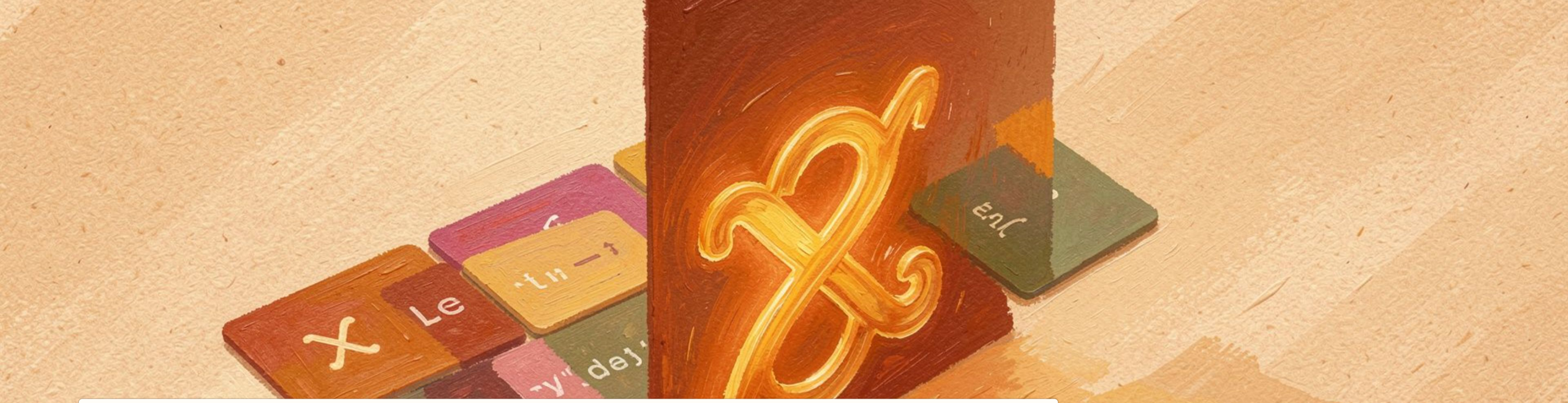
АО «АЛМАТИНСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»
https://atu.edu.kz/

Факультет «Информационные технологии»
Кафедра «Компьютерная инженерия»

Название курса: «Разработка клиентских Web-приложений»



Зиятбекова Гулзат Зиятбеккызы – PhD, ассоциированный профессор
g.ziyatbekova@atu.edu.kz



ЛЕКЦИЯ №13 · JAVASCRIPT

Массивы, объекты и современные возможности JavaScript

Ключевые структуры данных и методы работы с ними

01 · СТРУКТУРА ДАННЫХ

Массив хранит несколько значений в одной структуре

- Массив объединяет связанные данные в одной переменной
- Элементы можно перебирать и преобразовывать методами массива



Одна переменная — несколько связанных значений

ПРИМЕР

```
const students = ["Ayan", "Dana", "Ali"];
```

02 · ПОЗИЦИЯ

Индекс начинается с нуля

Первый элемент

students[0] возвращает "Ayan"

Индекс

Определяет позицию элемента в массиве

Нулевая точка

Первый элемент находится по индексу 0

Методы массивов решают типовые задачи



map()

Преобразовать каждый элемент



filter()

Выбрать элементы по условию



reduce()

Получить один итоговый результат

МЕТОД 01

map()

Создаёт новый преобразованный массив

Каждый элемент проходит одно и то же преобразование

Метод проходит по каждому студенту.

КОД

```
const upper = students.map(student =>
student.toUpperCase());
```

Результат: новый массив с именами в верхнем регистре

МЕТОД 02

filter() оставляет элементы по условию

Метод проверяет каждый элемент.

КОД

```
const result = students.filter(student =>
student.length > 3);
```

УСЛОВИЕ

length > 3

В результат попадают только имена длиннее трёх символов

МЕТОД 03 · ИТОГ

reduce() сводит массив к одному результату

Массив

Все элементы исходных данных

Итерация

На каждой итерации накапливает промежуточный результат

Один итог

Сумма, подсчёт, группировка и другие итоги

Главный вопрос: какой итог нужно получить из всех элементов?

Объект описывает сущность через пары «ключ — значение»

```
const student = { name: "Ayan", age: 20,
group: "IT-23" };
```

Ключи: name, age, group

Значения: имя, возраст и учебная группа



Сущность собирается из связанных характеристик

Свойство объекта доступно через точку



Название

student.name



Значение

Возвращает имя студента



Сущность

Объекты хранят связанные данные о сущности

Destructuring раскладывает объект по переменным

```
const { name, age } = student
```

- Свойства объекта извлекаются в отдельные переменные
- Код становится короче и удобнее для чтения



Одно выражение — отдельные переменные