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.

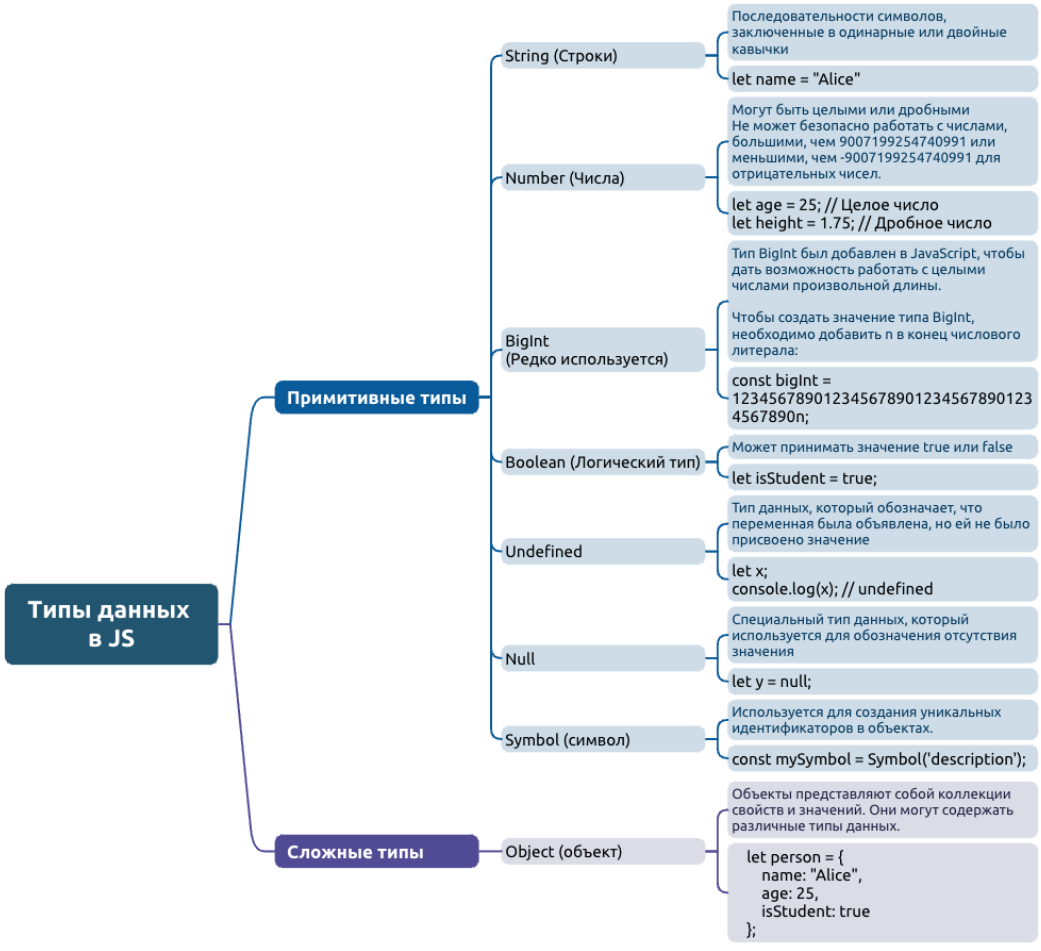
Практическое занятие 9. Основы синтаксиса JavaScript, запуск кода, работа с переменными, логическими операторами и консолью браузера

