხმევი არგუმენტები; • ჩაშენებული ფუნქციები და lambda გამოსახულებები; • გამონაკლისების დამუშავება: try / except / finally, საკუთარი გამონაკლისების შექმნა; • ფაილებთან მუშაობის საფუძვლები: ტექსტური ფაილების წაკითხვა/ჩაწერა, CSV და JSON ფორმატები.	
ლექცია 8 (2 საათი)	**პირველი შუალედური პროექტი** პირველი ბლოკის მასალის ინტეგრირებული გამოყენება პრაქტიკულ დავალებაში	
ბლოკი 2	**ობიექტზე ორიენტირებული პროგრამირების საფუძვლები**	
ლექცია 9 (2 საათი)	**OOP-ის საფუძვლები: კლასები, ობიექტები და ინკაფსულაცია** • კლასის და ობიექტის ცნება, კონსტრუქტორი (init), ატრიბუტები და მეთოდები; • ინკაფსულაციის პრინციპი: საჯარო, დაცული და პრივატული ატრიბუტები; • self-ის როლი და მაგისტრალური მეთოდები (str, repr).	
ლექცია 10 (2 საათი)	**მემკვიდრეობითობა** • ბაზისური და წარმოებული კლასები; • მეთოდის გადაფარვა (method overriding) და super()-ის გამოყენება; • მრავალჯერადი მემკვიდრეობითობა - კონცეპტუალური მიმოხილვა.	

ლექცია 11 (2 საათი)	**პოლიმორფიზმი და აბსტრაქცია** • პოლიმორფიზმის პრინციპი და მისი პრაქტიკული გამოვლინებები Python-ში; • აბსტრაქტული კლასები და მეთოდები (abc მოდული); • ბლოკის შეჯამება: OOP-ის ოთხი ბალავარი პრინციპი ერთიან სურათში; **ქვიზი N2: ობიექტზე ორიენტირებული დაპროგრამება**	
ლექცია 12 (2 საათი)	**მეორე შუალედური პროექტი** OOP-ზე დაფუძნებული პრაქტიკული პროექტი (მაგ., მცირე სიმულაცია ან მოდელირების სისტემა კლასების გამოყენებით)	
ბლოკი 3	**PYTHON-ის ბიბლიოთეკებთან მუშაობის საფუძვლები**	
ლექცია 13 (2 საათი	**ბიბლიოთეკები, პაკეტები და გარემოს მართვა** • ბიბლიოთეკის და პაკეტის ცნება; import-ის მექანიზმი; • პაკეტების მენეჯერი pip, ვირტუალური გარემოები (venv) - პრაქტიკული ინსტალაცია და მართვა; • ჩაშენებული ბიბლიოთეკები: random, math.	
ლექცია 14 (2 საათი)	**შესავალი NumPy-ში** • მასივები (ndarray) და მათი უპირატესობა სიებთან შედარებით; • ინდექსირება, ფორმის ცვლილება (reshape), ვექტორული ოპერაციები; • ბროუდკასტინგის საწყისი ცნება.	
ლექცია 15 (2 საათი)	**შესავალი Pandas-ში** • Series და DataFrame სტრუქტურები; • მონაცემთა ჩატვირთვა (CSV/Excel), ფილტრაცია, დაჯგუფება (groupby); • მონაცემების საწყისი დამუშავება.	

ლექცია 16 (2 საათი)	**მონაცემთა ვიზუალიზაცია — Matplotlib** • pyplot ინტერფეისი: ხაზოვანი, სვეტოვანი და scatter გრაფიკები; • გრაფიკის მორგება: ღერძები, ლეგენდა, სათაურები; • მრავალგრაფიკული განლაგება (subplots).	
ლექცია 17 (2 საათი)	**აღწერითი სტატისტიკის საფუძვლები** • ცენტრალური ტენდენციისა და გაბნევის საზომები (საშუალო, მედიანა, სტანდარტული გადახრა); • მონაცემთა წინასწარი დამუშავება: ნორმალიზაცია, სტანდარტიზაცია, აუტლაიერების; • საწყისი გამოვლენა; • Pandas/NumPy-ის გამოყენება სტატისტიკური მაჩვენებლების გამოსათვლელად.	
ლექცია 18 (2 საათი)	**გარე მონაცემებთან/API-სთან მუშაობის საფუძვლები** • HTTP მოთხოვნების საწყისი ლოგიკა, requests ბიბლიოთეკის გამოყენება; • JSON პასუხების დამუშავება და გარე API-დან მონაცემის მიღება (მათ შორის AI API-ების მაგალითზე).	
ლექცია 19 (2 საათი)	**მესამე შუალედური პროექტი** მონაცემთა ანალიზისა და ვიზუალიზაციის კომპლექსური პროექტი, რომელიც აერთიანებს NumPy, Pandas და Matplotlib-ს	
ბლოკი 4	**ფინალურ პროექტზე მუშაობა და ლექტორთან უკუკავშირი**	
ლექცია 20 (2 საათი)	**ინდივიდუალური/ჯგუფური კონსულტაციები** • ფინალური პროექტის თემის შერჩევა და დაგეგმვა; • კოდის რევიზია, დახვეწა და საკურსო მოთხოვნებთან შესაბამისობის შემოწმება.	