Python training program

- **Introduction to Web technologies. HTML structure. Formatting text with HTML**

•Introduction to the subject.
•Introduction to markup languages. HTML hypertext markup language.
•Major mistakes in writing tags.
•The structure of an HTML5 document.
•Page encoding and <meta> tags.
•Tag classification: linear and block.
•Text formatting model: headings and paragraphs. Elements <p>, <h1>, <h6>. Text alignment in block elements: lign attribute.
•Tag classification: logical and physical formatting.
•Practice: creating a simple web page.

- **Formatting with CSS. Lists. Css padding and margins**

•GSS - Cascading Style Sheets.
•Creating lists.
•G03 nested lists.
•Formatting lists with CSS.
•Definition descriptions: <dl>, <dd>, <dt> elements.
•Manage padding and margins.
•Practice: Creating Lists.

- **Graphics in web design. Graphics optimization. Hyperlinks. Principles website navigation**

•Graphic file formats in web.
•Ter <img /> and its attributes (src, alt, width, height, border).
•The background of the page is the background property.
•General information about hyperlinks.
•Absolute and relative addressing.
•Menu creation using list structure (<ul>, <li>), its formatting. display property. Convert link to block element.
•Pseudo-classes.
•OSS cursor property.
•Practice: work on developing an image gallery.
•CSS3 properties.
•Working with multimedia.

- **Tables**

•Creating a simple table. The <table>, <tr> and <td> tags.
•Merging cells: colspan, rowspan attributes.
•Tags for logical structuring of tables: <thead>, <tbody>, <tfoot>. Tags for logical grouping of columns: <colgroup>, <col>.
•Table frame control: attributes frame, rules.
•Practice: creating complex tables.
•Basics of tabular layout. An example of tabular layout: its disadvantages.

- **Positioning**

Web page layout Blocks

•The position property.
•Visibility, overflow properties.
•Practice.
•Block layout basics. Rules Layouts.
•Consideration of the simplest structures pages.
•Rubber structure. Blocks with negative margin

- **Forms. Frames**

•Introduction to forms.
•Form controls.
•Creating forms with HTML.
•Form elements from HTML5.
•Form validation with HTML5.
•Formatting form elements with CSS.
•Frames and their structure (theoretical intelligence).

- **Introduction to web programming in Python**

•Introduction to web programming and principles of web applications
•Python language, scopes.
•Introduction to Python.
•The Python interpreter and its environment.
•Data types, variables and syntactic constructions.

- **Branch operators, cycles, exceptions**

•Conditional statements and their syntax.
•The concept of exceptions.
•Cycles.

- **Strings, lists**

•Strings.
•Lists.

- **Modules, functions**

•Functions and modules.
•Advanced techniques for working with functions.
•Functional programming.
•Closure, currying, decorators.

- **Sorting, search**

•Sorting.
•Search.

- **Tuples, sets, dictionaries**

•Tuples, sets, dictionaries.
•Practical examples of use.

- **Files**

•Files context manager.
•File types text and binary.
•Practical examples of use.

- **Data packaging**

•Serialization and dese implementation.
•Pickle module.
•Json module.
•Third party serialization modules.

- **Control systems versions**

•Version control.
•The need for version control.
•Overview of version control systems.

- **OOP**

•Introduction to OOP.
•User-defined data types.
•Inheritance and encapsulation.
•Polymorphism
•Creating and managing behavior class instances.
•Metaclasses.

- **Data Structures**

•Related lists.
•Grex.
•Queue.
•Trees.

- **Patterns design**

•Patterns design.
•Causes of Patterns Design.
•Principles of application, choice of pattern design.
•Principles for dividing patterns into categories.
•Introduction to UML
•Using UML in the analysis of design patterns.
•Generating patterns.
•GT hand patterns.
•Behavior patterns.

- **MVC Pattern**

•MCC Pattern.
•Goals and objectives of the Model-View-Controller pattern.
•Model, view, controller.
•Examples of using the MVC pattern.

- **Design principles SOUD classes**

•Overview of the problems encountered in the design and development of classes.
•Class Design Principles SOUD.
•Examples of using SOUD principles.

- **Modular testing**

•Modular testing.
•Goals and objectives of modular testing.
•The Need for Modular Testing.
•Overview of Tools for Modular Testing.
•Unit Testing Tool python applications.

- **Parallel, multithreading and network programming**

•Parallel and multi-threaded programming.
•Network programming.

