

Введение в программирование на языке JavaScript

JavaScript делает сайты живыми: от кликов и форм до логики программ.

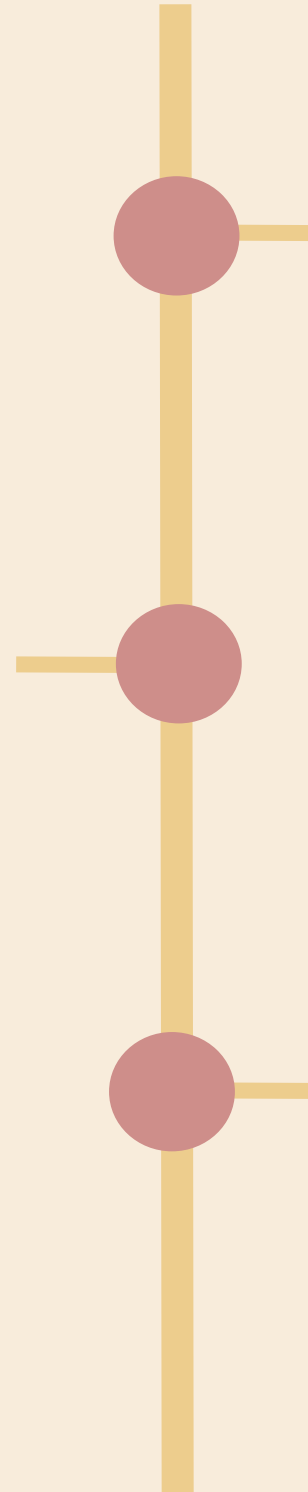
Почему JavaScript выбирают для школьного курса

На уроках информатики важно видеть результат сразу. В JavaScript кнопки реагируют на нажатия, формы проверяют ввод, а текст меняется без перезагрузки страницы — всё на понятных примерах.



Программа и алгоритм

Алгоритм — это точная последовательность действий для решения задачи. Если перепутать порядок шагов, ответ может стать неверным, даже при правильных командах.



Программа — это набор команд, записанных на языке программирования. В каждом учебном примере команды связаны одной целью: получить нужный результат на экране или в ответе.

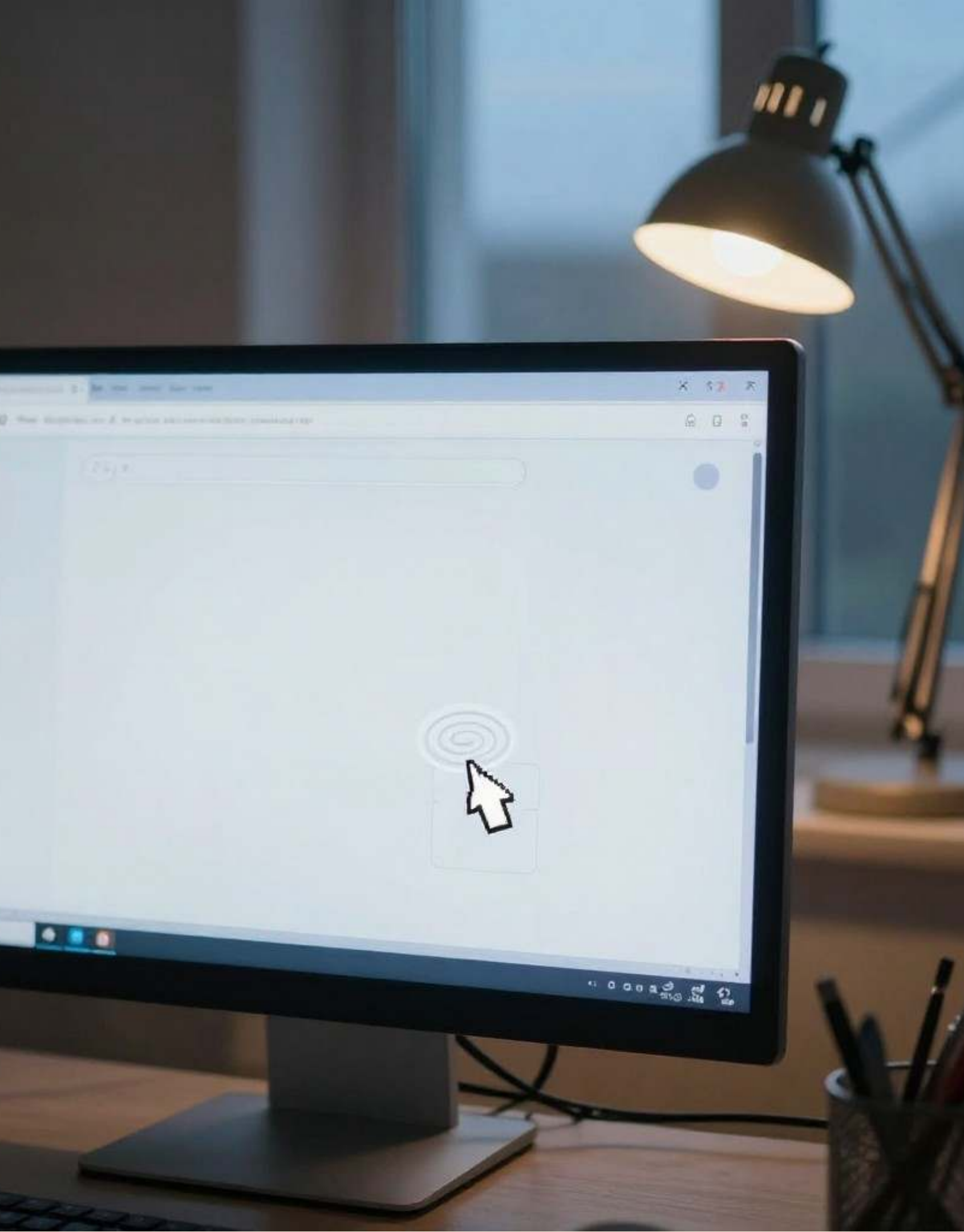
В школьных заданиях алгоритмы встречаются повсюду: посчитать сумму, проверить правильность ответа, найти наибольшее значение и повторить одно действие много раз.

