

## Практическая работа № 2

### Теоретический материал по основам JavaScript

#### 1. Подключение JavaScript к HTML

JavaScript может быть встроен непосредственно в HTML-документ или подключён как внешний файл. Это позволяет использовать его для динамического управления поведением веб-страницы.

Встроенный скрипт:

Используется для небольших фрагментов кода.  
Помогает быстро протестировать или внедрить функциональность.

**Пример:**

html

```
<script>
  console.log("Привет, мир!");
</script>
```

Внешний скрипт:

Рекомендуется для больших объемов кода.  
Улучшает организацию и читабельность кода.  
Минимизирует повторяющийся код и облегчает его поддержку.

**Пример:**

html

```
<script src="script.js"></script>
```

#### 2. Переменные

В JavaScript переменные используются для хранения данных, которые можно изменять. В зависимости от необходимости можно использовать:

- **let**: для изменяемых переменных. Значение можно менять.
- **const**: для неизменяемых переменных. Значение не может быть изменено после их объявления.

**Пример:**

**javascript**

```
let name = "JOHN"; // изменяемая переменная  
const age = 25; // неизменяемая переменная
```

### 3. Условия

Условные конструкции позволяют выполнять разные блоки кода в зависимости от заданных условий с помощью `if`, `else if` и `else`.

Использование `if` позволяет проверить ваше условие.  
`else if` предоставляет дополнительные условия.  
`else` выполняется, если ни одно из условий не истинно.

**Пример:**

**javascript**

```
if (age > 18) {  
  console.log("Совершеннолетний");  
} else {  
  console.log("Несовершеннолетний");  
}
```

Также можно использовать тернарный оператор для более компактной записи условия:

**javascript**

```
age>18?console.log("Совершеннолетний"):console.log("Несовершеннолетний")
```

### 4. Циклы

Циклы позволяют повторять блоки кода, что полезно при работе с коллекциями данных (например, массивами).

while: выполняет блок кода, пока условие истинно.

do...while: выполняет блок кода хотя бы один раз.

for: используется для четкого указания счетчика и условий завершения цикла.

**Пример:**

**javascript**

```
for (let i = 0; i < 5; i++) {  
  console.log(i);  
}
```

## 5. Функции

Функции — это блоки кода, которые можно вызывать для выполнения конкретной задачи. Они могут принимать параметры и возвращать значения.

Существует несколько способов объявления функций:

Декларативная функция:

**javascript**

```
function sum(a, b) {  
  return a + b;  
}
```

Функция-выражение:

**javascript**

```
const multiply = function (a, b) {  
  return a * b;  
}
```

Стрелочная функция:

**javascript**

```
const subtract = (a, b) => {  
  return a - b;  
}
```

## 6. Объекты и массивы

Объекты используются для хранения данных в виде пар "ключ-значение".

**Пример:**

javascript

```
let person = {  
  name: "John",  
  age: 30  
};
```

Массивы — это упорядоченные коллекции значений.

**Пример:**

javascript

```
let numbers = [1, 2, 3, 4, 5];
```

## 7. Методы массивов

JavaScript предлагает множество методов для работы с массивами. Вот некоторые из них:

- `push()`: добавляет элемент в конец массива.
- `pop()`: удаляет последний элемент массива.
- `shift()`: удаляет первый элемент массива.
- `unshift()`: добавляет элемент в начало массива.
- `map()`, `filter()`, `reduce()`: позволяют обрабатывать массивы и

возвращать новые массивы или значения.

**Пример:**

javascript

```
let arr = [1, 2, 3];
arr.push(4); // добавляет 4 в конец
console.log(arr); // [1, 2, 3, 4]
let arr = [1, 2, 3];
arr.push(4); // добавляет 4 в конец
console.log(arr); // [1, 2, 3, 4]
```

## Практические задачи

### Задача 1: Проверка возраста

Требование: Написать функцию, которая принимает возраст и возвращает строку "Совершеннолетний" или "Несовершеннолетний".

**javascript**

```
function checkAge(age) {
  if (age >= 18) {
    return "Совершеннолетний";
  } else {
    return "Несовершеннолетний";
  }
}
console.log(checkAge(20)); // "Совершеннолетний"
console.log(checkAge(15)); // "Несовершеннолетний"
```

### Задача 2: Сумма чисел массива

Требование: Написать программу, которая находит сумму всех чисел в массиве.

