



## Урок № 1

### Курс: «Динамический сайт: Java Script»

#### Тема: Знакомство с JavaScript. Основные конструкции языка

#### Цель урока

Ознакомиться с языком Javascript, его возможностями. Научиться вставлять javascript в html-страницу. Ознакомиться с типами данных и принципами создания переменных. Освоить вывод сообщений на экран, используя alert. Изучить операторы ветвления и цикла.

#### Инструментарий

Компьютер с установленным программным обеспечением для web-разработки (MS Visual Studio/ Dreamweaver/...), browser.

#### Назначение и возможности

Если мы создаем свой сайт или сайт компании (например, на заказ), то главными требованиями к нему будут привлекательность и нормальная работа в наиболее распространенных браузерах. Если с задачей привлекательности справляются HTML и CSS, то со второй возникают проблемы. Для ее решения используют язык программирования javascript (js).

Js используется в веб-программировании для:

- обеспечения нормальной работы в разных браузерах;
- получения информации о текущей дате, времени...;
- обработки событий на странице, которые "создал" пользователь (движения мышки, клики, нажатие клавиш клавиатуры...);
- изменение внешнего вида элементов страниц во время работы с сайтом;
- ввод и проверка правильности информации, которую дал пользователь;
- создание сложных анимационных эффектов;
- работа с дополнительными окнами (открытие, изменение размеров, видимости...).

Js не работает с базами данных и файлами на сервере, с файлами пользователя (клиента), не имеет доступ к оперативной памяти клиента

## Вставка js

Вставить js в html страницу можно двумя способами:

1. подключить как внешний файл;
2. вставить внутри html страницы.

Для работы с js необходимо:

- создать html страницу
- создать js блок.

Как это сделать рассмотрим далее.

## Типы данных и переменные

Так как скрипт предназначен для обработки информации, то ее следует где-то хранить. Для этого используют переменные.

Чтобы создать переменную надо написать

```
var name;
```

где var - обязательное слово, которое говорит о том, что перед нами переменная;

name - имя переменной. Имя переменной может быть каким угодно, т.е. любое слово. Если слов не достаточно, то к ним можно дописывать цифры. Например,

```
var name1;  
var x23;  
var Name1;
```

Но стоит помнить о том, что:

- имя не может начинаться с цифры,
- одинаковые имена, записанные с большой и малой букв, - это имена разных переменных.

В js четыре типа данных:

- числа;
- строки;
- логический тип (принимает значения "true/false" - истина/ложь);
- объекты (о них поговорим чуть позже).

Тип определяется, когда мы говорим, чему равна переменная. Например,

```
name1 = "Hello"; //строковый тип  
x23 = 5; //числовой тип  
Name1 = true; //логический тип
```

```
//это комментарий. То, что видите вы, но "не видит" скрипт
```

## Вывод на экран

Самый простой способ вывода информации на экран - это запись такого вида

```
alert("Тут какой-то текст");
```

Чтоб вывести значения переменной записывают так

```
var z;  
alert(z);
```

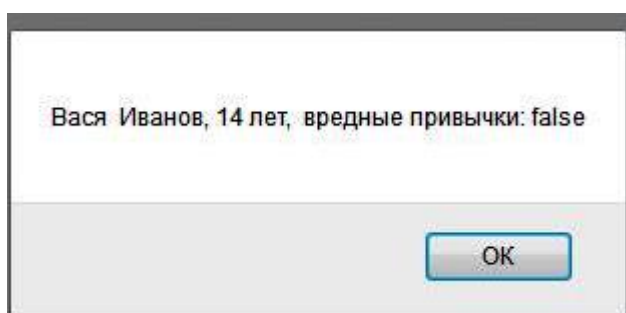
Для закрепления материала решим такую задачу.

Создать переменные всех простых типов, инициализировать в скрипте, вывести их значения. Например, вывести Ф.И.О. и возраст студента.

