

Введение в Javascript. Операторы

Frontend. Урок 18

Вопросы на повторение

1. Что такое Переменные в JS?
2. Какие типы данных есть JS?

Постановка проблемы

На данный момент мы можем запоминать и выводить какие-либо значения на экран. Но для более сложных программ нам понадобится что-то делать с этими значениями - например: сложить, умножить, сравнить, убрать пробелы, заменить символы и т.д.

Как мы можем решить эту проблему?

Операторы

Начнем с понятия операторов в JS. С операторами вы уже знакомы со школы - это "+", "-", "*", "/" и другие математические действия. В JS к ним добавляются специальные удобные для разработки операторы, о которых мы с вами и поговорим.

Операнд - то, к чему применяется оператор, это может быть что угодно - число, строка, переменная и т.д.

Унарный оператор - оператор, который применяется к одному операнду. Например, оператор унарный минус "-" меняет знак числа на противоположный:

```
let a = 32  
a = -a  
console.log(a) //-32
```

Бинарный оператор

Бинарный оператор - оператор, который применяется к двум операндам.

Тот же минус существует и в бинарной форме:

```
let a = 3, b = 4  
a = a - b  
console.log(a) //-1
```

Математические операторы

Плюс, минус, умножить, разделить и скобки работают также, как и в математике.

Дополнительные математические операторы:

- “%” - остаток от деления :

```
let a = 5  
console.log(a % 3)
```

- “**” - возведение в степень :

```
let a = 5  
console.log(a ** 3) //125
```

Строки

Строки тоже можно складывать - в этом деле важна последовательность, в отличие от чисел

```
let a = "сло"
let b = "во"
console.log(a + b) //слово
console.log(b + a) //восло
```

Преобразования значений

Есть два основных типа преобразования значений:

Неявные преобразования

Это когда разработчик ненамеренно преобразовывает один тип к другому.

Пример:

```
let a = 1
let b = "2"
console.log(a + b) // "12"
console.log(b + a) // "21"
```


Преобразования значений

В данном примере число преобразуется в строку - это заслуга бинарного сложения, который при появлении строки все преобразует в строку. А остальные математические операторы всё преобразовывают в числа:

```
let a = 5
let b = "2"
console.log(a - b) //3
console.log(b - a) //-3
```

Преобразования значений

Явные преобразования

Тут уже разработчик намеренно меняет тип значения, например используя унарный плюс:

```
let a = 5
let b = "2"
console.log(a + +b) //7
console.log(a + b) //52
```

Выражение `+b` тут преобразовывает строку в число.

Преобразование типов

Также, мы можем еще понятнее преобразовывать типы вызывая специальные методы:

```
let a = 5
let b = "2"
console.log(a + Number(b)) //7
console.log(a + 4) //9
console.log(String(a) + 4) //54
```

Тут метод `Number(..)` преобразовывает значение, которое ему передают в скобках (так называемый параметр) в число, а `String(..)` то же самое, но в строку.

Сокращенная арифметика, инкремент и декремент.

Мы можем использовать наши операторы двумя способами, представим, что нам нужно увеличить переменную “a” на 5, мы можем сделать это двумя способами:

Первый способ:

```
let a = 5  
a = a + 5 // a=10
```

Второй способ:

```
let a = 5  
a += 5 // a=10
```

Сокращенная арифметика, инкремент и декремент.

Аналогично существуют и такие операторы, как:

`-=`, `/=`, `*=`, `%=`, `**=`

Которые подчиняются такой же логике.

И последние два оператора, которые мы разберем на этом занятии
- инкремент(увеличитель) и декремент(уменьшитель).

Это унарные операторы для частного случая, когда нам нужно изменить значение на единицу в большую или меньшую стороны:

```
let a = 5  
a++ // a=6  
a-- // a=5
```

Практика А

Создание простого калькулятора, который выводит все действия, которые мы сегодня прошли.

Выполнить это нужно в песочнице (<https://runjs.co>), после выполнения задания создать в папке урока файл `taskA.js`, в него вставить свой код и запушить в свой репозиторий для проверки.

Практика В

Пройти уровни в игре, пользуясь руководством (pdf-файл)

<https://lab.reaal.me/jsrobot/>