Из чего состоит JavaScript-код

Код складывается из небольших элементов, которые вместе образуют программу: переменные хранят данные, операторы выполняют действия, функции объединяют шаги, а комментарии помогают объяснять.

Программа получается из связанных «кирпичиков»: данные → действия → объединение шагов в блоки.

Элемент	Назначение	Пример	Роль в программе
Переменная	Хранит данные	<code>let age = 14;</code>	Используется в вычислениях
Оператор	Выполняет действие/ сравнение	<code>+, -, =, ===</code>	Получается вычисление или проверка
Комментарий	Поясняет код	<code>// счётчик</code>	Не влияет на выполнение
Функция	Группирует команды	<code>function greet() { ... }</code>	Повторно используется



Где работает JavaScript

В браузере: интерактивность страницы

JavaScript выполняется в браузере и меняет страницу без полной перезагрузки. Он реагирует на клики, ввод текста, движения мыши и нажатия клавиш, то есть делает интерфейс «живым».

На сервере: обработка запросов

На сервере JavaScript помогает обрабатывать запросы пользователей и работать с данными. Благодаря этому можно, например, принимать форму, проверять информацию и отправлять результат обратно на страницу.

Как запускается скрипт JavaScript в браузере

Цепочка связывает текст программы, действия пользователя и изменения на экране.

5

Эти данные можно представить
в виде пяти основных
категорий.

