

Основы программирования

Что такое программа?

Компьютерная программа (она же приложение) — связка многочисленных строк специального текста. Он является специальным, потому что создан таким образом, чтобы машине было понятно, какие действия должны быть выполнены.

Что такое код?

Компьютерный код — это специальный текст, состоящий из набора пошаговых инструкций. Он не всегда содержит в себе нули и единицы, также в нём есть определённые слова и дополнительные символы. Компьютер считывает код, который сообщает ему, какие операции следует выполнить с данными.

Начнем

Давайте перейдём к практической части и попробуем написать какой-нибудь код. Основным элементом в компьютерной программе является переменная, которая в свою очередь определяет имя для какой-нибудь информации, будь то число или текст. Всегда нужно давать названия данным.

Чтобы переменная существовала, её нужно объявить. Объявление переменных служит в роли свидетельства о «рождении» для данных.

Пример:

```
let the_number_of_days_in_december
```

Типы данных JS

Языки программирования, в которых такое возможно, называются «динамически типизированными». Это значит, что типы данных есть, но переменные не привязаны ни к одному из них.

Число number

- `number` для любых чисел: целочисленных или чисел с плавающей точкой.

Строка string

- `string` для строк. Строка может содержать один или больше символов, нет отдельного символьного типа.

Логический тип boolean

- `boolean` для `true/false`.

NULL

- `null` для неизвестных значений – отдельный тип, имеющий одно значение `null`.

undefined

- `undefined` для неприсвоенных значений – отдельный тип, имеющий одно значение `undefined`.

Объект object

- `object` для более сложных структур данных.

Функции

Функции - ключевая концепция в JavaScript. Любая **функция это объект**, и следовательно ею можно манипулировать как объектом, в частности:

- передавать как аргумент и возвращать в качестве результата при вызове других функций (**функций высшего порядка**);
- создавать анонимно и присваивать в качестве значений переменных или свойств объектов.

Функция в JavaScript специальный тип объектов, позволяющий формализовать средствами языка определённую логику поведения и обработки данных.

Объявление функции

Объявление функции состоит из ключевого слова `function` и следующих частей:

- Имя функции.
- Список параметров (принимаемых функцией) заключенных в круглые скобки `()` и разделенных запятыми.
- Инструкции, которые будут выполнены после вызова функции, заключают в фигурные скобки `{ }`.

```
function square(number) { return number * number; }
```

Вызов функции

Объявление функции не выполняет её. Объявление функции просто называет функцию и указывает, что делать при вызове функции. **Вызов** функции фактически выполняет указанные действия с указанными параметрами.

```
square(5);
```

Циклы

При написании скриптов зачастую встаёт задача сделать однотипное действие много раз.

Например, вывести товары из списка один за другим. Или просто перебрать все числа от 1 до 10 и для каждого выполнить одинаковый код.

Для многократного повторения одного участка кода предусмотрены *циклы*.

Цикл «while»

```
while (condition) {  
    // код  
    // также называемый "телом цикла"  
}
```

Код из тела цикла выполняется, пока условие `condition` истинно.

Цикл «do...while»

```
do {  
    // тело цикла  
} while (condition);
```

Цикл сначала выполнит тело, а затем проверит условие `condition`, и пока его значение равно `true`, он будет выполняться снова и снова.

Цикл «for»

```
for (начало; условие; шаг) {  
    // ... тело цикла ...  
}
```

Начало выполняется один раз, а затем каждая итерация заключается в проверке *условия*, после которой выполняется *тело* и *шаг*.

Итеративный процесс

Итеративный процесс постоянно говорит *«я сейчас посчитаю все что можно и продолжу»* на каждом шаге рекурсии. Ему не нужно ничего запоминать вне вызова, он всегда считает все в первый возможный момент, и каждый шаг рекурсии может существовать в изоляции от прошлых, потому что вся информация передается из шага в шаг.

Рекурсивный процесс

Рекурсивный процесс постоянно говорит «я это запомню и потом посчитаю» на каждом шаге рекурсии. «Потом» наступает в самом конце.

