Частное учреждение образования

«Колледж бизнеса и права»

|  |
| --- |
| УТВЕРЖДАЮ  Заведующий  методическим кабинетом  \_\_\_\_\_\_\_\_\_\_Е.В.Паскал  «\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ |

|  |  |
| --- | --- |
| Специальность:  2-40 01 01 «Программное обеспечение информационных технологий» | Дисциплина: «Скриптовые языки программирования» |

**Лабораторная работа № 10**

**Инструкционно-технологическая карта**

Тема: Объекты основы. Литералы и свойства. Квадратные скобки. Свойство из переменной. Ограничения на имена свойств. Цикл for..in.

Цель: Научиться создавать объекты, создавать и изменять литералы и свойства, использовать квадратные скобки, использовать цикл for..in

Время выполнения: 2 часа

### Порядок выполнения работы

1. Изучить теоретические сведения к лабораторной работе.

2. Реализовать алгоритм решения задачи.

3. Отлаженную, работающую программу сдать преподавателю. Работу программы показать с помощью самостоятельно разработанных тестов.

4. Ответить на контрольные вопросы.

### Теоретические сведения

Объекты

В JavaScript существует 8 типов данных. Семь из них называются «примитивными», так как содержат только одно значение (будь то строка, число или что-то другое).

Объекты же используются для хранения коллекций различных значений и более сложных сущностей. В JavaScript объекты используются очень часто, это одна из основ языка. Поэтому важно понять их, прежде чем углубляться куда-либо ещё.

Объект может быть создан с помощью фигурных скобок {…} с необязательным списком свойств. Свойство – это пара «ключ: значение», где ключ – это строка (также называемая «именем свойства»), а значение может быть чем угодно.

Можно представить объект в виде ящика с подписанными папками. Каждый элемент данных хранится в своей папке, на которой написан ключ. По ключу папку легко найти, удалить или добавить в неё что-либо.

Пустой объект («пустой ящик») можно создать, используя один из двух вариантов синтаксиса:

let user = new Object(); // синтаксис "конструктор объекта"

let user = {};  // синтаксис "литерал объекта"

Обычно используют вариант с фигурными скобками {...}. Такое объявление называют литералом объекта или литеральной нотацией.

При использовании литерального синтаксиса {...} сразу можно поместить в объект несколько свойств в виде пар «ключ: значение»:

let user = {     // объект

    name: "John",  // под ключом "name" хранится значение "John"

    age: 30        // под ключом "age" хранится значение 30

  };

У каждого свойства есть ключ (также называемый «имя» или «идентификатор»). После имени свойства следует двоеточие ":", и затем указывается значение свойства. Если в объекте несколько свойств, то они перечисляются через запятую.

В объекте user сейчас находятся два свойства:

1. Первое свойство с именем «name» и значением «John».
2. Второе свойство с именем «age» и значением 30.

Можно сказать, что наш объект user – это ящик с двумя папками, подписанными «name» и «age».

В любой момент можно добавить в него новые папки, удалить папки или прочитать содержимое любой папки.

Для обращения к свойствам используется запись «через точку»:

// получаем свойства объекта:

alert( user.name ); // John

alert( user.age ); // 30

Значение может быть любого типа. Свойство с логическим значением:

user.isAdmin = true;

Для удаления свойства следует использовать оператор delete:

delete user.age;

Имя свойства может состоять из нескольких слов, но тогда оно должно быть заключено в кавычки:

let user = {

    name: "John",

    age: 30,

    "likes birds": true  // имя свойства из нескольких слов должно быть в кавычках

  };

Последнее свойство объекта может заканчиваться запятой:

let user = {

    name: "John",

    age: 30,

  }

Это называется «висячая запятая». Такой подход упрощает добавление, удаление и перемещение свойств, так как все строки объекта становятся одинаковыми.

Объект, объявленный через const, может быть изменён.

