



სასწავლო კურსის პროგრამა ვებ აპლიკაციის შექმნა Python-ის გამოყენებით

სასწავლო კურსის ხანგრძლივობა: 80 ლექცია

განრიგი: აუდიტორიული / Online სწავლების 2 საათი ორჯერ კვირაში.

თემატური გეგმა

1. ვებ-გვერდის შექმნა HTML5-ის და CSS3-ის გამოყენებით. 14 ლექცია
2. კლიენტის სცენარების შემუშავება JavaScript-ის გამოყენებით . . . 18 ლექცია
3. Web-აპლიკაციის შექმნა Python-ის გამოყენებით 48 ლექცია

WEB

HTML5 / CSS3

(14 ლექცია)

ლექცია 1

1. შესავალი
2. HTML/CSS/JS ზოგადი მიმოხილვა
3. სადველოპმენტო გარემოს მომზადება (vscode + extensions)
4. ჩაშენებული snippets, emmet, shortcuts
5. HTML ჩონჩხი (! snippet)
 - doctype, html, head, meta, title, body
6. HTML-ის ძირითადი თეგები
 - p, h1-h6, span, l, b, br, img, a, button

ლექცია 2

1. ერთმანეთში ჩაშენებული თეგები (nested)
2. სიები
 - ul, ol, li
3. ცხრილები

- table, thead, tbody, tr, th, td
4. CSS-ის მიმოხილვა, HTML-ში შემოტანის სამივე მეთოდის განხილვა
 - inline, internal, external
 5. CSS სელექტორები, იერარქია
 6. CSS-ის ზოგიერთი თვისებები
 - font-size, font-weight, color, background-color

ლექცია 3

1. ცხრილების გამეორება, პრაქტიკაში გამოყენება
2. ჩარჩოების განხილვა
 - border, border-radius, border-collapse
3. ტექსტების მართვა CSS-ის საშუალებით
 - font-style, text-align, text-decoration
4. inline/inline-block/block, განსხვავებები მათ შორის და მათი გამოყენება
5. height/width
6. box-model (margin/padding)
7. float
8. ზომის ერთეულები CSS-ში
 - px, %

ლექცია 4

1. id, class და მათი გამოყენება
2. div
3. SEO-ს მიმოხილვა
4. სემანტიკური ელემენტები
 - header, nav, main, section, aside, article, footer
5. ფორმების მიმოხილვა
6. ფორმის თეგები
 - label, Input, select, option, textarea
7. Input ტიპები
 - text, number, checkbox, radio, color, mail, file, password, date, range
8. ატრიბუტები
 - value, name, disabled, placeholder, checked, required, selected
9. მედია/აუდიო თეგები და მათი გამოყენება
10. Iframe + google map + youtube embedded

ლექცია 5

1. icons და მათი გამოყენება
2. fontawesome გამოყენება (cdn-ის გამოყენება)
3. svg და მისი გამოყენება
4. Icon font-სა და Svg შორის განსხვავება

5. ფლექსბოქსები და მათი გამოყენება

- display:flex/flex-direction/flex-wrap/justify-content/align-items/align-content/order/align-self/ row-gap, column-gap, gap
-

6. flexboxfroggy.com საიტის განხილვა

ლექცია 6

1. ანიმაციები

2. ფსევდო კლასები და hover გამოყენება

3. ანიმაციის გამოყენება

- animation/name/duration/delay/iteration-count/timing-function
- keyframe
- cubic-bazier მცირედი ახსნა

4. სპინერის აწყობა და ლოდერის პრინციპი

5. transform-ის შესახებ

- rotate, scale

6. transition განხილვა და გამოყენება

ლექცია 7

1. პოზიციონება

- static, absolute, relative, fixed, sticky

2. პოზიციონების კონტექსტი

- top, right, bottom, left

3. z-index

4. opacity და მისი გამოყენება

5. GitHub

- რეპოზიტორია
- ბრენჩი
- bash / vscode-ში Git-ის ინტეგრაცია / github desktop application (სურვილისამებრ)
- add, commit, push, merge
- დაჰოსტვა – github pages

ლექცია 8

1. PSEUDO-CLASS , მაგალითები

- hover, visited, active, focus და სხვა

2. კომბინატორები , SPACE + > + ~

3. PSEUDO –ELEMENT

4. ::BEFORE, ::AFTER , CONTENT,

5. სელექტორები ატრიბუტის გამოყენებით

6. PSEUDO – CLASSES

7. OBJECT-FIT , თვისებები COVER , CONTAIN, OBJECT - POSITION

ლექცია 9

1. გრიდები
 - a. display: grid
 - b. grid-tracks : grid-template-columns, grid-template-row, grid-auto-rows, grid-auto-columns
 - c. fr, repeat, minmax
 - d. place-items
 - e. row-gap, column-gap, gap
2. გრიდის ელემენტების განლაგება ხაზების მიხედვით:
 - a. grid-column-start, grid-column-end, grid-row-start, grid-row-end
3. გრიდის ელემენტების განლაგება არეალის მიხედვით:
 - a. grid-area
4. საიტის განლაგების აწყობა გრიდის გამოყენებით