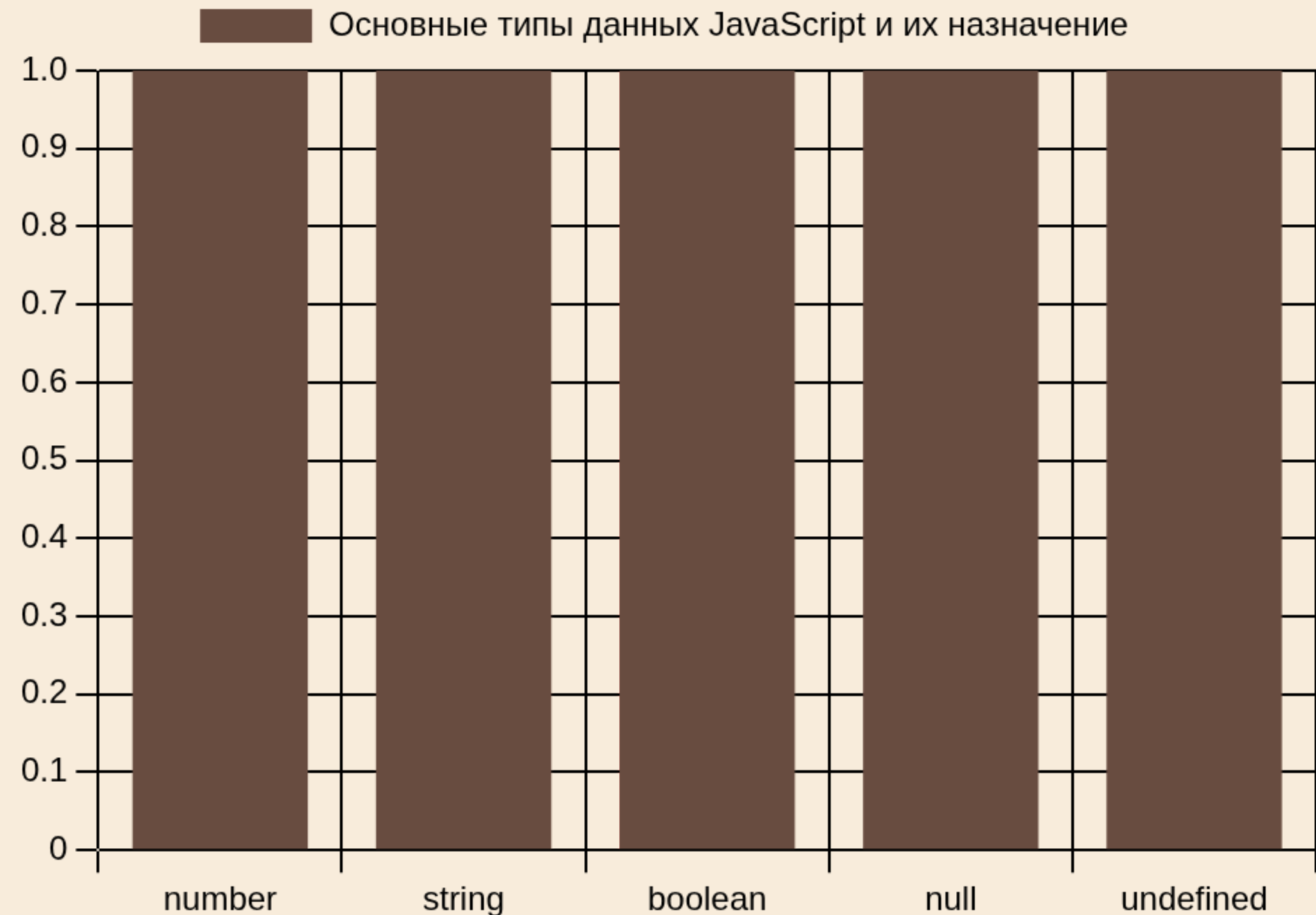
*Учебный материал по основам JavaScript (типизация для
школьного уровня).*

Переменная хранит значение и имеет имя; в JavaScript её создают через `let` и `const`. Типы данных подсказывают, какие значения допустимы и как с ними работать.

Основные типы данных в JavaScript

Типы помогают программе не путать, например, текст и число: вычисления выполняются с `number`, а сравнения логикой возвращают `true` или `false`.

Разные типы служат разным задачам: от хранения текста и значений до обозначения отсутствия данных.



Операторы и выражения



Арифметика

Операторы вычислений в JavaScript: `+`, `-`, `*`, `/`, `%`. С их помощью формируют выражения вроде суммы или процента, чтобы получить новое числовое значение.



Сравнение

Операторы сравнения определяют, что верно или неверно: `>`, `<`, `>=`, `<=`, `===`, `!==`. При сравнении результатом становится логическое значение `true` или `false`.



Логика

Логические операторы `&&`, `||` и `!` помогают строить условия. Например, можно проверить сразу несколько требований или изменить смысл условия с помощью отрицания.