**javascript**

```
let numbers = [1, 2, 3, 4, 5];
let sum = 0;
numbers.forEach(num => {
  sum += num;
});
console.log(sum); // 15
```

### Задача 3: Факториал числа

Требование: Программа, которая принимает число и возвращает его факториал.

javascript

```
function factorial(n) {  
  if (n === 0) return 1;  
  let result = 1;  
  for (let i = 1; i <= n; i++) {  
    result *= i;  
  }  
  return result;  
}  
console.log(factorial(5)); // 120
```

В одном файле расписать все задачи, можете просто комментировать предыдущие задачи.

### Разогревочные задачи :

1. Площадь прямоугольника  
Написать программу, которая принимает длину и ширину прямоугольника и выводит его площадь.
2. Периметр квадрата  
Написать программу, которая принимает длину стороны квадрата и выводит его периметр.
3. Температура по Фаренгейту  
Написать программу, которая преобразует температуру из Цельсия в Фаренгейты по формуле  $(C * 9/5) + 32$ .
4. Остаток от деления  
Программа, которая принимает два числа и выводит остаток от их деления.
5. Умножение двух чисел  
Написать программу, которая принимает два числа и выводит их произведение.

6. Проверка на четность  
Написать программу, которая принимает число и выводит, является ли оно четным.
  7. Перевод времени  
Программа, которая принимает количество часов и преобразует их в минуты.
  8. Сравнение двух чисел  
Программа, которая принимает два числа и выводит, какое из них больше или они равны.
  9. Возраст пользователя  
Программа, которая принимает год рождения пользователя и выводит его возраст.
  10. Простое сложение  
Программа, которая принимает два числа и выводит их сумму.
- 

## Циклы :

1. Вывести числа от 1 до 10  
Используя цикл, вывести на экран числа от 1 до 10.
  2. Сумма чисел от 1 до N  
Написать программу, которая принимает число n и находит сумму чисел от 1 до N.
  3. Факториал числа  
Программа, которая принимает число n и выводит его факториал (N!).
  4. Вывод четных чисел  
Используя цикл, вывести все четные числа от 1 до 20.
  5. Таблица умножения на 5  
Написать программу, которая выводит таблицу умножения на 5 (5x1, 5x2, ..., 5x10).
  6. Сумма чисел кратных 3  
Программа, которая находит сумму чисел от 1 до 50, которые кратны 3.
  7. Обратный отсчёт  
Программа, которая выводит числа от 10 до 1 в обратном порядке.
  8. Произведение чисел от 1 до N  
Написать программу, которая принимает число n и находит произведение всех чисел от 1 до N.
  9. Числа, кратные 7  
Программа, которая выводит все числа от 1 до 100, кратные 7.
  10. Печать символов  
Программа, которая принимает символ и число n, и выводит этот символ n раз.
-

## Условия:

1. Проверка числа на положительность  
Программа, которая принимает число и выводит "положительное", если оно больше 0, или "отрицательное", если меньше 0.
2. Оценка по баллам  
Программа, которая принимает количество баллов от 0 до 100 и выводит оценку (например, больше 90 — "отлично", больше 70 — "хорошо", и т.д.).
3. Проверка деления на 3  
Программа, которая принимает число и выводит "кратно 3", если число делится на 3 без остатка, иначе "не кратно".
4. Четное или нечетное  
Программа, которая проверяет, является ли введенное число четным или нечетным.
5. Наибольшее из двух чисел  
Программа, которая принимает два числа и выводит наибольшее из них.
6. Проверка високосного года  
Программа, которая принимает год и выводит, является ли он високосным.
7. Проверка температуры  
Программа, которая принимает температуру и выводит, нужно ли надеть куртку (например, если температура ниже 10 градусов).
8. Число делится на 5 и 11  
Программа, которая проверяет, делится ли число одновременно на 5 и 11.
9. Равны ли числа  
Программа, которая принимает два числа и выводит, равны они или нет.
10. Проверка символа  
Программа, которая принимает символ и проверяет, является ли он буквой (a-z или A-Z).