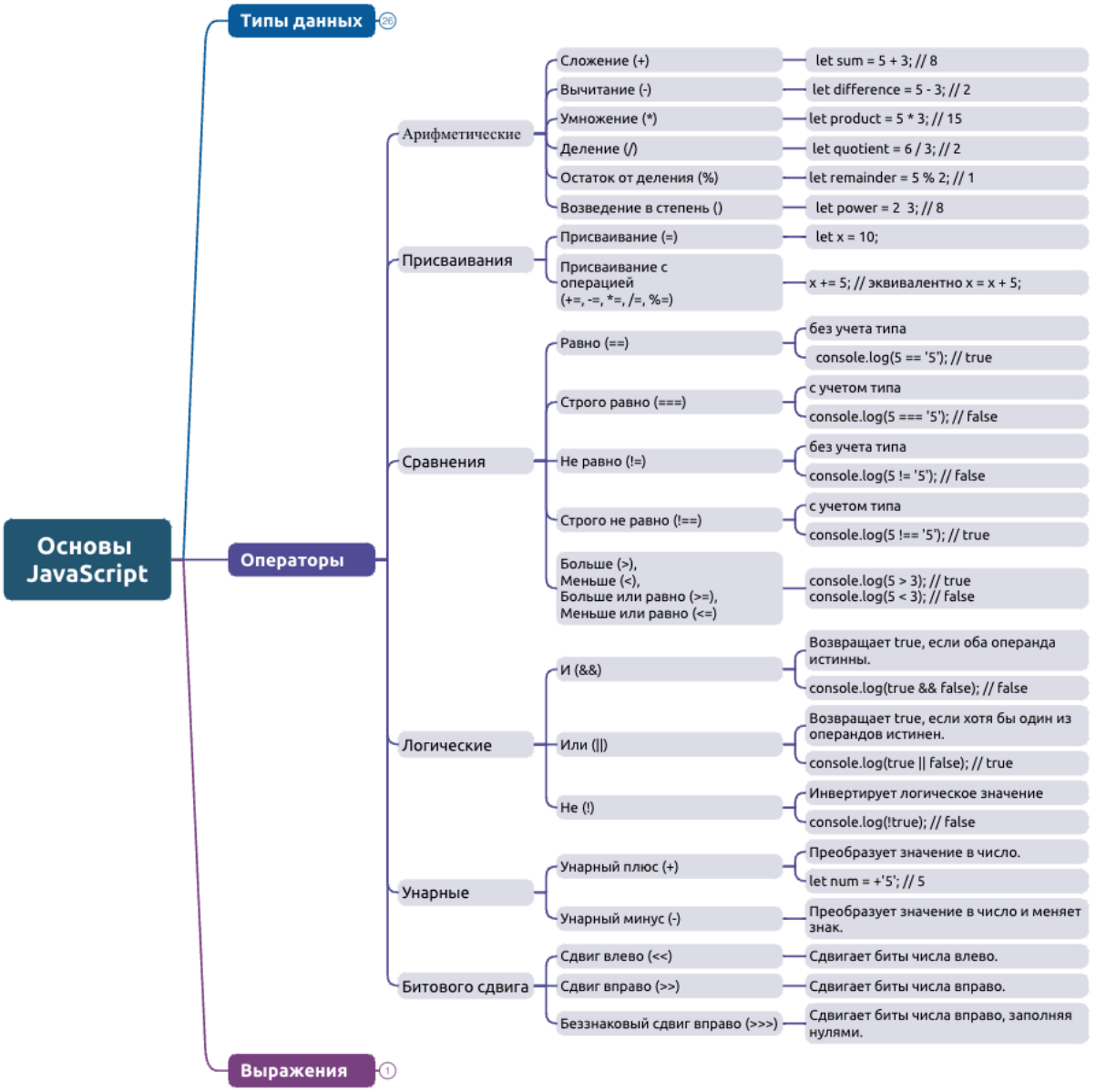
Изучаемые вопросы

1. Базовый синтаксис JavaScript.
2. Способы объявления переменных.
3. Основные команды: console.log(), alert(), prompt().
4. Работа с логическими операторами (&&, ||, !).
5. Работа с консолью браузера.

Краткая теория

JavaScript — это язык программирования, который позволяет создавать интерактивные элементы на веб-страницах. Он имеет простой и понятный синтаксис и поддерживает различные типы данных, такие как строки, числа, логические значения, массивы и объекты.





Объявление переменных:

- var: устаревший способ, используется для глобальных переменных.

- let: современный способ, ограничивает область видимости переменной блоком.

- const: используется для объявления констант.

Логические операторы:

- &&: логическое "И". Возвращает true, если оба операнда истинны.

- ||: логическое "ИЛИ". Возвращает true, если хотя бы один из операндов истинен.

- !: логическое "НЕ". Инвертирует логическое значение.