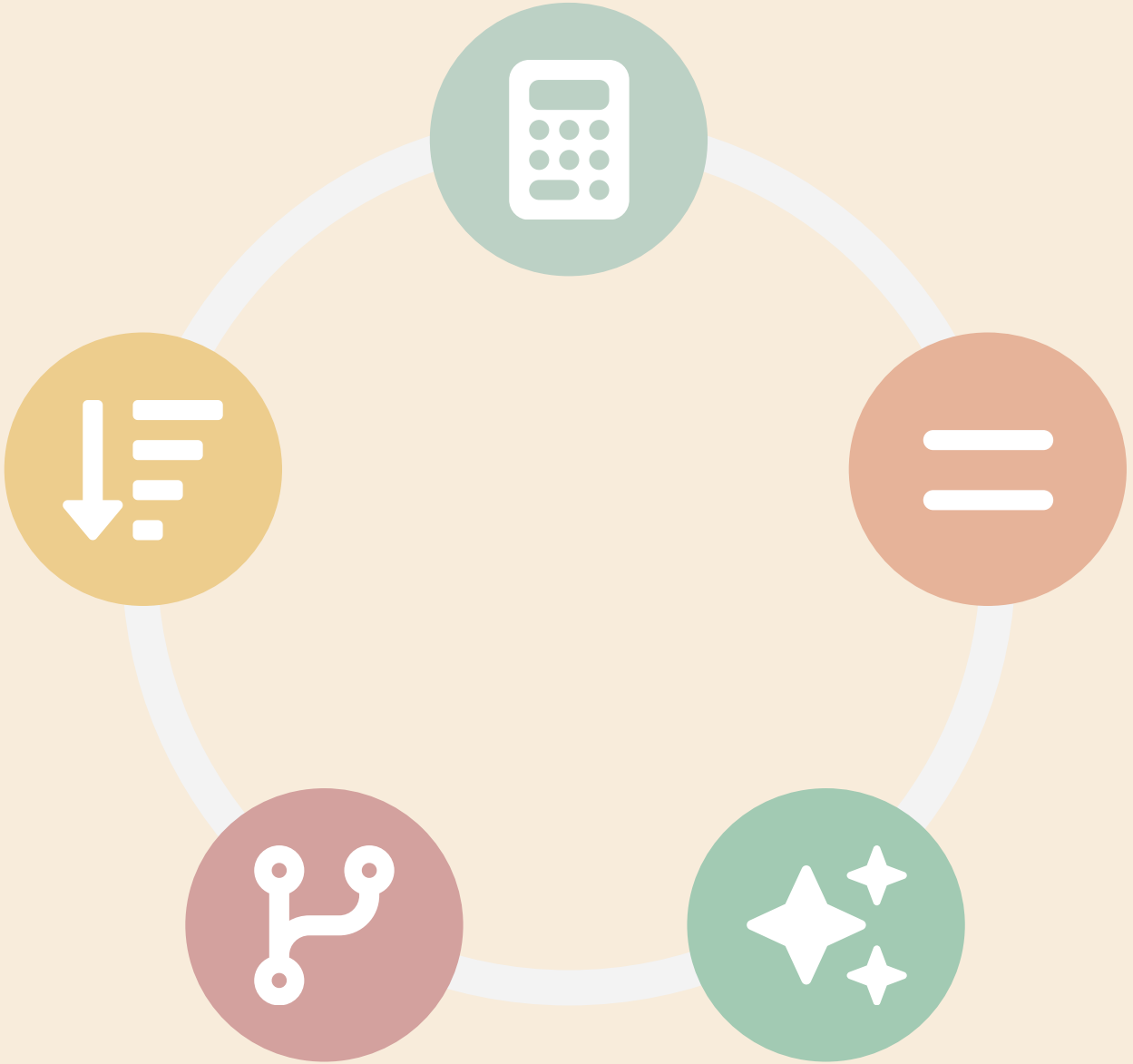
Приоритет

Когда в выражении много операций, порядок вычислений важен. Скобки часто используют, чтобы явно указать, что вычислять первым, и избежать ошибок.



Выражение даёт результат

Выражение — это запись, которая после вычисления возвращает значение. Его можно выводить, сравнивать или использовать дальше в условии и цикле.



Условный оператор if...else

Условный оператор помогает выбирать вариант действий. Если условие истинно, выполняется один блок команд, а если ложно — другой, и программа «поворачивает» в нужную сторону.

01

Примером служит определение результата теста: после вычисления количества баллов условие решает, выводить «прошёл» или «не прошёл». Так же работает проверка минимального порога.

03

02

В задачах школьного уровня условие обычно строится через сравнение значений или логическое выражение. Например, проверка пароля сводится к тому, совпадает ли введённый текст.

04

Такая конструкция делает программу гибкой: одно и то же приложение обрабатывает разные данные по-разному. Важно следить за корректными условиями и за тем, какой блок выполнится при true и false.

Проверка условия в программе

Ветвление зависит от результата проверки данных: true запускает один блок, false — другой.

Циклы for и while

Удобен, когда заранее известно количество повторений. Обычно он включает счётчик, условие продолжения и шаг, например увеличение числа после каждой итерации.

⋮

Цикл for

· ·

Цикл while

· ·

Где применяют на практике

⋮

Для вывода чисел по порядку, подсчёта очков, повторной проверки ответов и построения списка элементов на странице.