Например:

const user = {

    name: "John"

  };

  user.name = "Pete"; // (\*)

  alert(user.name); // Pete

Может показаться, что строка (\*) должна вызвать ошибку, но нет, здесь всё в порядке. Дело в том, что объявление const защищает от изменений только саму переменную user, а не её содержимое.

Определение const выдаст ошибку только если присвоить переменной другое значение: user=....

Квадратные скобки

Для свойств, имена которых состоят из нескольких слов, доступ к значению «через точку» не работает:

// это вызовет синтаксическую ошибку

user.likes birds = true

JavaScript видит, что произошло обращание к свойству user.likes, а затем идёт непонятное слово birds. В итоге синтаксическая ошибка.

Точка требует, чтобы ключ был именован по правилам именования переменных. То есть не имел пробелов, не начинался с цифры и не содержал специальные символы, кроме $ и \_.

Для таких случаев существует альтернативный способ доступа к свойствам через квадратные скобки. Такой способ сработает с любым именем свойства:

let user = {};

// присваивание значения свойству

user["likes birds"] = true;

// получение значения свойства

alert(user["likes birds"]); // true

// удаление свойства

delete user["likes birds"];

Сейчас всё в порядке. Важно, что строка в квадратных скобках заключена в кавычки (подойдёт любой тип кавычек).

Квадратные скобки также позволяют обратиться к свойству, имя которого может быть результатом выражения. Например, имя свойства может храниться в переменной:

let key = "likes birds";

// то же самое, что и user["likes birds"] = true;

user[key] = true;

Здесь переменная key может быть вычислена во время выполнения кода или зависеть от пользовательского ввода. После этого можно использовать её для доступа к свойству. Это даёт большую гибкость.

Пример:

let user = {

    name: "John",

    age: 30

  };

  let key = prompt("Что нужно узнать о пользователе?", "name");

  // доступ к свойству через переменную

  alert( user[key] ); // John (если ввели "name")

Запись «через точку» такого не позволяет:

let user = {

    name: "John",

    age: 30

  };

  let key = "name";

  alert( user.key ); // undefined

Ограничения на имена свойств

Как уже известно, имя переменной не может совпадать с зарезервированными словами, такими как «for», «let», «return» и т.д.

Но для свойств объекта такого ограничения нет:

// эти имена свойств допустимы

let obj = {

    for: 1,

    let: 2,

    return: 3

  };

  alert( obj.for + obj.let + obj.return );  // 6

Иными словами, нет никаких ограничений к именам свойств. Они могут быть в виде строк или символов.

Все другие типы данных будут автоматически преобразованы к строке.

Например, если использовать число 0 в качестве ключа, то оно превратится в строку «0»:

let obj = {

    0: "Тест" // то же самое что и "0": "Тест"

  };

  // обе функции alert выведут одно и то же свойство (число 0 преобразуется в строку "0")

  alert( obj["0"] ); // Тест

  alert( obj[0] ); // Тест (то же свойство)

Есть небольшой подводный камень, связанный со специальным свойством \_\_proto\_\_. Невозможно установить его в необъектное значение:

let obj = {};

obj.\_\_proto\_\_ = 5; // присвоим число

alert(obj.\_\_proto\_\_);

// [object Object], значение – это объект, т.е. не то, что ожидали

Присвоение примитивного значения 5 игнорируется.

Цикл «for…in»

Для перебора всех свойств объекта используется цикл for..in. Этот цикл отличается от изученного ранее цикла for(;;).

Синтаксис:

for (key in object) {

    // тело цикла выполняется для каждого свойства объекта

  }

К примеру, вывод всех свойств объекта user:

let user = {

    name: "John",

    age: 30,

    isAdmin: true

  };

  for (let key in user) {

    // ключи

    alert( key );  // name, age, isAdmin

    // значения ключей

    alert( user[key] ); // John, 30, true

  }

Важно, что все конструкции «for» позволяют объявлять переменную внутри цикла, как, например, let key здесь.

Кроме того, можно использовать другое имя переменной. Например, часто используется вариант «for (let prop in obj) ».