Пишем код

```
<body>  
<script type="text/javascript">  
//переменная строкового типа для хранения имени. Сразу задаем значение  
var name = "Вася";  
//переменная для хранения фамилии  
var surname;  
//задаем значение переменной типа "строка"  
surname = "Иванов";  
//переменная числового типа. Сразу задаем значения  
var age = 14;  
//переменная логического типа. Значение: 1 (true)- есть вредные привычки, 0 (false) -  
нет вредных привычек  
var gender = false;  
//выводим все на экран  
alert(name + " " + surname + ", " + age + " лет, вредные привычки: " + gender);  
</script>
```

Открываем нашу страницу в браузере. Результат работы скрипта - на рисунке ниже



## Ветвление

Структуру ветвления используют, когда надо принять решение. Записывается это так

```
if (условие)
{
//какие-то действия, если условие выполнилось
}
else
{
//какие-то действия, если условие не выполнилось
}
```

Условие – это то, на основании чего мы принимаем решение. Например, если температура воздуха на улице меньше нуля, надо надеть шапку, иначе – иду без шапки. Ветвлением это записывают так:

```
if (температура ниже нуля?)
{
//надеваю шапку
}
else
{
//иду без шапки
}
```

Т.е. else – это блок «иначе», его может не быть.

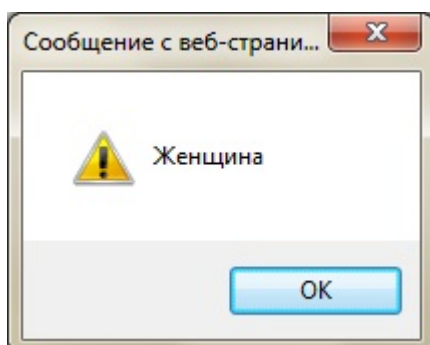
Рассмотрим задачу.

Ввести переменную логического типа, в зависимости от ее значения выводить пол (м/ж).

Пишем код

```
<script type="text/javascript">
//переменная для определения пола
var gender;
//задаем пол
gender = false;
// в зависимости от значения переменной, определяем пол
if (gender == true) {
    alert("Мужчина");
}
else {
    alert("Женщина");
}
</script>
```

Открываем нашу страницу в браузере. Результат работы скрипта - на рисунке ниже



Блоки ветвления могут вкладываться друг в друга. Для того, чтоб понять, как это, рассмотрим задачу.

Ввести переменную для задания возраста пользователя. В зависимости от возраста, выводить сообщения с текстом «ребенок/юноша (девушка)/взрослый».

Пишем код

```
//переменная для задания возраста
var age;
//задаем возраст
age = 18;
// в зависимости от значения переменной, определяем возрастную категорию
if (age <14) { //если меньше 14 - ребенок
    alert("Это ребенок");
}
else {//иначе
    if (age >= 14 && age < 18) //от 14 до 18 - подросток
    { alert("Это подросток"); }
    else //иначе - взрослый, т.е. больше 14, не попадает в интервал 14-18
    {alert("Это взрослый");}
}
```

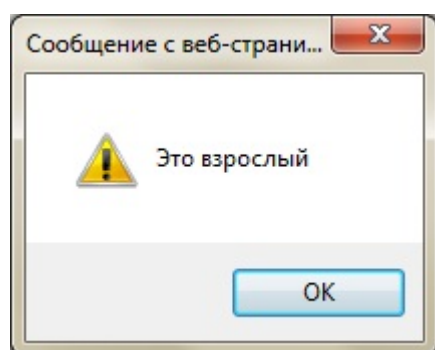
В составном условии `age >= 14 && age < 18`

знак `>=` больше или равно

знак `&&` «и», то есть больше или равно 14 и меньше 18.

Чтоб поставить «или» пишут `||` (2 вертикальные палочки между простыми условиями)

Открываем нашу страницу в браузере. Результат работы скрипта - на рисунке ниже



## Циклы

Конструкцию цикла стоит использовать, когда одно и то же действие повторяется несколько раз. Например, собираем салат. Берем помидор, моем, режем, кладем в миску. Берем огурец, моем, режем, кладем в миску. Берем перец, моем, режем, кладем в миску. То есть над каждым овощем выполняется одно и то же действие. Эту операцию мы будем повторять столько раз, сколько у нас разных ингредиентов.