⋮

Подходит, когда нужно повторять действие, пока выполняется условие. Команды могут выполняться ноль раз, если условие сразу ложно.

Сколько повторений заменяет цикл

Чем больше одинаковых действий в задаче, тем заметнее выгода: цикл превращает длинный «копипаст» в одну понятную конструкцию.

По значениям видно, что ручное повторение быстро разрастается. Цикл помогает менять логику в одном месте, а количество повторов легко регулировать.



Примеры для сравнения (1, 5, 10, 20 повторений)

Функции: повторное использование кода

01

Функция — это именованный блок кода для отдельной задачи. Она помогает не дублировать одинаковые действия и делает программу короче, а смысл — понятнее.

02

Функции могут принимать параметры и возвращать результат. Например, можно передавать число, получать сумму, а затем использовать результат в дальнейшем коде или выводе на страницу.

03

Одну и ту же функцию можно вызвать много раз с разными значениями. Так обрабатываются разные пользователи, вопросы викторины или варианты ответов, без переписывания логики.

Массивы и объекты



Массив: упорядоченный набор значений

Массив хранит несколько значений в определённом порядке, например список оценок, имён или чисел. К значениям обычно обращаются по индексу, поэтому удобно выполнять циклы по элементам.

Объект: данные с «полями»

Объект хранит связанные свойства и значения: имя ученика, класс, возраст и любимый предмет. Так данные «живут вместе», как карточка, и их можно логично использовать в программе.

События в браузере и их обработка

Событие	Что происходит	Пример	Значение для страницы
click	Пользователь нажимает элемент	Нажатие кнопки	Запуск счётчика или открытие меню
input	Меняется текст в поле	Ввод в строке поиска	Поиск или проверка формы
submit	Форма отправляется	Регистрация в анкете	Данные передаются для обработки
keydown	Нажата клавиша	Управление в игре	Движение или действие по клавиатуре

События помогают странице реагировать на действия пользователя: нажатия, ввод и отправку форм.

Одинаковые события дают разные эффекты, потому что обработчик связывает действие пользователя с конкретным кодом: обновить текст, проверить ввод или отправить данные.

Типичные ошибки начинающих

1

Часто пропускают скобки или кавычки: из-за этого браузер не понимает границы выражений и считает код «незавершённым». Ошибка почти всегда проявляется при запуске и подсвечивается в консоли.

2

Ещё одна частая ловушка — путаница между `=` и `===`. Первый знак присваивает значение, а второй сравнивает строго, поэтому результат проверки условия может стать неверным.

3

Иногда допускают опечатки в имени переменной: например, создали `score`, а дальше использовали `scores`. Тогда программа не найдёт нужное значение и остановит выполнение с ошибкой.

