

ДИСЦИПЛИНА	Фронтенд и бэкенд разработка
ИНСТИТУТ	Институт перспективных технологий и индустриального программирования
КАФЕДРА	Кафедра индустриального программирования
ВИД УЧЕБНОГО МАТЕРИАЛА	Практические занятия
ПРЕПОДАВАТЕЛЬ	Загородних Николай Анатольевич
СЕМЕСТР	1 семестр, 2024-2025 гг.

Практическое занятие 10. Основы работы с переменными, условными операторами, циклами, массивами и функциями в JavaScript.

Изучаемые вопросы

1. Объявление и использование переменных.
2. Условные операторы (if, else, switch).
3. Циклы (for, while).
4. Работа с массивами (добавление, удаление элементов, перебор).
5. Основные функции и их использование.

Краткая теория

Переменные

- let: объявляет переменную с блочной областью видимости.
- const: объявляет константу, значение которой нельзя изменить.

Условные операторы

- if: выполняет блок кода, если условие истинно.
- else: выполняет блок кода, если предыдущее условие ложно.
- switch: выбирает один из нескольких блоков кода для выполнения.

Циклы

- for: используется для повторения блока кода заданное количество раз.
- while: повторяет блок кода, пока условие истинно.

Массивы

Массивы — это упорядоченные коллекции данных. Основные методы работы с массивами:

- push(): добавляет элемент в конец массива.
- pop(): удаляет последний элемент массива.
- shift(): удаляет первый элемент массива.

- `unshift()`: добавляет элемент в начало массива.
- `forEach()`: выполняет указанную функцию для каждого элемента массива.

Объявление функции

Вначале идёт ключевое слово `function`, после него имя функции, затем список параметров в круглых скобках через запятую (в вышеприведённом примере он пустой) и, наконец, код функции, также называемый «телом функции», внутри фигурных скобок.

```
js > JS main.js > ...  
1  function имя(параметры) {  
2      ...тело...  
3  }  
4
```

Пример объявления функции

```
js > JS main.js > ...  
1  // Функциональное объявление  
2  function greet() {  
3      console.log("Hello, world!"); // Выводит приветствие  
4  }  
5  
6  // Вызов функции  
7  greet(); // Вывод: Hello, world!  
8
```

Условный оператор if

Условный оператор `if` позволяет выполнять блок кода, если заданное условие истинно.

```
js > JS main.js > ...  
1  let age = 18;  
2  
3  if (age >= 18) {  
4      console.log("Вы совершеннолетний."); // Вывод: Вы совершеннолетний.  
5  }  
6
```

Преобразование к логическому типу

JavaScript автоматически преобразует значения в логический тип для проверки условий. Например, `0`, `null`, `undefined`, `NaN`, пустая строка (`""`) и `false` считаются ложными (`false`), остальные — истинными (`true`).

```

js > JS main.js > ...
1   let value = 0;
2
3   if (value) {
4       console.log("Это истинное значение.");
5   } else {
6       console.log("Это ложное значение."); // Вывод: Это ложное значение.
7   }
8

```

Блок else выполняется, если условие в if ложно.

Несколько условий if else

Можно использовать несколько условий с помощью цепочки if ... else if ... else.

```

js > JS main.js > ...
1   let score = 85;
2
3   if (score >= 90) {
4       console.log("Отлично!");
5   } else if (score >= 75) {
6       console.log("Хорошо!"); // Вывод: Хорошо!
7   } else {
8       console.log("Попробуйте снова.");
9   }
10

```

Синтаксис switch

Оператор switch позволяет выполнять разные блоки кода в зависимости от значения переменной. Это удобно для проверки нескольких значений одной переменной.

```

js > JS main.js > ...
1   let fruit = "apple";
2
3   switch (fruit) {
4       case "banana":
5           console.log("Это банан.");
6           break;
7       case "apple":
8           console.log("Это яблоко."); // Вывод: Это яблоко.
9           break;
10      default:
11          console.log("Неизвестный фрукт.");
12  }
13

```

Пример работы

Вот пример использования оператора switch для определения дня недели по номеру.

```

js > JS main.js > ...
1   let day = 3;
2
3   switch (day) {
4       case 1:
5           console.log("Понедельник");
6           break;
7       case 2:
8           console.log("Вторник");
9           break;
10      case 3:
11          console.log("Среда"); // Вывод: Среда
12          break;
13      case 4:
14          console.log("Четверг");
15          break;
16      case 5:
17          console.log("Пятница");
18          break;
19      case 6:
20          console.log("Суббота");
21          break;
22      case 7:
23          console.log("Воскресенье");
24          break;
25      default:
26          console.log("Некорректный номер дня.");
27  }
28

```

Цикл while

Цикл while выполняет блок кода до тех пор, пока заданное условие истинно. Если условие ложно с самого начала, блок кода не выполнится ни разу.