### Контрольные вопросы

1. Что такое литеральная нотация?
2. Что такое висячая запятая?
3. Описать синтаксис объявления свойства.
4. При помощи какого оператора можно обратиться к свойству?

***Общее задание:***

**Задание 1**. Написать код, выполнив задание из каждого пункта отдельной строкой:

1. Создать пустой объект user.
2. Добавить свойство name со значением John.
3. Добавить свойство surname со значением Smith.
4. Изменить значение свойства name на Pete.
5. Удалить свойство name из объекта.

**Задание 2**. Написать функцию isEmpty(obj), которая возвращает true, если у объекта нет свойств, иначе false.

**Задание 3.** Есть объект, в котором хранятся зарплаты команды:

let salaries = {

    John: 100,

    Ann: 160,

    Pete: 130

  }

Написать код для суммирования всех зарплат и сохраните результат в переменной sum.

**Задание 4.** Создать функцию multiplyNumeric(obj), которая умножает все числовые свойства объекта obj на 2. multiplyNumeric не нужно ничего возвращать. Следует напрямую изменять объект.

**Индивидуальное задание.**

**Вариант 1.**

Описать объект с именем STUDENT, содержащую следующие поля:

* + фамилия и инициалы;
  + номер группы;
  + успеваемость (массив из пяти элементов).

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из десяти структур типа

STUDENT, записи должны быть упорядочены по возрастанию номера группы.

2. вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0.

3. если таких студентов нет, вывести соответствующее сообщение.

**Вариант 2.**

Описать объект с именем STUDENT, содержащую следующие поля:

* + фамилия и инициалы;
  + номер группы;
  + успеваемость (массив из пяти элементов).

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT, записи должны быть упорядочены по возрастанию среднего балла.

2. вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5.

3. если таких студентов нет, вывести соответствующее сообщение.

**Вариант 3**

Описать объект с именем STUDENT, содержащую следующие поля:

* фамилия и инициалы;
* номер группы;
* успеваемость (массив из пяти элементов).

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT, записи должны быть упорядочены по алфавиту.

2. вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2.

3. если таких студентов нет, вывести соответствующее сообщение.

**Вариант 4.**

Описать объект с именем AEROFLOT, содержащую следующие поля:

* название пункта назначения рейса;
* номер рейса;
* тип самолета.

Написать скрипт, выполняющую следующие действия:

1. Ввод с клавиатуры данных в массив, состоящий из семи элементов типа AEROFLOT, записи должны быть упорядочены по возрастанию номера рейса;

2. Вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры.

3. Если таких рейсов нет, выдать на дисплей соответствующее сообщение.

**Вариант 5.**

Описать объект с именем AEROFLOT, содержащую следующие поля:

* название пункта назначения рейса;
* номер рейса;
* тип самолета.

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из семи элементов типа AEROFLOT, записи должны быть размещены в алфавитном порядке по названиям пунктов назначения.

2. вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры.

3. если таких рейсов нет, выдать на дисплей соответствующее сообщение.

**Вариант 6.**

Описать объект с именем WORKER, содержащую следующие поля:

* фамилия и инициалы работника;
* название занимаемой должности;
* год поступления на работу.

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из десяти структур типа WORKER, записи должны быть размещены по алфавиту.

2. вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры.

3. если таких работников нет, вывести на дисплей соответствующее сообщение.

**Вариант 7.**

Описать объект с именем TRAIN, содержащую следующие поля:

* название пункта назначения;
* номер поезда;
* время отправления.

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из восьми элементов типа TRAIN, записи упорядочить по алфавиту по названиям пунктов назначения;

2. вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени.

3. если таких поездов нет, выдать на дисплей соответствующее сообщение.

**Вариант 8.**

описать объект с именем TRAIN, содержащую следующие поля:

* название пункта назначения;
* номер поезда;
* время отправления.

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из шести элементов типа TRAIN, записи должны быть упорядочены по времени отправления поезда.

2. вывод на экран информации о поездах, направляющихся в пункт, название которого введено с клавиатуры.