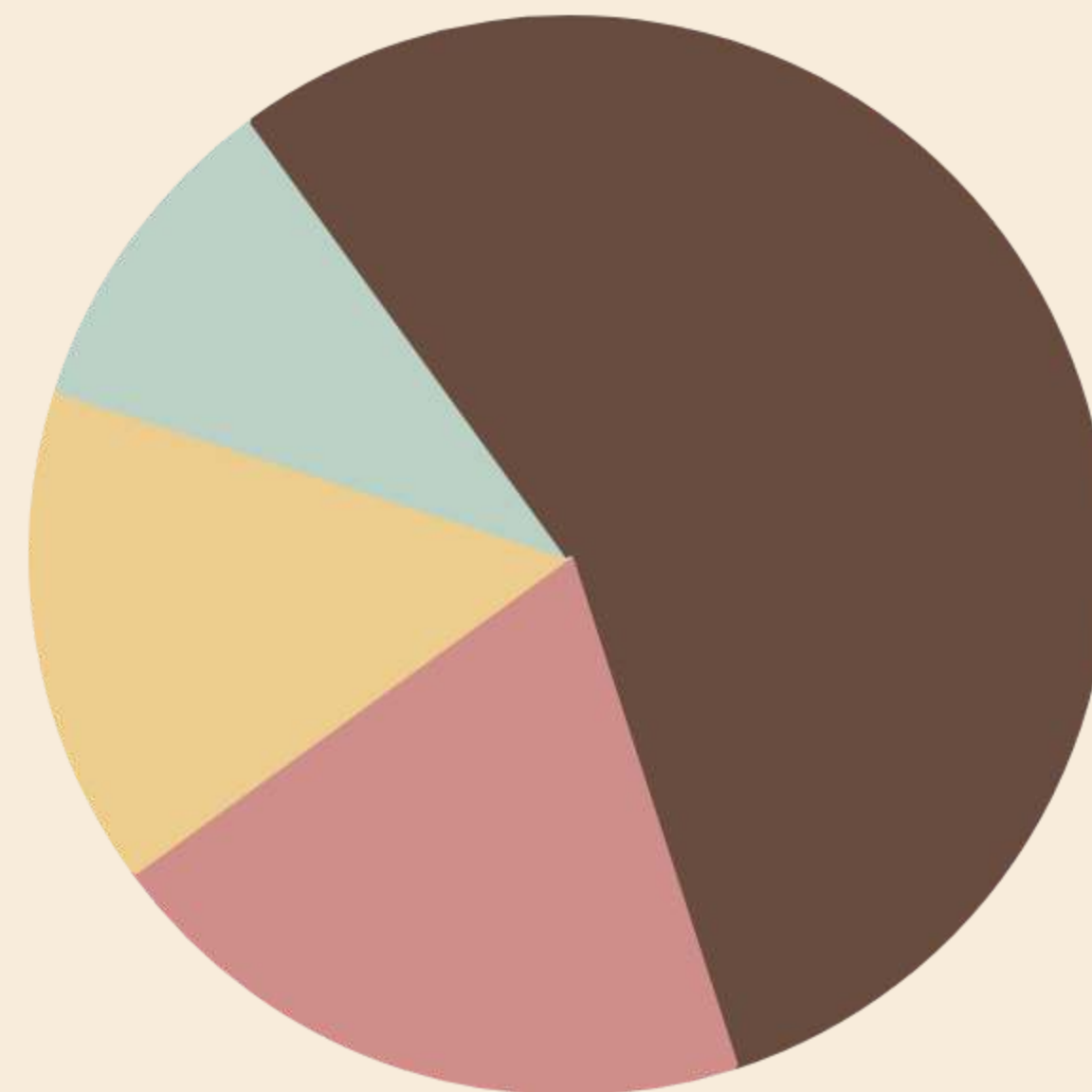
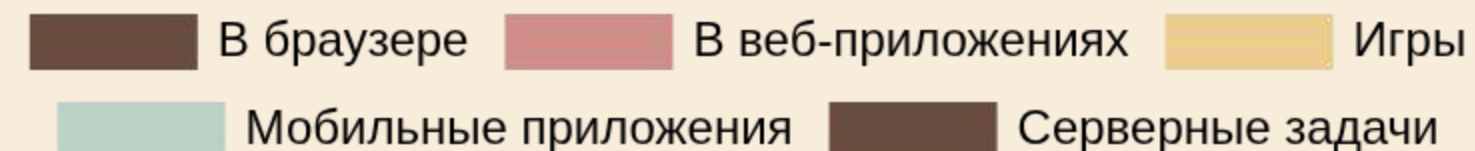
4

При отладке полезно читать сообщение об ошибке в консоли и искать строку, где она возникла. Для коротких программ помогают выводы через `console.log`, чтобы проверить, что вы считаете правильно.

Где применяется JavaScript

JavaScript особенно заметен там, где нужен отклик на действия пользователя: страницы становятся интерактивными. Также язык используют в веб-среде и для серверной обработки данных.

Наибольшая доля приходится на работу в браузере: интерактивность создаётся именно обработчиками событий. Остальные области дополняют это применение и расширяют возможности языка.



Обобщение по типичным школьным и образовательным сценариям (без точных статистических процентов)

Мини-проект по теме урока



Кнопка-счётчик

Создайте страницу с кнопкой и переменной счёта. При click увеличивайте значение, а затем меняйте текст на экране, чтобы результат обновлялся сразу.



Простая викторина

Добавьте вопрос и поле ответа. Условие проверит правильность, а функция обновит сообщение: «верно» или «попробуй ещё раз».



Переключение режима

Сделайте переключатель «день/ночь» или «режим обучения». В зависимости от условия меняйте стиль и подписи, используя функции для аккуратной замены текста.



Проверка ответа на вопрос

Добавьте поле ввода и кнопку. Если пользователь ввёл подходящее значение, функция показывает результат, иначе — подсказку; так тренируется ветвление if...else.

JavaScript — это шаг от идеи к работающему коду

JavaScript помогает создавать интерактивные веб-страницы и учит базовым конструкциям: программа и алгоритм, переменные, типы данных, операторы, условия, циклы, функции, массивы, объекты и события.