**Web Development Training Content**

**Lesson 1.** **High level introduction to computer science**

*Lecture*(2.5 hours, practice included)

* How computer works
* What is Internet
* File system
* HTML document structure
* HTML tags
  + text tags: <h1> - <h6>, <span>, <p>, <pre>
  + text styling tags: <i>, <b>, <em>, <strong>, <br>

*Practice*

Practice on the tags learned during the lecture. Created text content using different text formatting HTML tags.

**Lesson 2. Attributes and routes**

*Lecture*(1.5 hours)

* attributes
* Image (with **height**, **width**, **src**, **alt** attributes )
* <a> tag (with **href** attribute)
* list (with **reversed**, **type**, **start** attributes)
  + ordered list(<ol>)
  + unordered list(<ul>)
  + description list(<dl>)
* defining the routes to files

*Practice* (1 hours)

* Creating 3 types of lists
* Adding images with attributes
* Adding a link (a route to another file in our repository) to move to another page

**Lesson 3. The <form> element and its components**

*Lecture*(1.5 hour)

* <input> , <textarea>, <fieldset>, <legend>, <select>
* elements of <select> tag - <optgroup>, <option>
* attributes - type (text, password, email, radio, checkbox, submit, reset) , placeholder, id
* <button>

*Practice* (1 hour)

1. Create registration form with name, email, password fields (including required fields)
2. Create checkbox and radio field

**Lesson 4. <div> element. Styling elements in HTML**

*Lecture* (1.5 hours)

* <div> element
* Ways to style elements
  + using style attribute in tag
  + using <style> tag
  + creating style.css file and attaching it to html file
* Style properties
  + font-size
  + font-weight
  + color
  + background-color,
  + text-align
  + font–style
  + font-family
  + text-decoration
  + font-spacing
  + letter-spacing
  + line-height
* Google fonts. How to link google fonts.
* Styling using **class** and **id** attributes

*Practice* (1 hours)

Create <div> elements. Add id and class attributes to them and style using all the properties discussed during lecture with style attribute and <style> tag.

**Lesson 5.**

*Lecture* (1.5 hour)

* Inline and block elements
* **Display** property (block, inline, inline-block)
* Ways to set colors of elements (color keyword, rgb, rgba)
* Prioritizing styles in html. Overriding styles. **!important**
* **Border** property (border-right, border-left, border-top, border-bottom), shorthand
* **Margin** and **padding** (shorthand, specified for each side)
* **Border-radius** property

*Practice* (1 hour)

Create 6 divs on three rows: 2 - on first row, 1- on second row, 3 - on third row. Style them using properties discussed during lecture.

**Lesson 6. Styling background and hover selector**

*Homework discussion* (0.5 hours)

*Practice*(1 hours)

Make triangle roof for building

*Lecture* ( 1 hours)

* Background properties
  + Background-image
  + Background-size
  + Background-repeat
  + Background-position
* Hover (with transition)
* Cursor: pointer
* Transition, opacity
* Positioning (absolute, relative, fixed)

Create block with background image and text on it. Add hover effect.

**Lesson 7. Flex container. Parent–child relationship**

*Lecture* (1.5 hour)

* Flex container and its properties
  + flex-direction ✓
  + flex-wrap✓
  + justify-content✓
  + align-items✓
  + align-content
  + align-self
* Parent-child relationship
  + :nth-child
  + :first-child
  + :last-child
* Changing order of elements with **order** property
* Max-width, min-width, max-height, min-height

*Practice* (1 hour)

1. Create a div and 4 divs inside of it. Give them colors and arrange on centered line.
2. Place child divs in corners of parent div.

**Lesson 8. HTML layout**

*Lecture* (1.5 hour)

* HTML layout elements (nav, header, section, footer, aside, article)
* **Before** and **after** selectors
* Transition (**cubic bezier**)
* **Transform** property (scale, translate, rotate)

*Practice* (1hour)

Create the header of a website

**Lesson 9. Grid layout**

*Homework discussion* (0.5 hour)

Homework - to create a page of a website using what was learned during previous lesson

*Lecture* (1 hours)

* **Display** property with **grid** value
* Grid - template - column/row
* Grid - column/row - start/end
* Grid - gap
* Grid - area

*Practice* (1 hours)

Arrange items using grid.

\_\_\_\_\_\_\_\_\_\_`

**Lesson 10. Responsiveness**

*Lesson discussion* (0.5 hours)

*Lecture* (1 hour)

* What is responsive web design
* Media queries
* Orientation (portrait and landscape)

*Practice* (1 hours)

Create responsive table.

**Lesson 11. Keyframes animation**

*Homework discussion* (0.5 hour)

*Lecture* (1 hours)

* Usage of @keyframes
* Property values (name, duration, infinite, alternate)
* Adding many keyframe selectors and css styles in animation

*Practice* (1 hours)

Create animation moving an element to corners of the screen.