ლექცია 10

1. BOOTSTRAP - მიმოხილვა, უპირატესობა მის ანალოგებთან (tailwind.css, bulma)
2. BOOTSTRAP-ის შემოტანის გზების განხილვა , CDN / LOCAL FILE , CDN უპირატესობა ლოკალურ ფაილთან შედარებით.
3. ვერსიების განხილვა.
4. კომპონენტების ზოგადი მიმოხილვა
5. რამდენიმე ძირითადი კომპონენტების შემოტანა თვალსაჩინოებისთვის
6. სელექტორების კონფლიქტი და უპირატესობა
7. BOOTSTRAP კლასებთან მუშაობის სტანდარტი
8. დამორეზების კლასები , ტექსტებთან სამუშაო კლასების განხილვა

ლექცია 11

1. UTILITIES - ძირითადი კლასები განხილვა ,
2. BORDERS, COLORS, DISPLAY
3. POSITION კლასები
4. SHADOW კლასები
5. HELPERS - ძირითადი კლასები
6. POSITION - კლასები
7. LAYOUT - ყველა ძირითადი კლასის განხილვა , CONTAINER , ROW , CONTAINER-FLUID
8. BREAKPOINTS – SM , MD, LG, XL , XXL ,
9. BREAKPOINTS გამოყენება პრაქტიკაში , მათი დანიშნულება RESPONSIVE ვებ დიზაინში
10. BREAKPOINTS - გამოყენება , GRID სისტემის შიგნით
11. განლაგებები GRID სისტემის შიგნით , justify-content, align-items
12. BREAKPOINTS გამოყენება utilities კლასების შიგნით
13. პრაქტიკული სამუშაო BOOTSTRAP გამოყენებით RESPONSIVE საიტის მარქაფის შექმნა

ლექცია 12

1. CSS უნიტები
 - px/rem/em/ch/vh/vw/%
2. რესპონსიული დიზაინი
3. მედია query
 - min-width,max-width
4. min-width,max-width,min-height,max-height თვისებები
5. overflow
6. ცვლადები
7. calc
8. რესპონსიული ჰედერის გაკეთება js გარეშე
9. შუალედური პროექტის (ვადა მე-15 ლექციამდე)
10. figma-ს ზოგადი მიმოხილვა (რესურსები: <https://www.figma.com/community/tag/web>)

ლექცია 13

1. რა არის SCSS ზოგადი მიმოხილვა, გამოყენება თანამედროვე ვებ დეველოპმენტში
2. NODE.JS ინსტალაცია
3. NPM განხილვა
4. NPM გამოყენებით SASS ინსტალაცია
5. SCSS - სინტაქსი
6. PARENT-SELECTOR, @EXTEND , VIRIABLES , ფსევდო კლასების/ელემენტის გამოყენება, @MIXIN, @MIXIN პარამეტრები , @INCLUDE, @USE

ლექცია 14

1. პროექტის განხილვა (ჰაფინესების შეფასება)
2. პროექტის განხილვა (ლექტორების შეფასება)
3. შიდა განხილვა ჯგუფში

JavaScript

(18 ლექცია)

ლექცია 15

1. JS-ის მიმოხილვა
2. ცვლადების ინიციალიზების მეთოდები (let/var/const) და განსხვავება მათ შორის
3. ცვლადების ტიპები
(char/string/number/boolean/undefined/NaN/null/array/object/function)
4. მოქმედებები რიცხვებზე + - / * %
5. შედარებების ოპერატორები ==, ===, >=, <=, > <
6. კონსოლი
 - console.log(), console.warn(), console.error()
7. typeof – მონაცემების ტიპის გადამოწმება
8. კომენტარები

ლექცია 16

1. დიალოგური ფანჯრები prompt, confirm, alert
2. document.write-საწყისი მიმოხილვა
3. Number() - კონვერტირებისთვის
4. ლოგიკური ოპერატორები
5. Nested ლოგიკური ოპერატორები
6. Switch

ლექცია 17

1. მათემატიკური ობიექტის განხილვა
 - Math.pow, Math.sqrt, Math.abs Math.random(), Math.round(), Math.floor(), Math.ceil()
2. სტრინგზე მოქმედებები
3. სტრინგის მეთოდები
 - search, slice, substring, replace, length, toUpperCase, toLowerCase, concat, trim, trimLeft, trimRight, includes
4. ლოგიკური ოპერატორები (AND/OR/NOT კონიუქცია/დიზიუნქცია/ნეგაცია)

ლექცია 18

1. ლუპი - while და for
2. do while და for of ლუპები
3. პრაქტიკა ლუპზე

ლექცია 19

1. მასივები
2. მასივის მეთოდები
 - length, pop, shift, unshift, push, includes
3. მასივის გამოყენება ლუბთან ერთად
4. დავალებების შესრულება ლუბის საშუალებით

