JavaScript - переменные

переменные, типы данных, преобразование типов данных

Переменные

Переменная – это «именованное хранилище» для данных.

В JavaScript есть два ограничения, касающиеся имен переменных:

- Имя переменной должно содержать только буквы, цифры или символы $ и _.
- Первый символ не должен быть цифрой.

Регистр имеет значение

Переменные с именами apple и APPLE – это две разные переменные.

Нелатинские буквы разрешены, но не рекомендуются

Зарезервированные имена

Существует список зарезервированных слов, которые нельзя использовать в качестве имён переменных, потому что они используются самим языком.

Например: let, class, return и function зарезервированы.

Var

var – это устаревший способ объявления.

Переменные, объявленные с помощью var, имеют функциональную область видимости (function scope), это означает, что они доступны во всей функции, в которой были объявлены.

При использовании var, переменные могут быть объявлены повторно без предупреждения, что может привести к нежелательным ошибкам в коде.

Переменные var поднимаются (hoisted) в начало своей области видимости, что может вызывать путаницу, когда разработчик не осознает это поведение.

Let

let – это современный способ объявления.

let появилось в ECMAScript 6 (ES6) и предоставило блочную область видимости (block scope) для переменных.

Переменные, объявленные с помощью let, доступны только в блоке, в котором были объявлены.

Переменные let не могут быть повторно объявлены в той же области видимости, что помогает избежать ошибок.

Поднимаются только до начала блока, в котором объявлены.

Const

const – похоже на let, но значение переменной не может изменяться.

const также появилось в ECMAScript 6 (ES6) и используется для объявления переменных с неизменяемым значением.

Значение переменной, объявленной с помощью const, не может быть переназначено после инициализации.

Как и переменные, объявленные с помощью let, const имеют блочную область видимости.

Как и let, переменные const не могут быть повторно объявлены в той же области видимости.

Типы данных

Числа (Number): Представляют как целые числа, так и числа с плавающей запятой. Например: 5, 3.14, -10.

Строки (String): Последовательность символов, заключенная в одинарные ' или двойные " кавычки. Например: "Hello", 'JavaScript'.

Логические значения (Boolean): Могут быть true или false, используются для логических выражений и условных операций.

Undefined: Представляет значение, которое не было присвоено переменной или функции. По умолчанию переменные в JavaScript имеют значение undefined.

Null: Представляет "ничего", "пустоту" или "отсутствие значения". Обычно используется для явного указания на отсутствие значения.

Symbol (с версии ECMAScript 6): Уникальное и неизменяемое значение, используемое для создания уникальных идентификаторов объектов.

BigInt (с версии ECMAScript 2020): Представляет целые числа произвольной длины. Например: 1234567890123456789012345678901234567890n.

Объекты (Objects):

Объекты представляют собой коллекции пар ключ-значение. Ключи являются строками (или Symbol), а значения могут быть любого типа данных, включая другие объекты.

Объекты могут быть созданы с помощью фигурных скобок {} или с помощью конструктора new Object().

Функции (Functions):

Функции в JavaScript являются объектами первого класса, что означает, что они могут быть переданы как аргументы, возвращены из других функций и сохранены в переменных.

Функции могут быть объявлены с помощью ключевого слова function, или как стрелочные функции (с версии ECMAScript 6).

Массивы (Arrays):

Массивы в JavaScript являются специальными типами объектов, предназначенными для хранения упорядоченных коллекций элементов.

Массивы могут содержать элементы любого типа данных, и доступ к элементам осуществляется по индексу.

Специальные значения:

NaN (Not-a-Number): Особое числовое значение, которое представляет некорректное математическое действие.

Infinity и -Infinity: Положительная и отрицательная бесконечности, используемые для представления переполнения числа или деления на ноль.

Преобразование типов данных

В JavaScript преобразование типов — это процесс автоматического приведения значений одного типа данных к другому типу во время выполнения операций или вычислений. Преобразование типов может происходить явно (явное преобразование) или неявно (неявное преобразование)

Явное преобразование типов:

Явное преобразование выполняется с помощью встроенных функций-конструкторов или методов, специально предназначенных для этой цели. Например:

```
let numString = "123";
let num = Number(numString); // Явное преобразование строки в число
```

Неявное преобразование типов:

Неявное преобразование типов происходит автоматически во время выполнения операций или вычислений, когда оператор или функция ожидает значение определенного типа.

Примером неявного преобразования может служить использование операторов или выражений с различными типами данных:

```
let x = 5 + "10"; // Неявное преобразование числа в строку
console.log(x); // Выведет "510"
```

Преобразование в строку:

Преобразование в строку происходит с помощью функции String() или при контексте, ожидающем строку. Например:

```
let num = 123;
let str = String(num); // Преобразование числа в строку
```

Преобразование в число:

Преобразование в число выполняется с помощью функции Number() или при контексте, ожидающем числовое значение. Например:

```
let str = "123";
let num = Number(str); // Преобразование строки в число
```

Преобразование в логическое значение:

Преобразование в логическое значение выполняется с помощью функции Boolean() или при контексте, ожидающем логическое значение. Например:

```
let x = 0;
let bool = Boolean(x); // Преобразование числа в логическое значение
```

Ресурсы:

- Переменные - ТЫК
- Типы данных - ТЫК
- Преобразование типов - ТЫК
- Зарезервированные имена переменных - ТЫК