Когда известно действие и количество раз, которое его надо повторить, используют **цикл со счетчиком**. Его записывают так:

```
for( var i=0;i<N;i++)
{
//действие;
}
```

Здесь `i` – это счетчик. Его значение будет изменяться от нуля (`var i=0`) до `N` (`i<N`). Шаг изменения – единица (`i++`).

Чтоб задать шаг изменения, отличный от 1, пишем

`i+=step`

step – переменная или число, на которое будет увеличиваться счетчик

`i-=step`

step – переменная или число, на которое будет уменьшаться счетчик

Переменную-счетчик (i) мы можем использовать только внутри цикла, то есть внутри фигурных скобок.

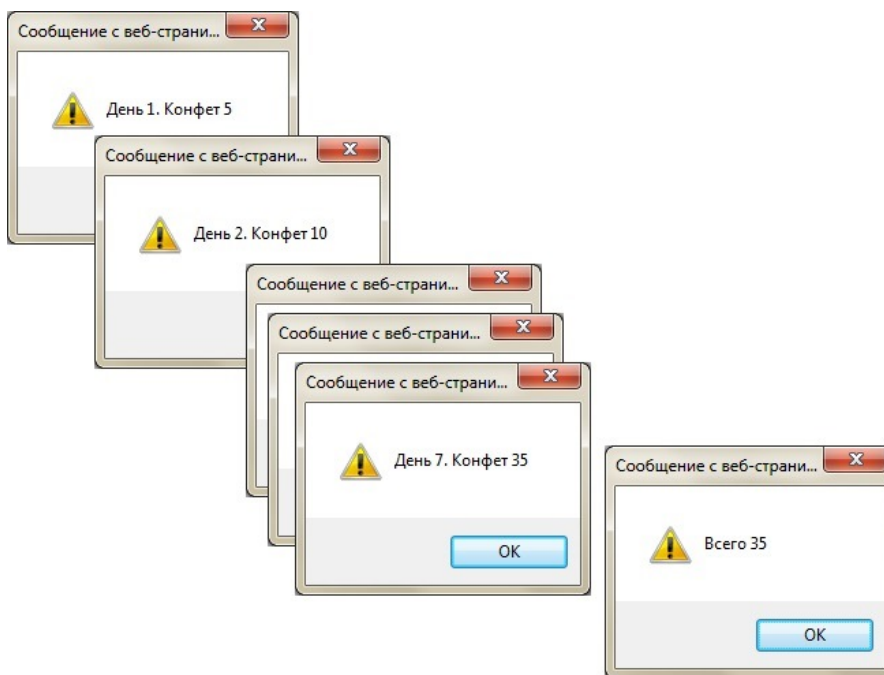
Рассмотрим задачу.

Задать переменную, отвечающую за количество конфет. Каждый день количество увеличивается на N. Вывести количество конфет через неделю (месяц...).

Пишем код

```
var candy = 0; //количество конфет. Изначально конфет нет
var N = 5; //сколько конфет нам буду давать каждый день
for (var i = 1; i <=7; i++) { //с 1 по 7 день получать конфеты
    candy = candy + N;
    alert("День " + i + ". Конфет " + candy);
}
alert("Всего " + candy);
```

Открываем нашу страницу в браузере. Результат работы скрипта - на рисунке ниже



Иногда мы не знаем сколько раз надо выполнить одно и то же действие. Возвращаясь к примеру с салатом, это значит, что мы будем выполнять действия пока овощи не кончатся.

Проверили, есть ли овощ. Если да, то «обработали» его, опять посмотрели, есть ли еще.

В подобных случаях можно использовать **цикл с предусловием**. Он записывается так:

`while(условие )`

{

//действия

}

Рассмотрим задачу.

Задать переменную, отвечающую за количество конфет. Оговорить условие прибавления конфет. Найти за сколько дней конфет станет N штук.