ლექცია 20

1. მასივის მეთოდები
 - concat, join, slice, splice
2. შსავალი ალგორითმებში
 - სიტყვიერი ჩანაწერი/ბლოქსქემა/ფსევდო კოდი
3. bubble sort, ვიდეოთი ასხნა, ფსევდო კოდით, კოდის დაწერა

ლექცია 21

1. სტრინგის დამატებითი მეთოდები
 - split, charAt
2. მატემატიკური ფუნქციები
 - parseInt, parseFloat, isNaN, isFinite
3. pattern დახატვა ფორით (პირამიდის დახატვა)
 - <https://www.101computing.net/wp/wp-content/uploads/Python-Patterns.png>
4. ორ განზომილებიანი მასივები (მატრიცა)

ლექცია 22

1. სამომხარებლო ფუნქციები (function)
2. void/return
3. რენდომული ფერის გენერირება rgb,rgba
4. რეკურსიის ასხნა
5. რეკურსიის მაგალითები. fibonacci

ლექცია 23

1. ანონიმური ფუნქციები (callbacks)
2. arrow ფუნქციები
3. მასივების მეთოდების ხელით აწყობა
 - filter, every, some
4. ჩაშენებული მასივის მეთოდები
 - filter, every, some, forEach, map

ლექცია 24

1. მასივის მეთოდები
 - copyWithin, indexOf, fill, find, findIndex, reverse, reduce, toString
2. ternary operator
3. date-ს მიმოხილვა
 - GETFULLYEAR(), GETMONTH(), GETDATE(), GETHOURS(), GETMINUTES(), GETSECONDS()
 - SETFULLYEAR(), SETMONTH(), SETDATE(), SETHOURS(), SETMINUTES(), SETSECONDS()
 - toLocaleDateString(), toLocaleString(), toLocaleTimeString()
4. toString() მეთოდი

ლექცია 25

1. ობიექტები
2. for in ლუპი
3. bubble sort გამოყენება ობიექტებზე / array.sort

ლექცია 26

1. შესავალი DOM-ში
2. ელემენტების სელექტორები
 - querySelector, querySelectorAll, getElementsByTagName, getElementById, getElementsByClassName
3. ივენტები
 - addEventListener გამოყენება
4. კონტენტის ცვლილება
 - textContent InnerText InnerHtml
5. ვიზუალურად tic tac toe აწყობა (მხოლოდ X და O დამატება)
6. მოქმედი increment/decrement აწყობა

ლექცია 27

1. todo აპლიკაციის შემუშავება
2. input ველებთან მუშაობა
3. sweetAlert2 განხილვა
4. image upload | base64 –ის შესახებ
 - FileReader

ლექცია 28

1. input ველთან მუშაობა keypress keyup ივენტთან ერთად
2. Regex - search(), replace(), მოდიფიკატორები (l, g, m), ნიმუშები, metacharacter

ლექცია 29

1. JSON/XML ახსნა
2. storage ების მიმოხილვა და გამოყენება პრაქტიკაში
 - localStorage/sessionStorage/cookie
3. API ქოლების შესავალი
 - GET/POST/PUT/PATCH/DELETE
 - რესურსი GET ისთვის <https://github.com/public-apis/public-apis>
4. XML გაგზავნა - new XMLHttpRequest();

ლექცია 30

1. პრომისები
2. async/await
3. fetch

ლექცია 31

1. კლასები
 - constructor, new
2. OOP
3. ინკაპსულაცია
 - public, private, protected
4. მემკვიდრეობითობა
 - Extends

ლექცია 32

1. პრაქტიკა API ქოლებზე
2. შუალედური პროექტის განხილვა

Web-აპლიკაციის შექმნა Python-ის გამოყენებით

სასწავლო კურსის ხანგრძლივობა: 48 ლექცია

კურსის მიზანი

მსმენელმა შეისწავლოს web-აპლიკაციის შექმნა Python-ის გამოყენებით.

სასწავლო კურსის დასრულებისას მსმენელს ეცოდინება შემდეგი:

- ეცოდინება Python-ის გამოყენებით ვებ-აპლიკაციების შემუშავების სირთულეებში გარკვევა;
- გაერკვეული იქნება ოპ-ის მექანიზმების რეალიზაციის თავისებურებებში Python-ში;
- ეცოდინება ფორმების მონაცემთა დამუშავება და ანალიზი;
- ეცოდინება Python-ის სტანდარტული შესაძლებლობის გამოყენება;
- ეცოდინება რეგულარული გამონათქვამების გამოყენება;
- გაცნობიერებული იქნება ფუნქციონალური პროგრამირების პრინციპები;
- შეეძლება cookies ფაილებში მომხმარებლის მონაცემების შენახვა;
- სესიების მექანიზმებთან მუშაობა;
- ქსელური ურთიერთქმედების პრინციპების გაცნობიერება;
- მონაცემთა წყაროებთან ურთიერთობა;
- ვებ-აპლიკაციაში AJAX-ის ჩანერგვა;
- შეეძლოს ვერსიის მართვის სისტემის გამოყენება;
- ესმოდეს ბრძანებითი ურთიერთქმედების საფუძვლები;
- შეეძლოს გამოიყენოს პროექტირების პატერნები;
- გამოიყენოს უნიტის ტესტირება;
- ესმოდეს და ერკვეოდეს MVC პატერნის სირთულეებში;
- შეეძლოს შექმნას ვებ-პროექტები Python-ის და MVC პატერნის გამოყენებით;
- Flask/Bottle ის გამოყენება;
- Django ჩარჩოს დახმარებით ვებ-აპლიკაციების შემუშავება.