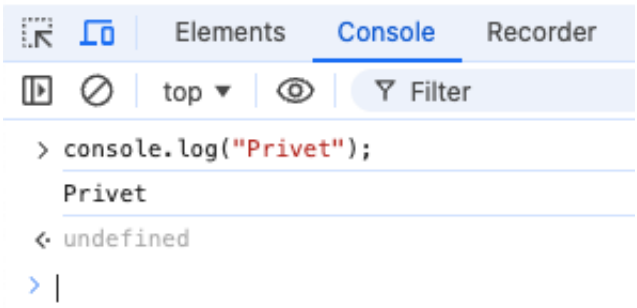
Основные команды:

- console.log(): выводит информацию в консоль браузера.

- alert(): показывает всплывающее окно с сообщением.

- prompt(): запрашивает ввод у пользователя.

Пример использования **console.log() (**используется для вывода информации в консоль).



- console.error() Используется для вывода ошибок в консоль.

- console.warn() Используется для вывода предупреждений.

- console.table() Позволяет выводить данные в виде таблицы, что особенно удобно для объектов и массивов.

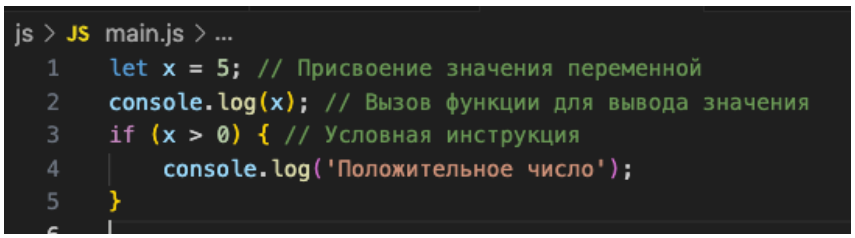
Синтаксис JavaScript: переменные, типы данных, работа с консолью

Структура JS кода

Инструкции и функции в JavaScript — это разные концепции, хотя они могут пересекаться. Давайте разберёмся, в чём их различия:

Инструкция — это отдельная команда или выражение, которое выполняет определённое действие. Например, присвоение значения переменной, вызов функции или условный оператор.

Пример



Функция — это блок кода, который можно вызывать многократно. Она может принимать параметры и возвращать значение.

Примеры

Основные различия JS инструкций и функций

1. Назначение.

- Инструкции выполняют действия.

- Функции группируют код для повторного использования.

2. Структура.

- Инструкции могут быть простыми (например, присвоение) или сложными (например, условные конструкции).

- Функции имеют своё собственное тело и могут содержать несколько инструкций.

3. Повторное использование.

- Инструкции выполняются один раз, когда они достигаются в коде.

- Функции могут быть вызваны много раз из разных мест в коде.

Таким образом, инструкции и функции — это разные элементы языка JavaScript. Инструкции — это базовые единицы выполнения кода, а функции — это более сложные конструкции, позволяющие организовать и повторно использовать код.

Инструкции и комментарии в JavaScript

Инструкции — это команды, выполняющие действия. Например, alert('Привет, мир!') показывает сообщение. Инструкции можно разделять точками с запятой, но обычно их записывают на новых строках для удобства чтения:

Точку с запятой можно не ставить, если следующая инструкция начинается с новой строки. Однако в некоторых случаях это может привести к ошибкам, как в примере:

```
js > JS main.js
1   alert('Hello')
2   [1, 2].forEach(alert);
3
```

Здесь отсутствие точки с запятой вызывает ошибку, так как JavaScript объединяет инструкции.

Рекомендуется всегда ставить точки с запятой для избежания ошибок.

Комментарии полезны для временного отключения кода и улучшения его читаемости.

Комментарии в JavaScript

1. Однострочные комментарии:

- Используются для комментирования одной строки кода.
- Синтаксис: //

```
js > JS main.js
1   // Это однострочный комментарий
2   alert('Привет'); // Этот код выводит приветствие
3
```

2. Многострочные комментарии:

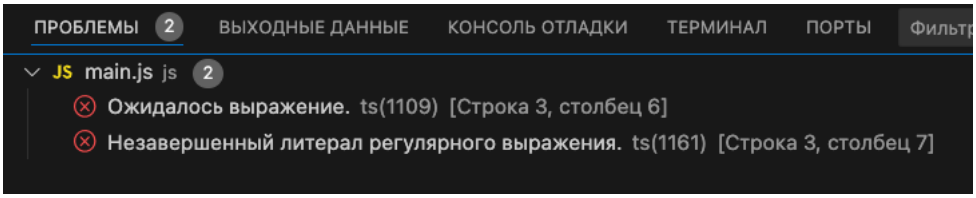
- Используются для комментирования нескольких строк кода.
- Синтаксис: /* ... */

```
js > JS main.js
1   /*
2    Это многострочный комментарий.
3    Он может занимать несколько строк.
4    */
5   alert('Мир');
```

