№79

Логический тип Ошибка в задаче №1 и задаче №2 - это выход за границы массива. В JavaScript индексация массива начинается с 0, поэтому последний элемент массива имеет индекс `arr.length - 1`, а не `arr.length`, как указано в этих задачах. Таким образом, код попытается обратиться к элементу за пределами массива, что приведет к получению `undefined`.

Синтаксический тип Ошибка в задаче №3 - это попытка создать объект с помощью неправильного синтаксиса. Вместо фигурных скобок для создания массива использовались фигурные скобки, что является некорректным для создания массива в JavaScript.

Синтаксический тип Ошибка в задаче №4 - это опечатка в слове `length`. Необходимо написать `length`, а не `lenght`.

№103

Синтаксический тип Ошибка в задаче №1 и задаче №3 - это использование неправильного синтаксиса для доступа к свойствам объекта. В JavaScript, когда вы обращаетесь к свойствам объекта, вы должны использовать квадратные скобки `[]` или точечную нотацию `.` и указывать имя свойства в виде строки. В задачах №1 и №3 использована переменная `x` без кавычек, что пытается обратиться к свойству объекта с именем, равным значению переменной `x`, а не к свойству с именем `'x'`.

Логический тип Ошибка в задаче №2 - это попытка обратиться к свойству объекта, используя переменную, содержащую имя ключа. Вместо этого необходимо использовать квадратные скобки и передать имя ключа в виде строки, чтобы обратиться к свойству объекта.

Логический тип Ошибка в задаче №4 - это попытка обратиться к свойству `length`, которого нет в объекте `obj`. В JavaScript объекты не имеют встроенного свойства `length`, поэтому вы не можете получить количество свойств объекта напрямую таким образом.

Тип ошибки в этих задачах - это ошибки синтаксиса и ошибки доступа к свойствам объектов в JavaScript. Вот их подробное описание:

1. Ошибка в №1 и №3: Обращение к свойствам объекта должно быть сделано с использованием квадратных скобок или точечной нотации, и имя свойства должно быть указано в виде строки. В коде указано `obj[x]`, что пытается обратиться к свойству объекта с именем, равным значению переменной `x`, а не к свойству с именем `'x'`.

2. Ошибка в №2: При обращении к свойству объекта по ключу, содержащемуся в переменной, нужно использовать квадратные скобки и передавать имя ключа в виде строки. В коде указано `obj.key`, что пытается обратиться к свойству с именем `'key'`, а не к свойству, имя которого содержится в переменной.

Логический тип 3. Ошибка в №4: Объекты в JavaScript не имеют встроенного свойства `length`, поэтому попытка получить его значение приведет к ошибке. Вместо этого для подсчета количества свойств объекта можно использовать методы, такие как `Object.keys(obj).length`, чтобы получить количество ключей (элементов) в объекте.

№141

В этих задачах тип ошибки - это ошибки логики или типов данных.

Логический тип 1. Ошибки логики: Такие ошибки происходят, когда условия в блоках `if` и `else` не соответствуют тому, что требуется от кода. Например, в задачах №1 и №2 неправильно оцениваются условия для проверки суммы чисел.

Синтаксический тип 2. Ошибки типов данных: Происходят, когда значения переменных не соответствуют ожидаемому типу. Например, в задачах №2 и №5 используются строки вместо чисел при проверке условий.

Синтаксический тип 3. Ошибки синтаксиса: В задаче №9 присутствует ошибка, где переменная `str` не определена перед использованием.

Синтаксический тип 4. Неправильное использование методов: В задачах №10 и №12 присутствуют опечатки при обращении к свойству `length`.

№166

В этих задачах присутствуют различные типы ошибок:

Логический тип 1. Логические ошибки: Эти ошибки возникают, когда логика кода неверна. Например, в задаче №1 условие `i > 10` никогда не будет истинным, поэтому цикл не выполнится ни разу.

Синтаксический тип 2. Ошибки счетчика цикла: В задачах №2, №3, №4, №5, №6, №10, №11, №12, №14 есть проблемы с инициализацией, условием и/или инкрементом счетчика цикла, что приводит к неправильной работе цикла.

Синтаксический тип 3. Ошибки обработки данных: Например, в задачах №7, №8, №9, №15, №16 происходят ошибки при вычислении суммы элементов массива или объекта из-за неправильной обработки данных.

Синтаксический тип 4. Ошибки синтаксиса: В задачах №20 и №21 присутствуют ошибки в синтаксисе операторов: вместо оператора сравнения `===` использован оператор присваивания `=`.

№199  
В этом наборе задач присутствуют различные типы ошибок:

Синтаксический тип 1. Ошибки преобразования типов данных: Например, в задачах №1, №2, №3, №4, №5 происходят ошибки при работе с числами и строками. Метод `split()` разбивает строку на массив подстрок, но элементы этого массива остаются строками, а не числами. При выполнении операций суммирования или умножения строк JavaScript склеивает их, а не выполняет арифметические операции.

Логический тип 2. Логические ошибки: В задаче №3 в цикле значение переменной `sum` перезаписывается на каждой итерации, вместо того чтобы прибавлять к нему значение очередной цифры.