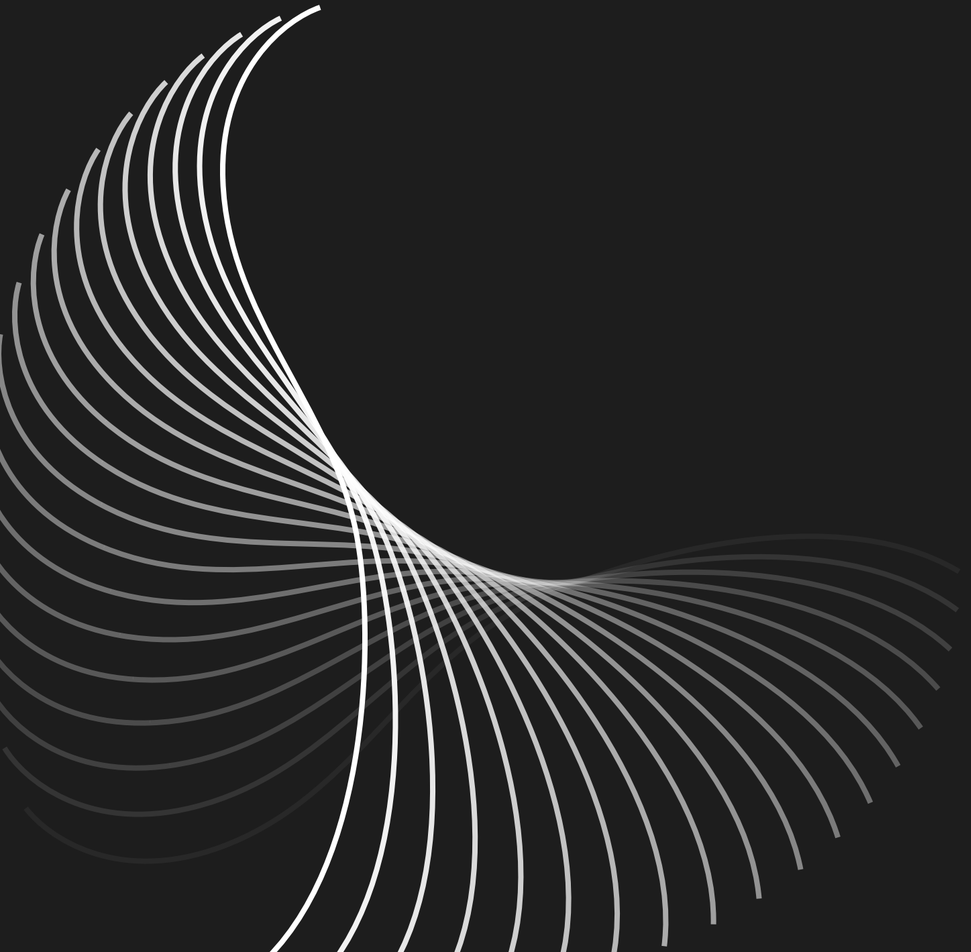
JavaScript

Урок 3. Циклы



язык программирования для веб-разработки №1

Введение

- Часто нужно сделать одно и то же действие много раз. И далеко не всегда мы знаем, СКОЛЬКО раз.
- Если нам нужно сделать одно действие n-ное количество раз, нам понадобятся циклы.
- В JS есть два вида циклов. Мы рассмотрим оба, а вы выберете, какой лучше. Но есть ситуации, когда нам поможет только один из них.

Цикл for

```
for (let i = 0; i < 10; i++)
```

Ой... как сложно... Действительно выглядит запутанно. for с английского - для, потому что мы создаем специальную переменную, которая занимается подсчетом, сколько раз выполняется цикл. Цикл for подойдет, когда мы знаем количество итераций или можем это число вычислить.
Итак, что в скобках?

Цикл for

```
for (let i = 0; i < 10; i++)
```

Сначала мы создаем переменную и задаем ей начальное значение. В языках программирования часто отсчет идет с нуля, а не с единицы, вы привыкнете. И на это есть причина!

Потом мы указываем условие продолжения цикла. Если перед началом итерации ответ на вопрос i < 10 - ДА, значит цикл продолжается. Когда ответом станет НЕТ, цикл закончится.

Потом пишется шаг переменной. Не всегда этот шаг равен +1. Кстати, запись i++ - сокращение для i = i + 1

Цикл for

Итак, как это работает? Давайте выведем в консоль цифры от 0 до 99. Вручную это будет огого как долго. Воспользуемся циклом.
Отсчет начинаем с 0 - ведь нам нужен ноль в начале цикла.
Условие выхода из цикла удобнее писать после того, как пропишем шаг.
Шаг. С каждой итерацией число i будет увеличиваться на 1
Вернемся к условию. i будет расти-расти-расти на единицу с каждой итерацией. В какой-то момент оно станет больше какого-то числа... Какого?
100!

```
for (let i = 0; i < 100; i++) {
    console.log(i);
}
```

Примеры

1. 100 раз вывести “Hello world”

```
for (let i = 0; i < 100; i++) {
    console.log("Hello world");
}
```

2. Посчитать от 10 до 1

```
for (let i = 10; i > 0; i--) {
    console.log(i);
}
```

3. Посчитать от 1 до числа, который ввел пользователь

```
let max = prompt("Введите число");
for (let i = 1; i <= max; i++) {
    console.log(i);
}
```

Цикл while

```
while (i == 100)
```

Иногда мы не знаем, сколько раз сработает цикл. Допустим, мы играем с пользователем в угадайку - предлагаем угадать загаданное число (или животное. или цвет. или что угодно). Мы даем бесконечное число попыток. Откуда нам знать, за сколько попыток пользователь угадает? Это непредсказуемая информация, причем меняющаяся при каждом запуске программы.

Цикл while

while (i == 100)

На самом деле механизм похож на for. Но в for встроен шаг изменения переменной. Он создан для ситуаций, когда есть последовательность чисел - итератор внутри for изменяется по каким-то определенным правилам. while не включает в себя механизм изменения переменной и начальное значение, мы указываем лишь условие захода на новую итерацию. while - с английского пока. Поэтому можно перевести как “Пока число не будет равно 100”. Когда это число станет равно 100, никто знать не может!

Примеры

1. Пользователь угадывает число

```
let num = 45;
let usernum = prompt("Угадайте число от 1 до 100");

while (usernum != num) {
    usernum = prompt("Попробуйте еще раз. Угадайте число от 1 до 100");
}
alert("Вы угадали, это число " + num + "! ");
```

2. Посчитать от 1 до 100

```
let i = 1;

while (i <= 100) {
    console.log(i);
    i++;
}
```

3. Посчитать от 1 до числа, который ввел пользователь

```
let max = prompt("Введите число");

let i = 1;
while (i <= max) {
    console.log(i);
    i++;
}
```