3. Вложенные комментарии:

- Как вы правильно заметили, вложенные многострочные комментарии не поддерживаются:

```
js > JS main.js
1   /*
2     /* Это не сработает! */
3     */
4
```



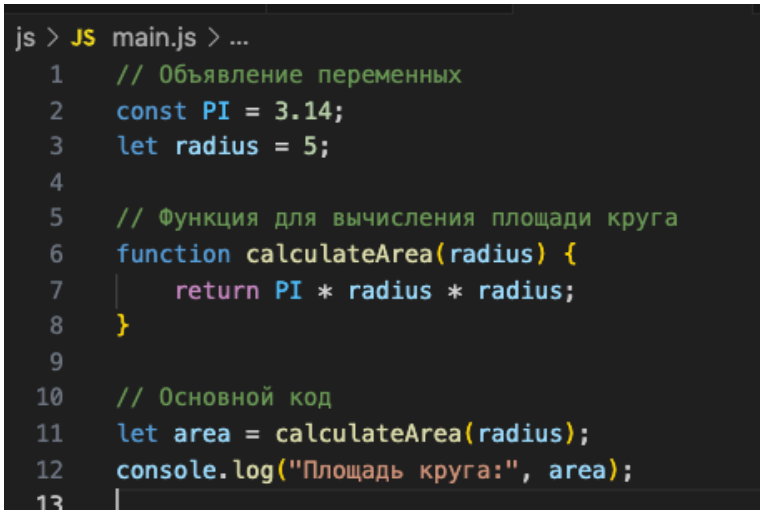
Структура JavaScript кода:

1. Объявление переменных и констант. В начале файла или функции объявляются все необходимые переменные.

2. Функции. Определение функций, которые будут использоваться в коде.

3. Основной код. Логика программы, включая условные конструкции и циклы.

4. Обработка событий. Если это клиентский код, добавляются обработчики событий для взаимодействия с пользователем.



Операторы в JavaScript

Операторы в JavaScript делятся на несколько категорий:

1.1. Арифметические операторы

Эти операторы используются для выполнения математических операций.

- Сложение (+). Складывает два числа.

```
let sum = 5 + 3; // 8
```

- Вычитание (-). Вычитает одно число из другого.

```
let difference = 5 - 3; // 2
```

- Умножение (*). Умножает два числа.

```
let product = 5 * 3; // 15
```

- Деление (/). Делит одно число на другое.

```
let quotient = 6 / 3; // 2
```

- Остаток от деления (%). Возвращает остаток от деления.

```
let remainder = 5 % 2; // 1
```

- Возведение в степень (). Возводит число в степень.

```
let power = 2  3; // 8
```

1.2. Операторы присваивания

Эти операторы используются для присвоения значений переменным.

- Присваивание (=). Присваивает значение переменной.

```
let x = 10;
```

- Присваивание с операцией (+=, -=, *=, /=, %=). Выполняет операцию и присваивает результат.

```
x += 5; // эквивалентно x = x + 5;
```

1.3. Операторы сравнения

Эти операторы сравнивают значения и возвращают логическое значение (true или false).

- Равно (==). Сравнивает два значения на равенство (без учета типа).

```
console.log(5 == '5'); // true
```

- Строго равно (===). Сравнивает два значения на равенство (с учетом типа).

```
console.log(5 === '5'); // false
```

- Не равно (!=). Сравнивает два значения на неравенство (без учета типа).

```
console.log(5 != '5'); // false
```

- Строго не равно (!==). Сравнивает два значения на неравенство (с учетом типа).

```
console.log(5 !== '5'); // true
```

- Больше (>), Меньше (<), Больше или равно (>=), Меньше или равно (<=).

```
console.log(5 > 3); // true
console.log(5 < 3); // false
```

1.4. Логические операторы

Эти операторы используются для выполнения логических операций.

- И (&&). Возвращает true, если оба операнда истинны.

```
console.log(true && false); // false
```

- Или (||). Возвращает true, если хотя бы один из операндов истинен.

```
console.log(true || false); // true
```

- Не (!). Инвертирует логическое значение.

```
console.log(!true); // false
```

1.5. Унарные операторы

Эти операторы работают с одним операндом.