სასწავლო კურსის დასრულებისას სტუდენტმა უნდა ჩააბაროს პრაქტიკული დავალება და თეორიული გამოცდა კურსის მასალის შესაბამისად. გამოცდაზე დასაშვებად სტუდენტს ჩაბარებული უნდა ჰქონდეს ყველა საშინაო და პრაქტიკული გამოცდა. პრაქტიკული დავალება უნდა მაქსიმალურად მოიცავდეს მასალას სასწავლო კურსის სხვადასხვა ნაწილებიდან.

თემატური გეგმა

მოდული 1. Python-ზე Web-პროგრამირების შესავალი.	1 ლექცია
მოდული 2. პირობითი ოპერატორები, ციკლები და გამონაკლისები	2 ლექცია
მოდული 3. სტრიქონები, სიები.	2 ლექცია
მოდული 4. ფუნქციები.	3 ლექცია
მოდული 5. სორტირება, ძიება.	2 ლექცია
მოდული 6. ტუპლები, სიმრავლეები, ლექსიკონები.	2 ლექცია
მოდული 7. ფაილები.	2 ლექცია
მოდული 8. ვერსიების მართვის სისტემა.	1 ლექცია
მოდული 9. ოპ.	3 ლექცია
მოდული 10. მონაცემთა სტრუქტურა.	2 ლექცია
მოდული 11. მონაცემთა შეფუთვა.	2 ლექცია
მოდული 12. პატერნი MVC.	2 ლექცია
მოდული 13. SOLID პროექტირების პრინციპები.	1 ლექცია
მოდული 14. მოდულარული ტესტირება.	1 ლექცია
მოდული 15. პარალელური, მრავალგანშტოებიანი და ქსელური პროგრამირება.	3 ლექცია
მოდული 16. შესავალი მონაცემთა ბაზებთან მუშაობაში.	3 ლექცია
მოდული 17 Python-ში მონაცემთა ბაზების გამოყენება.	2 ლექცია
მოდული 18. ფრეიმვორქები.	12 ლექცია
მოდული 19. Python-ის საშუალებით ჩატ-ბოტების შექმნა.	2 ლექცია

მოდული 1

(1 ლექცია)

Python-ზე Web-პროგრამირების შესავალი.

ლექცია 1

Web-პროგრამირებაში შესავალი და web-აპლიკაციების მუშაობის პრინციპები.

- ტექნოლოგია კლიენტ-სერვერის საფუძვლები
- HTTP პროტოკოლის მიმოხვლა, Request და Response სტრუქტურასთან გაცნობა.
- Web-პროგრამირების საფუძველი და მისი წარმოშობა. სერვერული web-პროგრამირების განსხვავება კლიენტის ტიპისგან. მიზნები, ამოცანები, განვითარების მიმართულება, მოკლე ისტორია.
- Web-გვერდის ჩატვირთვის პრინციპები და ეტაპები.

პროგრამირების ენების მიმოხილვა.

- პროგრამირების მთავარ პარადიგმებთან გაცნობა.
- პროგრამირების თანამედროვე ენების მიმოხილვა.
- ალგორითმის კონცეფცია.
- Python-ის ენასთან გაცნობა, გამოყენების სფეროები.
- Python-ში შესავალი. Python-ის ინტერპრეტატორი და მისი გარემოცვა.

Python-ში შესავალი.

- ინტერპრეტატორის კონცეფცია და დაყენების წყობა.
- გაცნობა პროგრამირების გარემოებებთან:
 - IDLE და Python Shell -ის პროგრამირების სტანდარტული პაკეტი;
 - IDE PyCharm, Spyder, Visual Studio;
 - Atom.

მონაცემთა ტიპები, ცვლადები და სინტაქსური კონსტრუქციები.

- ტიპი და მნიშვნელობა.
- Python-ის ენაში ცვლადები, როგორც ობიექტი.
- ცვლადების სახელები და რეზერვირებული სიტყვები.
- კონსტრუქციები.
- ოპერატორები და ოპერანდები.
- ოპერატორთა პრიორიტეტები.

- ოპერაციები ცვლადებზე.
- პროგრამის შესრულების წყობა.
- შესავალი/გასავალი.
- ტიპების გარდაქმნა.
- სინტაქსური და ლოგიკური შეცდომები, მათთან მუშაობა.