Пусть нам надо собрать N=200 конфет. Каждый день нам будут давать по 30 штук.

Пишем код

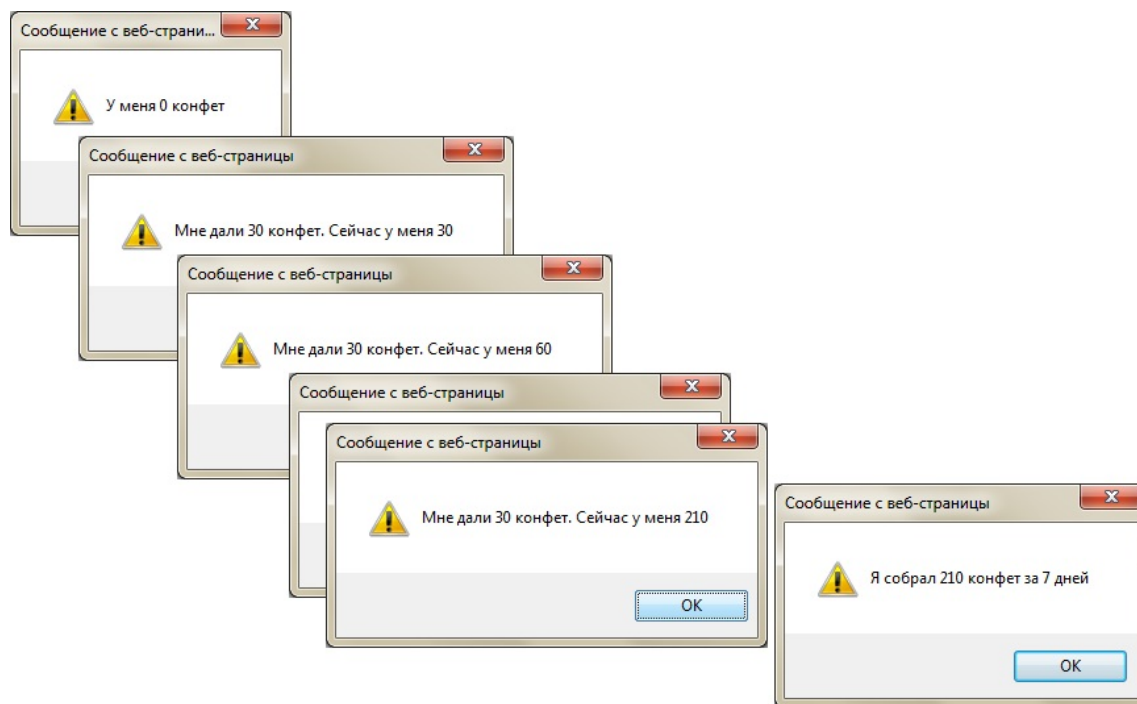
```
var N = 200; //количество конфет, которое надо собрать
var days = 0; //количество дней, за которое собираем конфеты
var cd = 30; //количество "прибывающих" конфет в день
var my_candy = 0; //количество конфет, которое уже собранно

alert("У меня " + my_candy + " конфет");
while (my_candy < N) {
    my_candy = my_candy + cd;
    alert("Мне дали " + cd + " конфет. Сейчас у меня " + my_candy);
    days++;
}

alert("Я собрал " + my_candy + " конфет за " + days + " дней");
```

Здесь каждый день будем выводить сообщение о «положении дел», а если у нас есть желаемое количество конфет (N=200) или больше, то скажем об этом.

Открываем нашу страницу в браузере. Результат работы скрипта - на рисунке ниже



Существует также цикл с постусловием. Записывается он так

```
do{
//действия
} while(условие );
```

Отличается от предыдущего тем, что в таком случае действия выполняются хотя бы 1 раз, а лишь потом выполнится проверка условия.

Если говорить о салате, то мы берем помидор, моем, режем, кладем в миску. Спрашиваем, есть ли еще овощи. Если нет, завершаем работу, иначе – опять выполняем действия.