- **Introduction to working with databases**

•Introduction to Database Theory
•SELECT, INSERT, UPDATE queries, DELETE.
•Multi-table databases.
•Normalization.
•Multi-table queries.
•Aggregation functions.
•The concept of grouping. Keyword group by.
•Having keyword. Comparative having and where analysis.
•Subqueries.
•Operators for using subqueries, joins.
•Query execution plan.
•Query optimization.
•The concept of a transaction. Usage transactions.
•Representation.
•Triggers.

- **Use of databases in Python, Numpy and Pandas libraries**

•Using databases.
•ORM systems.
•NoSQL databases.
•Working with databases, serialization data.

- **Teamwork, management programming projects**

•Software project management.
•The reasons for the emergence of the discipline «Management of software projects».
•Gantt charts.
•Software Management Questions projects.
•Analysis of domain terms.
•Project characteristics.
•The costs associated with the project.
•General overview of models and methodologies development process.
•Learn more about Scum

- **Frameworks**

•Classification of web frameworks.
•Asynchronous web applications.
•Django framework for creat web applications.
•Models and ORMs.
•Working with the administrative part, setting up the display.
•Template language and web form creation.
•Building of forms and pages.
•Standard tasks.
•Dive into Django.

- **Introduction to JavaScript**

•Scripts executed on the side client.
•JavaScript.
•The history of the creation of JavaScript.
•Differences between JavaScript and Java, JScipt, ECMAScript.
•Versions of JavaScript.
•Concept of Document Object Model.
•Concept of the Browser Object Model.
•Embedding in HTML documents. Editor of JavaScript code.
•Tag <noscript>.
•Basics of syntax.
•Variables. Naming rules variables.

•Data types.
•Operators.
•Data input/output. Dialog boxes.
•Conditions. Cycles. Function.
•Arguments object.
•The area in the variable scope. The keyword this.
•Recursion.

- **Object. Arrays. Array object. Strings. String object. Object date. Math object. Introduction to object-oriented programming**

•Objects.
•Arrays.
•Strings.
•Delays and intervals. Periodic function call.
•Three fundamental principles of the OOP.
•Concept of class and object terms JavaScript.
•Properties.
•Methods.
•Accessory properties.
•Constructor.
•Concept of «prototypes».
•Inheritance.

- **Event handling**

•Handling events in scripts.
•Managing Styles and Elements web pages.
•The event object and its properties.
•Default event handlers (standard handlers), preventing the standard handler from being called.
•Image object. Picture management and rollovers.

- **Browser Object Model. Document Object Model**

•Browser Object Model objects.
•Document Object Model.
•DOM via BOM difference.
•DOM objects. Node hierarchy.
•GBO properties and methods of the DOM. DOM event model.
•Changing the DOM tree.
•Introduction to Document Link Objects.
•Selection and text control range: Selection and TextRange objects.
•HTML5 DOM Features.

- **Forms**

•Application of forms. Placement of elements form in HTML.
•Forms collection. Creation and programming of form elements.

- **Validation of forms. Usage Cookie**

•A RegExp object. Rules for recording regular expressions.
•String and RegExp object methods for working with regular expressions.
•Data Validation.
•Cookie forms.
•Advantages and disadvantages of cookies
•Creation, use and deletion cookie.

- **Drawing with canvas, media support**

•Basic features of canvas.
•Support for media features.

- **JSON, Ajax**

•JSON.
•Goals and objectives of JSON.
•JSON Syntax.
•JSON object.
•Setting up a custom user serialization in JSON. toJSON method.
•Synchronous and asynchronous requests.
•AJAX.
•XMLHttpRequest object.
•XMLHttpRequest Methods and Properties.

- **Introduction to jQuery**

•jQuery features.
•jQuery versions.
•jQuery connection.
•Accessing page elements with the $.
•The concept and types of the selector.
•Traversing. DOM traversal methods. Method filter, next, nextAll, prev, prevAll, siblings, etc.

- **Events and jQuery**

•Creating event handlers with using jQuery.
•Removing event handlers.
•Event object and jQuery.
•Impact on event handling.
•Start event processing.

- **Styles and Animation**

•CSS Method.
•Show and hide elements.
•Show and hide methods.
•Creating effects.
•Animation.

- **Interacting with the DOM**

•Creating new DOM elements.
•Inserting, moving and copying DOM elements.
•Interaction with attributes.

- **AJAX and jQuery**

•JGON.
•AJAX mechanisms inside the library JQuery.
•Using the GET method.
•Using the POST method.
•Events and AJAX within jQuery.
•Error processing.

- **Using the jQuery plugins**

•The concept of a jQuery plugin.
•Plugin connection.
•Plugin examples.

FINAL FINAL PROJECT