

Semi-structured Data

Tutorials' Description

These tutorials introduce the concepts and practical applications of semi-structured data in modern web systems. It covers JavaScript programming, MVC architecture, asynchronous communication, XML processing in Python, and data exchange between systems and databases. Students will develop a complete functional web application integrating front-end and back-end technologies.

Learning Objectives

1. Apply advanced JavaScript programming concepts.
2. Design applications using the MVC architectural pattern.
3. Implement asynchronous client-server communication.
4. Develop dynamic web interfaces connected to server data.
5. Parse and generate XML files using Python.
6. Integrate backend-generated data into a web user interface.
7. Import and export semi-structured data between database systems.
8. Design and implement a complete data-driven web application.

Tools and Technologies

- JavaScript (ES6+)
- HTML5 / CSS3
- MVC Architectural Pattern
- XMLHttpRequest / AJAX
- Python 3
- XML (Extensible Markup Language)
- JSON (JavaScript Object Notation)
- Relational Databases (MySQL)

Tutorial Schedule (12 Sessions)

1. JavaScript Refresher

Review of JavaScript fundamentals: variables, functions, arrays, DOM manipulation, and events.

2. Object-Oriented Programming in JavaScript

Objects, classes, constructors, prototypes, encapsulation, and modular design.

3. MVC Pattern: First Glance

Introduction to Model-View-Controller architecture and separation of concerns.

4. MVC Pattern: Handling More Complex Data

Managing structured and nested data with MVC and dynamic views.

5. Introduction to XMLHttpRequest

Asynchronous communication, HTTP requests, and handling server responses.

6. Building a Fully Functional Web Application

Developing a complete dynamic web application with server-side data integration.

7. First Evaluation

Assessment covering JavaScript, MVC, and asynchronous programming.

8. Parsing an XML File in Python

Reading and extracting structured data from XML using Python.

9. Creating an XML File in Python

Generating structured XML documents programmatically.

10. Linking Backend Data to the User Interface

Integrating Python-generated data into a web interface.

11. Importing and Exporting Data Between Databases

Data transformation, migration, and semi-structured data exchange.

12. Second Evaluation

Second assessment on desktop apps that use XML.