მოდული 2

(2 ლექცია)

პირობითი ოპერატორები, ციკლები

ლექცია 2

პირობითი ინსტრუქციები და მათი სინტაქსი.

- „ბლოკის“ შესრულება.
- ლოგიკური გამონათქვამები და ოპერატორები.
- განშტოების ოპერატორები if ... else.
- ჩერთმანეთში ჩასმული კონსტრუქციები.

ლექცია 3

ციკლები.

- იტერაციის კონცეფცია.
- ციკლი while.
- უსასრულო ციკლები.
- მმართველი ოპერატორები continue, break და else.
- ციკლი for.
- ადგილობრივი და გლობალური ცვლადები.

მოდული 3

2 ლექცია

სტრიქონები, სიები

ლექცია 4

სტრიქონები.

- კოდირება ASCII, Unicode, UTF-8, Byte-code.
- სტრიქონი – სიმბოლოების უცვლელი წყობა.
- სტრიქონების მეთოდები.
- სტრიქონებთან მუშაობის თავისებურებები.
- სტრიქონის დაჭრა.
- მოდული string.
- ბაიტები და კოდირებები.
- რეგულარული გამონათქვამები.

ლექცია 5

სიები.

- კლასიკური მასივის კონცეფცია.
- ობიექტების კოლექციის კონცეფცია.
- ბმულისებრი მონაცემთა ტიპი list.
- სიების შექმნა.
- სიების გენერატორები.
- სიების მეთოდები.
- მიკუთვნების ოპერატორი in.
- სიების თავისებურებები, ბმულები და კლონირება.
- ელემენტის ძიება.
- მატრიცები.

მოდული 4

3 ლექცია

ფუნქციები

ლექცია 6

ფუნქციები და მოდულები.

- რა არის ფუნქცია?
- ფუნქციის მიზნები და ამოცანები.
- ჩაშენებული ფუნქციები.
- მათემატიკური ფუნქციები და შემთხვევითი რიცხვები.
- ფუნქციის გამოცხადების სინტაქსი.
- არგუმენტები და დაბრუნებული მნიშვნელობები.

ლექცია 7

ფუნქციებთან მუშაობის დამატებითი მეთოდები.

- არგუმენტების გახსნა და შეფუთვა.
- დეფოლტ არგუმენტები, Key-არგუმენტები.
- ხედვის სფერო, LEGB-ის წესი.
- ლოკალური და გლობალური ცვლადები ფუნქციებში.
- ფუნქცია როგორც პირველი კლასის ობიექტი.
- რეკურსია.

ლექცია 8

ფუნქციონალური პროგრამირება.

- რა არის ფუნქციონალური პროგრამირება?
- ანონიმური ფუნქციები lambda.
- მოდული functools.
- ფუნქციები map(), reduce(), filter(), zip().
- უმაღლესი რანგის ფუნქციები.

მოკლე ჩართვა (closure).

კარინგი.

დეკორატორები.

გამონაკლისის კონცეფცია.

- გამონაკლისის მცნებები.
- გამონაკლისის გადაკვეთა.
- try ... except-თან მუშაობის თავისებურებები.

მოდული 5

2 ლექცია

სორტირება, ძიება

ლექცია 9

სორტირება.

- ოპტიმალურობა.
- სორტირება ბუშტუკით.
- სორტირება შერწყმით.
- შელის სორტირება.
- პირამიდული სორტირება.
- სწრაფი სორტირება.

ლექცია 10

ძიება.

- ხაზობრივი ძიება.
- ბინარული ძიება.

მოდული 6

2 ლექცია

ტუპლები, სიმრავლეები, ლექსიკონები

ლექცია 11

ტუპლები.

- უცვლელი ობიექტების კოლექცია.
- ტუპლების გამოყენება და თავისებურებები.

სიმრავლეები.

- სიმრავლეების მათემატიკური კონცეფცია.
- მონაცემთა ტიპი set(), frozenset().
- ოპერაციები სიმრავლეებზე.
- სიმრავლეების გამოყენება.

ლექცია 12

ლექსიკონები.

- ასოციაციური მასივები.
- ჰეშ-ცხრილები
- ლექსიკონის შექმნა.
- ლექსიკონის მეთოდები.
- ერთმანეთში ჩაღაგებული ლექსიკონები

გამოყენების პრაქტიკული მაგალითები.

მოდული 7

2 ლექცია

ფაილები

ლექცია 13

ფაილები.

- ფაილების სისტემა, ფორმატების რეალიზირების თავისებურებები.
- მუშაობა ფაილებთან:
 - გახსნა;
 - დახურვა;
 - კითხვა;
 - ჩაწერა.

კონტექსტის მენეჯერი.

ლექცია 14

ფაილთა ტიპები, ტექსტური და ბინარული.

გამოყენების პრაქტიკული მაგალითები.

მოდული 8

1 ლექცია

ვერსიის მართვის სისტემა

ლექცია 15

რა არის ვერსიის მართვა?

რისთვის არის საჭირო ვერსიის მართვა.