**Lesson 12. Introduction to JS**

*Homework discussion*( 0.5 hours)

*Practice* (1 hours)

Practice to repeat all topics discussed during HTML/CSS course.

*Lecture* (1 hour)

* What is Javascript
* The role of javascript in web development

**Lesson 13.Variables and data types**

*Lecture* (1 hour)

* Variables and its types
* Declaration with **var**, **let** and **const** (with hoisting explained)
* Primitive types
* Mathematical and logical operators
* Comparisons
* **For ..** loop

*Practice* ( 1.5 hour)

Discussing some cases related to lecture topics.

**Lesson 14. Loops. If conditional statement**

*Lecture* (1 hours)

* Loops
  + For …
  + While …
  + Do … while ..
* If … else …

*Practice* (1.5 hours)

Discussing cases and solving tasks related to loops and if conditions.

**Lesson 15. Functions**

*Homework discussion* (0.5 hours)

*Lecture* (1 hour)

* What factorial is
* Nested loops
* Functions
  + Function declaration
  + Function expression
  + Arrow function

*Practice* (1 hours)

Calculate factorial of six (using for or while loop)

**Lesson 16. Functions**

*Homework discussion* (0.5 hours)

*Lecture* (1 hours)

* Function expressions
* **return** keyword in function

*Practice* (1 hour)

Write a function to get absolute value of number.

Write a function to calculate a factorial of number.

**Lesson 17. Functions, loops. JS scope**

*Homework* (0.5 hours) +*puzzles discussion* (0.5 hours)

*Lecture*

* Global scope
* Local scope
* Function scope

*Practice* (1.5 hour with lecture)

* Get all a, b numbers for which a\*b=2\*(a+b)
* Exchange the values in two variables (using third variable and using math operations).
* Write a function to return first n elements of Fibonacci sequence.
* Discussing cases related to functions and loops, local and global variables.

**Lesson 18. Global and local variables. Visibility of variables**

*Puzzle discussion* (0.5 hours)

*Homework discussion* (0.5 hours)

*Practice* (1.5 hour)

Discussing cases related to scopes, functions.

**Lesson 19. Closures. Arrow functions. Callbacks**

*Puzzle discussion* (1 hours)

1. Catching the mouse moving through 10 boxes
2. Removing numbers on odd positions

*Practice + Lecture* (1.5 hours)

Discussed cases related to closures, arrow functions, callbacks and arguments in functions.

**Lesson 20.**

*Puzzle discussion* (1 hour)

1. Mushroom puzzle

*Practice* (1.5 hour)

Discussing cases related to closures, arrow functions, callbacks and arguments in functions.

**Lesson 21.**

*Puzzle discussion* (1 hour)

1. 50 trucks puzzle.

*Practice* (1.5 hour)

Discussing cases related to variables, functions.

**Lesson 22. Array. Array methods**

*Lecture*(1.5 hours)

* Creating array and accessing its elements
* **Length** property
* Array methods
  + toString()
  + join()
  + concat()
  + splice()
  + slice()
  + reverse()
  + sort()

*Practice* (1 hours)

Using array methods get max and min elements of array.

**Lesson 23. Array methods**

*Lecture* (1.5 hours)

* Adding elements with **push()** and **unshift()**
* Removing elements from array with **pop()** and **shift()**
* Math object methods
  + pow()
  + sqrt()
  + cbrt()
* Array methods
  + map()
  + every()
  + some()
  + forEach()
  + includes()
  + reduce()

*Practice* (1 hours)

* Fill array b with squares of elements of array a.
* Add not negative numbers of array to new array and sort it.

**Lesson 24. String. String methods**

*Lecture* (1 hour)

* String length
* String methods
  + slice()
  + replace(), replaceAll()
  + toUpperCase(), toLowerCase()
  + concat()
  + trim(), trimStart(), trimEnd()
  + charCodeAt()
  + split()
  + indexOf(), lastIndexOf()

*Practice* (1.5 hour)

* Count how many “l” a string has.
* Write to function to check if a string is palindrome in two ways:

1. Using string methods

2. Reversing the string using loop).

**Lesson 25. Objects**

*Lecture* (1.5 hours)

* Object definition
* Accessing and adding object properties
* Object equality
* Operator **in**
* Iteration over object
* **this** keyword
* Object methods

*Practice* (1 hours)

* Write a function to find the first not repeating character in string.
* Create 2 objects with properties about yourself. First one should have number keys, second one should have string keys.

**Lesson 26. Objects**

*Lecture* + *Discussion cases* related to objects. (1.5 hours)

* Function reuse
  + bind()
  + call()
  + apply()
* Function length

*Practice* (1 hours)

* Create two book objects with name, page, price properties. Add method getInfo() to get information (properties) about those two objects.