3. если таких поездов нет, выдать на дисплей соответствующее сообщение.

**Вариант 9.**

Описать объект с именем TRAIN, содержащую следующие поля:

* название пункта назначения;
* номер поезда;
* время отправления.

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из восьми элементов типа TRAIN, записи должны быть упорядочены по номерам поездов.

2. вывод на экран информации о поезде, номер которого введен с клавиатуры.

3. если таких поездов нет, выдать на дисплей соответствующее сообщение.

**Вариант 10.**

Описать объект с именем MARSH, содержащую следующие поля:

* название начального пункта маршрута;
* название конечного пункта маршрута;
* номер маршрута.

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из восьми элементов типа MARSH, записи должны быть упорядочены по номерам маршрутов.

2. вывод на экран информации о маршруте, номер которого введен с клавиатуры.

3. если таких маршрутов не т, выдать на дисплей соответствующее сообщение.

**Вариант 11.**

Описать объект с именем MARSH, содержащую следующие поля:

* название начального пункта маршрута;
* название конечного пункта маршрута;
* номер маршрута.

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из восьми элементов типа MARSH, записи должны быть упорядочены по номерам маршрутов.

2. вывод на экран информации о маршрутах, которые начинаются или оканчиваются в пункте, название которого введено с клавиатуры.

3. если таких маршрутов нет, выдать на дисплей соответствующее сообщение.

Описать объект с именем NOTE, содержащую следующие поля:

* фамилия, имя;
* номер т елефона;
* дата рождения (массив из трех чисел).

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из восьми элементов типа NOTE, записи должны быть упорядочены по датам рождения.

2. вывод на экран информации о человеке, номер телефона которого введен с клавиатуры.

3. если такого нет, выдать на дисплей соответствующее сообщение.

**Вариант 13.**

Описать объект с именем NOTE, содержащую следующие поля:

* фамилия, имя;
* номер телефона;
* дата рождения (массив из трех чисел).

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из восьми элементов типа NOTE, записи должны быть размещены по алфавиту.

2. вывод на экран информации о людях, чьи дни рождения приходятся на месяц, значение которого введено с клавиатуры.

3. если таких нет, выдать на дисплей соответствующее сообщение.

**Вариант 14.**

Описать объект с именем NOTE, содержащую следующие поля:

* фамилия, имя;
* номер телефона;
* дата рождения (массив из трех чисел).

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из восьми элементов типа NOTE, записи должны быть упорядочены по трем первым цифрам номера телефона.

2. вывод на экран информации о человеке, чья фамилия введена с клавиатуры.

3. если такого нет, выдать на дисплей соответствующее сообщение.

**Вариант 15.**

Описать объект с именем ZNAK, содержащую следующие поля:

* фамилия, имя;
* знак Зодиака;
* дата рождения (массив из трех чисел).

Написать скрипт, выполняющую следующие действия:

1. ввод с клавиатуры данных в массив, состоящий из восьми элементов типа ZNAK, записи должны быть упорядочены по датам рождения.

2. вывод на экран информации о человеке, чья фамилия введена с клавиатуры.

3. если такого нет, выдать на дисплей соответствующее сообщение.

**Литература**

**Диков, А.В.** Клиентские технологии веб программирования: JavaScript и DOM: учебное пособие / А.В. Диков. – СПб: Лань, 2020 – 124 с.

**Читанамбри, Кирупа.** Изучаем React / Кирупа Читанамбри – 2-е изд. – М.: Эксмо, 2019. 368 с.

**Хавербеке, Марейн.** Выразительный JavaScript. Современное веб-программирование. / Марейн Хавербеке – 3-е изд. – СПб.: Питер, 2019. – 480 с.:

Преподаватель Рогалевич А.В.

Рассмотрено на заседании цикловой

комиссии программного обеспечения

информационных технологий №10

Протокол № \_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Председатель ЦК\_\_\_\_\_\_\_\_ В.Ю.Михалевич