ვერსიის მართვის სისტემების მიმოხილვა.

- CVS.
- SVN.
- Git.

- ვერსიის მართვის სხვა სისტემები\ი.

Git.

- რა არის Git?
- Git-ის მიზნები და ამოცანები.
- მთავარი ტერმინები:
 - რეპოზიტორიები;
 - კომიტი;
 - განშტოება (Fork);
 - სამუშაო კატალოგი.
- Git-თან ოპერაციები:
 - დაყენება;
 - რეპოზიტორიის შექმნა;
 - რეპოზიტორში ფაილის დამატება;
 - რეპოზიტორში კომიტის ჩაწერა;
 - სამუშაო კატალოგის მიმდინარე მდგომარეობის მიღება;
 - განშტოებების (Fork) ასახვა;
 - დაგროვებად ბუფერთან მუშაობა;
 - git remote;
 - git push;
 - git pull;
 - სხვა ოპერაციები.

გარე სერვისების გამოყენება (github).

მოდული 9

3 ლექცია

ოოპ

ლექცია 16

შესავალი ოოპ-ში.

- ოოპ-ის კონცეფცია.
- ინკაპსულაცია.

- მემკვიდრეობა.
- პოლიმორფიზმი.
- Python-ში ობ-ის რეალიზირების თავისებურებები, „იხვისებრი ტიპიზაცია“ (Duck typing).

ლექცია 17

მონაცემთა ტიპები, განსაზღვრული მომხმარებლის მიერ.

- კლასის ეგზემპლარი.
- კლასები და ობიექტები.
- ველები (თვისებები), კლასის მეთოდები.
- მეთოდების გადატვირთვა.
- Magic-მეთოდები, კონსტრუქტორები.
- სტატიკური მეთოდები და კლასის მეთოდები.

მემკვიდრეობა და ინკაპსულაცია.

- public, protected და private მეთოდები.
- მრავლობითი მემკვიდრეობა და MRO (მეთოდების დაშვების წყობა).

პოლიმორფიზმი.

- ოპერატორების გადატვირთვა.
- Magic-მეთოდების რეალიზირება.

ლექცია 18

კლასთა ეგზემპლარების შექმნა და მართვა.

- ფუნქტორები.
- დეკორატორები.
- მართვადი ატრიბუტები.
- თვისებები.
- დესკრიპტორები.

მეტაკლასები.

- მეტაკლასების მოდელები.
- მეთოდი კონსტრუქტორი `__new__()`.
- ინსტრუქციის პროტოკოლი `class`.

მოდული 10

2 ლექცია

მონაცემთა სტრუქტურები

ლექცია 19

დაკავშირებული სიები.

- რა არის სია?
- ერთმანეთს და ორმანეთს დაკავშირებული სია.
- გამოყენების პრაქტიკული მაგალითები.

სტეკი.

- რა არის სტეკი?
- LIFO-ს პრინციპი.
- გამოყენების პრაქტიკული მაგალითები.

ლექცია 20

რიგი.

- რა არის რიგი?
- რიგების სახეობები
- გამოყენების პრაქტიკული მაგალითები.

ხეები.

- რა არის ხე?
- ხეთა სახეობები
- გამოყენების პრაქტიკული მაგალითები.

მოდული 11

2 ლექცია

მონაცემთა შეფუთვა

ლექცია 21

სერიალიზაცია და დესერიალიზაცია.

- რა არის სერიალიზაცია?
- რა არის დესერიალიზაცია?
- სერიალიზაციის და დესერიალიზაციის მიზნები და ამოცანები.
- გამოყენების პრაქტიკული მაგალითები.

ლექცია 22

მოდული pickle.

მოდული json.

სერიალიზაციის გარე მოდულები.

მოდული 12

2 ლექცია

პატერნი MVC

ლექცია 23

რა არის პროექტირების პატერნები.

რა არის პატერნი MVC?

პატერნი Model-View-Controller-ის მიზნები და ამოცანები.

Model.

- რა არის Model?
- Model-ის მიზნები და ამოცანები.

ლექცია 24

View.

- რა არის View?
- View-ის მიზნები და ამოცანები.

Controller.

- რა არის Controller?
- Controller-ის მიზნები და ამოცანები.

პატერნი MVC-ის გამოყენების მაგალითები.

მოდული 13

1 ლექცია

SOLID პროექტირების პრინციპები

ლექცია 25

პრობლემათა მიმოხილვა, რომლებიც გვხდება პროექტირებისას და შემუშავებისას.

SOLID კლასების პროექტირების პრინციპები.

- ერთეული პასუხისმგებლობის პრინციპი (The Single Responsibility Principle).
- ღია/დახურული პრინციპი (The Open Closed Principle).
- ბარბარა ლისკოვის ჩანაცვლების პრინციპი (The Liskov Substitution Principle).
- ინტერფეისის სეგრეგირების პრინციპი (The Interface Segregation Principle).
- დამოკიდებულების ინვერსიის პრინციპი (The Dependency Inversion Principle).