**Lesson 27.**

*Lecture* (1.5 hour)

* Object.entries(), Object.values(), Object.keys();
* **Match** object
  + floor()
  + random()
  + ceil()

*Practice* (1 hour)

* Collatz conjecture: if the number is even - divide it by 2, if the number is odd - multiply it by 3 and add until we get 1. This process will eventually reach the number 1. Find the number from 1 to 100 range which takes the most steps to reach number 1.
* Create an empty array. Add 10 random numbers from 100 to 200 range. And find the maximum of the array.

**Lesson 28.**

*Lecture* (1.5 hour)

* Math object
  + max()
  + min()
* Spread operator
* Destructuring

*Practice* (1 hour)

* Create a function that accepts radius as argument and returns the square and length of the circle.
* Create a function that accepts a string and an array of strings as an argument . Function should replace the words of string that are in array with \* and return the new string. (**hard**).

**Lesson 29. Date object. JSON format***Lecture* (1.5 hours)

* Date object
* Date methods
  + getFullYear & setFullyear
  + getMonth & setMonth
  + getMinutes & setMinutes
  + getDate & setDate
  + getDay & setDay
  + getHours & setHours
  + getMilliseconds & setMilliseconds
  + getTime & setTIme
  + parse
* Json
  + parse()
  + stringify()

*Practice* (1 hours)

* Write a function getting year as an argument which returns true if it is a leap year, otherwise returns false.

**Lesson 30. Constructor. Prototype**

*Lecture* (1 hour)

* Object, Number, String, Boolean, Array constructors
* **fromCharCod()**
* Object constructor
* Prototypes

*Practice* (1.5 hour)

* Create number arrays. It should contain -1. Create a new array which should contain sorted elements of the number array ,meanwhile the -1 should be in the same position.
* A string consists of charcodes. Create a function that will convert charcode into char and return a char string.

**Lesson 31. Constructor. Prototype**

*Homework discussion*

*Discussing cases* related to constructors, prototypes (1 hour)

*Practice* (1.5 hour)

* Create Book function–constructor with title, pub\_year and price properties. Create two objects using the constructor. Create a show method in the prototype of the Book which will return properties of Book objects.
* Create a function which accepts two arrays as arguments and returns a new array of their common elements.

**Lesson 32. Basis principles of OOP**

*Lecture* ( 1.5 hours)

* Classes
* Inheritance, encapsulation, polymorphism
* Heap,call stack and eventloop memory
* **\_proto** and **prototype**

*Discussing cases* related to objects.

*Practice* (1.hour)

* Two arrays (*a* and *b*) and number *v* are given. Find out if there is *a* pair of numbers - one is from a and another one is from *b*  - the sum of which is equal *v*. Write a function which will return true if there is such a pair, false - if there is not.

**Lesson 33. Basis principles of OOP**

*Discussing cases* related to objects ( including Inheritance, encapsulation, polymorphism). (2.5 hour)

**Lesson 34. Classes**

*Homework discussion*

*Lecture*

* Constructor in a class
* **Super** keyword
* **Static** initialization blocks
* **Instanceof**

*Practice*

* Write a function that accepts an array and a number as an argument. It should
  + return index of number, if the number is in array
  + return -1, if the number is not in array
* Create Book class with title, pub\_year and price properties and getinfo method which will return information about Book objects. Create two objects using the constructor.

**Lesson 35.**

*Discussing cases* related to past topics.   
*Lecture*

* **SetTimeOut**

*Practice*

* There is an array of fruit. Make a new array with those elements where they are not repeating, sort according to quantity.

**Lesson 36.**

*Discussing cases* on past topics.

**Lesson 37. DOM manipulation**

*Lecture*

* nodeList
* Getting element references
  + getElementByID
  + getElementByClass
  + getElementByTagName
  + selectQuery (selectQueryAll)
* DOM properties
  + innerHTML
  + value
  + textContent
* Getting classes of element and manipulating
  + classList
  + add
* Manipulating attributes
* appendChild
  + getAttribute
  + setAttribute

**Lesson 38. AddEventListener**

*Lecture*

* addEventListener method
* **event** object
* events
  + click
  + mouseenter
  + mouseleave
  + stoppropagation
* event properties
  + target
  + style
  + background

**Lesson 39. Building calculator**

**Lesson 40. Introduction to GIT, Github**

*Lecture* (2.5 hour)

* Version Control Systems
* What is GIT
* Initializing git in a folder
* Main git commands
  + add
  + commit
  + push
  + stash
  + pull
  + log
  + diff
  + show
* What is Github
* Pushing git repository to github
* Cloning repository from github
* Adding collaborators

**Lesson 41.**

*Lecture* (2.5 hour)

* Promise
* async, await
* Iterators and Generators

**Lesson 42. Introduction to GIT, Github**

*Lecture* (2.5 hour)

* Map
* Set
* Event loop
* Macro and micro tasks