

Программа обучения WEB-программированию

Учебный план программы «WEB-программирование»

1. Introduction, acquaintance with terms and ideas of webdevelopment, tools
2. HTML files and browsers, tags, structure of HTML document, dev tools, text formatting tags
3. Numeric systems: binary, decimal. Bits, bytes, RAM, CPU.
4. Inline elements and blocks, text formatting tags continued, case sensitivity, HTML entities
5. Numeric systems, continued: binary, decimal, hexadecimal
6. File tree and paths, images, comments, lists. Dummy images and texts. Links. HTML formatting
7. How CPU thinks. Simplest algorithms, variables. Numeric systems, finished.
8. Metadata, charset, languages. Quotations. What CSS is. Selectors. Color and font properties. Text emphasising. Internal and external CSS.
9. Algorithms: conditions, loops, special cases
10. Sections vs divs. More of links. Google fonts. Complex selectors.
11. Algorithms continued: loops, out of loop conditions, special cases
12. Web safe fonts. Tables. Id vs class. Box model and border-margin-padding.
13. JS engine. History. Features. Standards. DOM. JS console.

14. Algorithms continued: search
15. Box model and border-margin-padding continued. Table continued
16. DOM continued. Datatypes. Variables. Internal and external script. Equality comparison.
Conditions
17. Algorithms continued: order, complexity
18. Background properties. Transparency. Floating table. Wrappers.
19. Precedence. Concatenation. Datatype continued. Comments. Modal windows.
Conditions continued
20. Algorithms continued: games with simple strategy. Version controle.
21. Units: px-%-em-rem. More of text formatting.
22. Ternary operator. Switch.
23. Layout: wrappers continued, float, fixes for float
24. Switch continued. Functions. "For" loop. Arrays.
25. Viewport and units. Reminder of box-model and inline vs inline-block vs block.
26. Functions continued. Function expressions. Short arithmetic notation. Arrays
continued.
27. Positioning. Z-index. Abbreviated color hex code.
28. "For" loop continued. Arrays continued. "While" and "do ... while" loops.
29. Reminder: rgba vs hexadecimal color notation. Gradient. Shadow.
30. Complex datatypes. Functions continued. Objects.
31. Flexbox - first look. Combinator selectors.
32. Objects continued: initialization, methods, this. Reserved words. Interaction with DOM

33. Drawing simple shapes with CSS. Transform.
34. Overview of 5 simple projects requiring DOM interaction. Arrays and Objects continued.
35. Flexbox continued. CSS tables.
36. Memory lifecycle, garbage collector. References
37. Centering methods.
38. References continued. Events. Inline and external event treatment.
39. Button links. Centering methods continued. [attr=value] selectors.
40. "onload" event and <script> location. Callback functions. More array methods.
41. String manipulation. Regular expressions - first look. DOM - element creation.
42. Appearance change by CSS+JS. ClassList interface.
43. JS and style interaction continued. Regular expressions continued.
44. Name tag styling project. Quiz project. Overview of 5 intermediate game projects.
45. Recursion. Hanoi towers algorithm.
46. Recursion and stack overflow. Window object. Execution context. Execution stack. Date and time methods.
47. Recursion continued. Execution context and execution stack continued. Hoisting. Scope. "this". Regular expressions continued. More array methods.
48. Execution stack vs scope chain. "this" continued. OOP. Constructors and instances. Inheritance.
49. Clock and word games project.
50. Inheritance: prototype and prototype chain. Function constructor. "new" operator. Object.create. Passing functions as arguments.

51. Display. Navigation. Input fields, textarea, labels and form.
52. Regular expressions continued. Two-dimensional arrays. Call(), apply(), bind(), reduce().
Variable environment
53. Dropbox. Radiobuttons. Checkbox. Submit.
54. Going deeper for inheritance, scoping, execution stack. Export/import. Aggregate submodules
55. CSS @ rules. Thumb rules for getting design decisions. Not only Google fonts. Hebrew fonts.
56. "var" vs "let". Const. "rest" and "spread" operator. Can I use? Babel. Destructuring.
Template strings. Default arguments. Arrow functions
57. Thumb rules for getting design decisions continued. Cross-browser. Installing icon fonts
58. More arrow functions. Closures. Currying. Symbol datatype. Inheritance. Classes
59. Thumb rules for getting design decisions continued. What is responsive. Planning responsive site project.
60. Inheritance continued. Method chaining. Compose. ES-7, ES-8, ES-10 innovations. Try-catch
61. Basic animation - transition. Media queries. Fluid grid. Starting responsive site project.
62. Finding "this". Latest array methods. Promises. IIFE
63. Finishing responsive site project. Possible - with a touch of JQuery.
64. Promises continued. Possible addition to the project - frontend part of authentication (dummy backend)
65. Flexbox, grid

66. Backend. What is NodeJS. History. How NodeJS works with DB. Asynch architecture.
Install. Read and write files. Modules
67. Flexbox and grid - continued
68. Modules continued. Built in modules. IIFE. Using built-in modules, like 'fs', 'path', 'os'.
Sync and async methods. Events
69. EventEmitter. Listener. Preparing to create a server: client-server, request-response,
protocol, TCP, socket, port
70. Creating a server and working with it. Routing. 404. Starting a project.
71. JSON. HTML templates. Url.parse. Continuing the project.
72. npm. Dependencies. Package.json. Nodemon. Finishing the project. Overview of the
homework project.
73. Buffers. Streams. Pipes. Routing with pipes. Redirection. NodeJS event loop. Express
and ejs
74. Express and ejs continued. Middleware. Body-parser. POST and GET requests. Other
templating engines. Set global variables
75. What is DB. SQL. Relational vs NoSQL. Installing MySQL, creating and populating DB.
76. SQL continued. Querying DB
77. Connecting to MySQL. What is MVC. Starting NodeJS+MySQL project.
78. Finishing NodeJS + MySQL project.
79. What is Angular, how it works. MVC model implementation. Planning a project
80. What is Bootstrap, how it works. Angular: modules, controllers, views. Project
continued



81. Firebase authentication. Angular templates. Routing. Populating database. Project continued.

82. Finishing project.

Организатор имеет право изменять, корректировать программу в случае необходимости.