- Унарный плюс (+). Преобразует значение в число.

```
let num = +'5'; // 5
```

- Унарный минус (-). Преобразует значение в число и меняет знак.

```
let neg = -5; // -5
```

1.6. Операторы битового сдвига

Эти операторы выполняют операции со значениями на уровне битов.

- Сдвиг влево (<<). Сдвигает биты числа влево, добавляя нули справа.

```
let a = 5; // В двоичном представлении: 0000 0101
let result = a << 1; // Сдвигаем на 1 бит влево
console.log(result); // 10 (0000 1010)
```

- Сдвиг вправо (>>). Сдвигает биты числа вправо. Если число положительное, то слева добавляются нули. Если число отрицательное, то слева добавляются единицы (знак числа сохраняется).

```
let b = 20; // В двоичном представлении: 0001 0100
let result3 = b >> 1; // Сдвигаем на 1 бит вправо
console.log(result3); // 10 (0000 1010)
let c = -20; // В двоичном представлении: 1110 1100 (в формате дополнительного кода)
let result4 = c >> 1; // Сдвигаем на 1 бит вправо
console.log(result4); // -10 (1111 0110)
```

- Беззнаковый сдвиг вправо (>>>). Сдвигает биты числа вправо, заполняя нулями.

```
let d = -20; // В двоичном представлении: 1110 1100 (в формате дополнительного кода)
let result5 = d >>> 1; // Сдвигаем на 1 бит вправо без знака
console.log(result5); // 2147483638 (в двоичном представлении: 0111 1111 ... 1110)
```

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

Codecademy — JavaScript. Интерактивные уроки по JavaScript (на английском).

https://www.codecademy.com/learn/introduction-to-javascript

Задания

Задание 1: Приветствие

1. Откройте инструменты разработчика в вашем браузере (обычно F12) и перейдите на вкладку "Консоль".

2. Напишите программу, которая:

- Запрашивает у пользователя его имя с помощью prompt().
- Запрашивает у пользователя его возраст.
- Выводит приветственное сообщение с использованием alert(), в котором будет указано имя и возраст пользователя.
- Выводит ту же информацию в консоль с помощью console.log().

3. Создайте переменные для хранения имени и возраста, используйте let или const для объявления переменных.

Задание 2. Проверка возраста

1. Расширьте программу из задания 1:

- После приветствия, проверьте, является ли пользователь совершеннолетним (18 лет и старше).
- Если да, выведите сообщение "Вы совершеннолетний", если нет — "Вы несовершеннолетний".

- Используйте логический оператор для проверки возраста.

Задание 3. Угадай число

1. Напишите программу, которая:

- Генерирует случайное число от 1 до 10 и сохраняет его в переменной.

- Запрашивает у пользователя угадать это число с помощью prompt().

- Если пользователь угадал, выводите сообщение "Поздравляем! Вы угадали число!".

- Если не угадал, проверьте, было ли введенное число меньше 5 или больше 5 и выведите соответствующее сообщение.

Задание 4. Проверка пароля

1. Напишите программу, которая:

- Запрашивает у пользователя пароль с помощью prompt().

- Проверяет, совпадает ли введенный пароль с заранее заданным (например, "12345").

- Если пароли совпадают, выводите сообщение "Доступ разрешен".

- Если не совпадают, выводите сообщение "Доступ запрещен".

- Добавьте проверку на то, что введенный пароль не пустой с использованием логического оператора.

Задание 5. Простой калькулятор

1. Напишите программу, которая:

- Запрашивает у пользователя два числа и оператор (+, -, *, /) с помощью prompt().

- Выполняет соответствующее арифметическое действие в зависимости от введенного оператора.

- Если оператор неверный, выводите сообщение "Неверный оператор".

- Используйте логические операторы для проверки корректности ввода.

Критерии оценивания

1. Синтаксис (30%). Правильное использование синтаксиса JavaScript.

2. Работа с переменными (30%). Корректное объявление и использование переменных.

3. Использование команд (20%). Правильное применение команд alert(), prompt() и console.log().

4. Логические операторы (20%). Корректное использование логических операторов для проверки условий.

Форма сдачи

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