SOLID პრინციპების გამოყენების მაგალითები.

მოდული 14

1 ლექცია

მოდულარული ტესტირება

ლექცია 26

რა არის მოდულარული ტესტირება?

მოდულარული ტესტირების მიზნები და ამოცანები.

მოდულარული ტესტირების აუცილებლობა.

მოდულარული ტესტირების ინსტრუმენტების მიმოხილვა.

Python-აპლიკაციისთვის მოდულარული ტესტირების ინსტრუმენტი.

მოდული 15

3 ლექცია

პარალელური, მრავალგანშტოებიანი და ქსელური პროგრამირება

ლექცია 27

პარალელური და მრავალგანშტოებიანი პროგრამირება.

- დინების (Thread) შექმნა.
- დინების (Thread) სინქრონიზაცია.
- ამოცანების რიგები (Task queue).
- GIL და Python-ში მრავალგანშტოების რეალიზაციის თავისებურებები.
- პროცესები და პროცესებს შორის მონაცემთა გადაცემა.

ლექცია 28

ქსელური პროგრამირება.

- პროტოკოლი HTTP/HTTPS.
- მოდელი OSI, tcp/udp.
- კლიენტ-სერვერის მოდელი.

ლექცია 29

- მრავალგანშტოებადი და ასინქრონული ექო-სერვერის რეალიზაცია.
- Apache და Nginx.

მოდული 16

3 ლექცია

შესავალი მონაცემთა ბაზებთან მუშაობაში

ლექცია 30

შესავალი მონაცემთა ბაზების თეორიაში.

- ისტორია და განვითარების ეტაპები.
- მონაცემთა ბაზების კონცეფცია და მონაცემთა ბაზების მართვის სისტემა.
- არსებული მონაცემთა ბაზების მოდელების შედარება:
 - ფაილების მოდელი;
 - ქსელური მოდელი;
 - იერაქრიული მოდელი;
 - რელაციური მოდელი;
 - ობიექტ-ორიენტირებული მოდელი.
- რელაციური მონაცემთა ბაზების კონცეფცია.
- მონაცემთა ბაზის მართვის სისტემა MySQL:
 - რა არის MySQL;
 - MySQL-ის განვითარების ისტორია;
 - MySQL-ის ვერსიები;
 - MySQL-ის ინსტალაცია.
- ცხრილები:

- პირველადი გასაღები (primary key);
- მნიშვნელობა დეფოლტად;
- უნიკალურობა.

■ მონაცემთა ტიპები.

■ ინდექსები.

■ რეკუესტები (Requests):

- SQL-ის ენის შესავალი;
- SQL ენის სტანდარტები;
- კონცეფციები DDL, DML, DCL.

მოთხოვნები SELECT, INSERT, UPDATE, DELETE.

■ ოპერატორი SELECT:

- ოპერატორი FROM;
- ოპერატორი WHERE;
- ოპერატორი ORDER BY.

■ საკვანძო სიტყვა IN, BETWEEN, LIKE.

■ ოპერატორი INSERT.

■ ოპერატორი

UPDATE.

■ ოპერატორი DELETE.

ლექცია 31

მრავალცხრილოვანი მონაცემთა ბაზები.

■ ერთ-ცხრილოვან მონაცემთა ბაზებთან ურთერთქმედების ანომალიები:

- განახლების ანომალიები;
- ჩასმის ანომალიები;
- წაშლის ანომალიები.

■ მრავალცხრილოვანი მონაცემთა ბაზების პრინციპები:

- მრავალცხრილოვანი მონაცემთა ბაზების შექმნის მიზეზები;
- გარე გასაღები (Foreign key);
- კავშირები. კავშირების ტიპები;
- მონაცემთა მთლიანობა.

ნორმალიზაცია.

- ნორმალიზების აუცილებლობა.
- ნორმალური ფორმის კონცეფცია.

- პირველი ნორმალური ფორმა.
- მეორე ნორმალური ფორმა.
- მესამე ნორმალური ფორმა.
- ბოის-კოდის ნორმალური ფორმა.

მრავალრიცხოვანი მოთხოვნები.

- მრავალრიცხოვანი მოთხოვნის შექმნის პრინციპები.
- CROSS JOIN.

ლექცია 32

აგრეგირების ფუნქცია.

- ფუნქცია COUNT.
- ფუნქცია AVG.
- ფუნქცია SUM.
- ფუნქცია MIN.
- ფუნქცია MAX.

დაჯგუფების კონცეფცია. საკვანძო სიტყვა GROUP BY.

საკვანძო სიტყვა HAVING. შედარებითი ანალიზი HAVING და WHERE.

ქვემოთხოვნები (Subqueries).

- Subqueries შექმნის და გამოყენების აუცილებლობა.
- Subqueries და მრავალცხრილოვანი მოთხოვნების შედარება.
- Subqueries მუშაობის პრინციპი.

ოპერატორები ქვემოთხოვნებში გამოყენებისთვის, გაერთიანება.