## Функции:

1. Сложение двух чисел  
Написать функцию, которая принимает два числа и возвращает их сумму.
2. Умножение двух чисел  
Написать функцию, которая принимает два числа и возвращает их произведение.
3. Проверка четности числа  
Написать функцию, которая принимает число и возвращает `true`, если число четное, и `false`, если нет.
4. Факториал числа  
Написать функцию, которая принимает число `n` и возвращает его факториал.



5. Проверка деления на 3  
Написать функцию, которая принимает число и возвращает `true`, если число делится на 3, и `false` в противном случае.
  6. Перевод градусов Цельсия в Фаренгейты  
Написать функцию, которая принимает температуру в Цельсиях и возвращает её в Фаренгейтах.
  7. Возведение в степень  
Написать функцию, которая принимает число и степень, в которую нужно его возвести, и возвращает результат.
  8. Минимум из двух чисел  
Написать функцию, которая принимает два числа и возвращает наименьшее из них.
  9. Максимум из двух чисел  
Написать функцию, которая принимает два числа и возвращает наибольшее из них.
  10. Проверка возраста  
Написать функцию, которая принимает возраст и возвращает строку "Совершеннолетний" или "Несовершеннолетний" в зависимости от значения.
- 

## **Массивы (использование методов `forEach`, `find`, `push`, `unshift`, `map`, `pop`, `shift`, `concat`, `splice`, `slice`, `join`, `split`):**

1. Вывести каждый элемент массива  
Использовать `forEach` для вывода каждого элемента массива на экран.
2. Найти первый четный элемент массива  
Использовать `find` для нахождения первого четного числа в массиве.
3. Добавить элемент в конец массива  
Использовать `push` для добавления нового элемента в конец массива.
4. Добавить элемент в начало массива  
Использовать `unshift` для добавления нового элемента в начало массива.
5. Удвоить значения всех элементов массива  
Использовать `map`, чтобы вернуть новый массив, где каждый элемент удвоен.
6. Удалить последний элемент массива  
Использовать `pop`, чтобы удалить последний элемент массива.
7. Удалить первый элемент массива  
Использовать `shift`, чтобы удалить первый элемент массива.
8. Объединить два массива  
Использовать `concat`, чтобы объединить два массива в один.
9. Удалить элемент из массива по индексу  
Использовать `splice` для удаления элемента из массива по указанному индексу.

10. Извлечь подмассив  
Использовать `slice` для получения подмассива.
  11. Преобразовать массив в строку
  12. Преобразовать строку в массив
  13. Сумма всех чисел в массиве  
Использовать `forEach`, чтобы найти сумму всех элементов массива.
  14. Увеличить каждое число в массиве на 10  
Использовать `map`, чтобы увеличить каждое число в массиве на 10.
  15. Найти индекс первого отрицательного числа  
Использовать `find` для нахождения первого отрицательного числа в массиве.
- 

## Объекты :

1. Создать объект с именем и возрастом  
Создать объект, который содержит свойства `name` и `age`, и вывести их значения.
2. Добавить новое свойство в объект  
Создать объект и добавить к нему новое свойство `city`.
3. Удалить свойство из объекта  
Создать объект и удалить из него свойство `age`.
4. Изменить значение свойства  
Создать объект и изменить значение свойства `name`.
5. Проверка наличия свойства  
Написать функцию, которая принимает объект и имя свойства, и проверяет, существует ли это свойство в объекте.
6. Проверка значения свойства  
Написать функцию, которая принимает объект и имя свойства, и возвращает значение этого свойства.
7. Сравнение двух объектов  
Написать функцию, которая сравнивает два объекта по значению их свойств.
8. Итерация по свойствам объекта  
Использовать цикл `for...in`, чтобы вывести все свойства объекта

## Заключение

Подведем итоги: JavaScript является мощным инструментом для создания интерактивных веб-приложений. Понимание переменных, условий, циклов,

функций, объектов и массивов поможет вам успешно решать задачи и создавать динамичные интерфейсы. Каждая из приведённых практических задач приносит практический опыт работы с основными концепциями языка, что необходимо для дальнейшего изучения.