Синтаксис

```

js > JS main.js
1   while (условие) {
2       // Код, который будет выполняться, пока условие истинно
3   }
4

```

Пример

```

js > JS main.js > ...
1   let count = 0;
2
3   // Цикл выполняется, пока count меньше 5
4   while (count < 5) {
5       console.log(count); // Вывод: 0, 1, 2, 3, 4
6       count++; // Увеличиваем count на 1
7   }
8

```

Цикл for

Цикл for используется для выполнения блока кода определенное количество раз. Он часто используется, когда известен диапазон значений.

Синтаксис

```
js > JS main.js
1   for (инициализация; условие; итерация) {
2       // Код, который будет выполняться в каждой итерации
3   }
4
```

Пример

```
js > JS main.js > ...
1   // Цикл от 0 до 4
2   for (let i = 0; i < 5; i++) {
3       console.log(i); // Вывод: 0, 1, 2, 3, 4
4   }
5
```

Основы работы с массивами

Массив — это упорядоченная коллекция данных, которая может содержать элементы любого типа, включая другие массивы. Массивы в JavaScript являются объектами и позволяют хранить набор значений.

Объявление массива

Массив можно объявить несколькими способами:

```
3   // Используя литерал массива
4   let fruits = ['apple', 'banana', 'orange'];
5
6   // Используя конструктор Array
7   let numbers = new Array(1, 2, 3);
8
```

Нумерация элементов массива

Элементы массива нумеруются с нуля. Первый элемент имеет индекс 0, второй — 1 и так далее.

```
js > JS main.js
1   console.log(fruits[0]); // Вывод: 'apple'
2   console.log(fruits[1]); // Вывод: 'banana'
3
```

Изменение или добавление элементов

Можно изменять существующие элементы или добавлять новые:

```
js > JS main.js
1  fruits[1] = 'mango'; // Изменяем 'banana' на 'mango'
2  fruits.push('grape'); // Добавляем 'grape' в конец массива
3  console.log(fruits); // Вывод: ['apple', 'mango', 'orange', 'grape']
4
```

Определение количества элементов в массиве

Для определения количества элементов используется свойство `length`.

```
js > JS main.js
1  console.log(fruits.length); // Вывод: 4
```

Полезные источники информации

1. MDN Web Docs — JavaScript (Полное руководство по JavaScript)

<https://developer.mozilla.org/ru/docs/Web/JavaScript>

2. Learn JavaScript — Учебник на русском

Учебный ресурс с примерами и заданиями.

<https://learn.javascript.ru>

3. JavaScript.info - подробный курс по JavaScript на русском языке.

<https://learn.javascript.ru>

Тренажеры

<https://codepen.io/pen>

Задания

Задание 1: Объявление переменных и условные операторы

1. Создайте переменные для хранения вашего имени и возраста.
2. Напишите условие, которое проверяет, достигли ли вы 18 лет. Если да, выведите сообщение "Вы совершеннолетний", если нет — "Вы несовершеннолетний".

Задание 2: Циклы

1. Напишите цикл `for`, который выводит числа от 1 до 10 в консоль.
2. Используйте цикл `while`, чтобы вывести те же числа, но в обратном порядке.

Задание 3: Работа с массивами

1. Создайте два массива: один для хранения тем лекций, другой — для практик.

Пример:

```
const lectures = ['Тема 1', 'Тема 2', 'Тема 3'];
```

```
const practices = ['Практика 1', 'Практика 2', 'Практика 3'];
```

2. Добавьте в массивы по одной новой теме лекции и практике с помощью методов `push()` или `unshift()`.
3. Выведите все темы лекций и практик в консоль с помощью метода `forEach()`.
4. Напишите функцию, которая принимает массив и выводит его элементы в строку через запятую.

Задание 4: Манипуляции с массивами

1. Напишите функцию, которая принимает массив тем лекций и возвращает массив, содержащий только те темы, которые начинаются с буквы "О".
2. Выведите результат работы функции в консоль.

Критерии оценивания

1. Синтаксис (30%): Правильное использование синтаксиса JavaScript.
2. Работа с переменными (20%): Корректное объявление и использование переменных.
3. Условные операторы (20%): Правильное применение условных операторов.
4. Циклы (20%): Корректное использование циклов для перебора данных.
5. Работа с массивами (10%): Умение манипулировать массивами и использовать функции.

Форма сдачи

Студенты должны представить скриншоты выполнения заданий в консоли. Plusом будет предоставление кода с комментариями.