- ოპერატორები Subqueries გამოყენებისთვის:
 - ოპერატორი EXISTS;
 - ოპერატორი ANY/SOME;
 - ოპერატორი ALL.
- queries შედეგების გაერთიანება:
 - გაერთიანების პრინციპები;
 - საკვანძო სიტყვა UNION;
 - საკვანძო სიტყვა UNION ALL.
- გაერთიანება JOIN:
 - კონცეფცია inner join;
 - კონცეფცია left join;
 - კონცეფცია right join;

- კონცეფცია full join.

მოთხოვნის შესრულების გეგმა.

მოთხოვნების ოპტიმიზაცია.

ტრანზაქციის კონცეფცია. ტრანზაქციის გამოყენება.

ვიუები (Views)

- ვიუს (View) შექმნა.
- ვიუს (View) მოდიფიცირება.
- ვიუს (View) წაშლა.
- ვიუს (View) საშუალებით მონაცემთა ცვლილება.

შენახული პროცედურები (Stored Procedures).

ტრიგერები.

მოდული 17

2 ლექცია

Python-ში მონაცემთა ბაზების გამოყენება

ლექცია 33

მონაცემთა ბაზების გამოყენება.

ORM სისტემები.

- გამოყენების თეორია და პრაქტიკა.
- SQLAlchemy.
- PonyORM და სხვა.

ლექცია 34

NoSQL მონაცემთა ბაზები.

- NoSQL საფუძველბი
- მონაცემთა ბაზების მთავარი ტიპები.

- Redis.
- MongoDB.

მონაცემთა ბაზებთან მუშაობა, მონაცემთა სერიალიზაცია.

- XML DOM ფორმატში ჩაწერა და წაკითხვა.
- StAX და SAX-პარსერი.

მოდული 18

12 ლექცია

ფრეიმვორკი Flask და Django

ლექცია 35

web-ფრეიმვორკების კლასიფიკაცია.

- Web-ფრეიმვორკი Flask.
- შაბლონების მექანიზმი და შაბლონების ენა Jinja2.
- სესიები და ფორმები.
- WEB-ფრეიმვორკი Bottle.

ლექცია 36

ფრონტენდთან ინტეგრირება

- სტილები
- თემფლეიტები
- JS

ლექცია 37

ასინქრონული web-აპლიკაცია.

- ფრეიმვორკი Tornado.
- ბიბლიოთეკა Twisted.

ლექცია 38

Django – ფრეიმვორკი web-აპლიკაციის შესაქმნელად.

- Django-ს დაყენება.
- Django-ს პროექტის შექმნა.
- Django-პროექტის სტრუქტურა (url-view-model-template).
- პატერნი MVC/MVT.

ლექცია 39

მოდელი და ORM.

- მოდელები და ველები.
- კავშირები ცხრილებს შორის, მონაცემთა ბაზების პროექტირება და რეალიზაცია ORM მექანიზმის დახმარებით.
- მიგრაციები.
- მოდელების მენეჯერები.
- ადმინისტრაციული ნაწილი.

ლექცია 40

ადმინისტრატორის მოდულის შემუშავება.

- ადმინისტრატორის ინტერფეისის გამართვა. სორტირება, ფილტრი, ველების რელაქტირება.
- მონაცემთა გამოყვანა.

ლექცია 41

შაბლონების ენა და web-ფორმის შექმნა.

- შაბლონები.
- როუტინგი და Views
- სინტაქსი, ლოგიკური კონსტრუქციები.
- კონტექსტური პროცესორი.

ლექცია 42

ფორმების და გვერდების შექმნა.

- გვერდის დამატება.

- GET/POST მოთხოვნები.
- სტატიკური შიგთავსი და დინამიური.
- ავტორიზაცია.
- წვდომის უფლების შეზღუდვა.
- ვალიდაცია.
- AJAX.

ლექცია 43

სტანდარტული დავალებები.

- ჩანერგილი class-based views.
- Django CBV-სთან ფორმების გამოყენება.

ლექცია 44

სტანდარტული ამოცანები.

- ავთენტიფიკაცია.
- პაგინაცია.

ლექცია 45

Django ADVANCE.

- Middleware.
- სიგნალები.
- შეტყობინებები.
- სესიები.

ლექცია 46

Django ADVANCE-2.

- Django-ში ძველი სტრუქტურები.
- მიუსინები.
- Django დებაგერი.
- ლოგირება, ფოსტის გაგზავნა.

მოდული 19

2 ლექცია

საკურსო პროექტი ჩეტ-ბოტების შექმნა Python-ის საშუალებით.

ლექცია 47

საკურსო პროექტის მიმოხილვა

რა არიოს ჩეტ-ბოტი?

ჩეტ-ბოტების მიზნები და ამოცანები.

ჩეტ-ბოტების არქიტექტურა.

ჩეტ-ბოტის პრაქტიკული მაგალითი.

ლექცია 48

საკურსო პროექტის პრეზენტაცია