

Введение в JS

План занятия

- ◆ Настройка окружения
- ◆ Структура кода
- ◆ Типы данных
- ◆ Переменные

Что такое JavaScript

JavaScript (JS) — **высокоуровневый** (понятный скорее людям, а не машинам) язык программирования. Он относится к языкам с **динамической типизацией** (т.е. тип к переменной применяется как только указывается, чему она равна) и входит в группу **интерпретируемых** (т.е. код читает не машина, а другая программа (интерпретатор)) языков.

Если HTML и CSS — языки для разметки страницы, то JS помогает «оживлять» эти страницы, добавляет *функциональность и динамичность*.

Как подключить JS

Есть два способа подключения JS к файлу HTML:

1. Код пишется прямо в теге `<script>`

Его можно расположить где угодно в программе. Обычно код располагают в самом конце, т.к. программа считывается сверху вниз. Поэтому, сначала она успеет обратиться к структуре страницы, а потом подключить к ней функциональность.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>заголовок</title>
  </head>
  <body>
    Содержимое страницы.
    <script>
      JavaScript
    </script>
  </body>
</html>
```

Как подключить JS

2. Подключение внешнего файла (во многих случаях лучше так)

Загружается с помощью тега `<script>` и атрибута `src` с путём к файлу. Файл JS сохранятся с расширением `.js`

```
<body>  
    Содержимое страницы.  
    <script src="js/inno.js"></script>  
</body>
```

Как выглядит js-файл

Файл начинается со строки **"use strict"**. Над "use strict" могут быть записаны *только комментарии*.

Эта строка включает режим браузера для использования всех современных функций языка (обычно называют строгим режимом). Каждая функция или инструкция пишется с новой строки для читабельности кода.

В большинстве случаев точку с запятой после каждой инструкции можно не ставить, если есть переход на новую строку (но это не значит что так будет работать всегда!).

Комментарии:

```
// однострочный
/*
  многострочный
  комментарий
*/
```

Пример

Функция нужна для выполнения действий. Функция `alert` - выводит текст на экран в диалоговое окно.

Имя функции `alert`, после чего для вызова её мы пишем круглые скобки, в которые передаём параметры (у нас это строка).

```
"use strict";  
alert('INNODOM!');
```

Подтвердите действие
INNODOM!

OK

Типы данных

Каждый ЯП имеет свои **встроенные типы данных**. JS - это динамический и слаботипизированный язык программирования. Это значит, что тип определяется автоматически, по мере выполнения программы.

JS включает 8 типов (6 из которых - примитивные и 2 - нет)

Типы данных

- Number (Число)
- String (Строка)
- Boolean (Логический тип) - правда или неправда
- Undefined (Неопределённый тип) - не имеет значение
- BigInt - большие числовые значения (длинные арифметические значения)
- Symbol (в ECMAScript 6)
- Null (Null тип) - примитив, его тип - это объект, означает несуществующий объект («я ещё не знаю, что здесь будет»)
- Object (Объект)

Переменные

Переменная – это область памяти под своим именем для хранения данных. Для создания переменных используется ключевое слово **let**. Значение в переменную присваивается с помощью оператора присвоения =

```
let innodom = 'Учитесь с нами!'
```

Переменная с именем *innodom* и присвоенным значением - это строка, а значит переменная имеет тип **string**.

Область памяти можно представить в виде бокса, который хранит записанные данные (бокс с названием). Если создать переменную без значения, то она будет неопределенной, но ошибки не возникнет.

! В старых кодах можно наткнуться на объявление переменных через ключевое слово *var*, сейчас это устаревший подход.

Переменные

Объявить несколько переменных можно через запятую для сокращения кода. Две записи будут эквивалентны:

```
let i=1  
let n=2  
let o=3
```

```
let i=1,n=2,o=3
```

Объявлять переменные с одинаковым именем нельзя, возникнет ошибка (разные области памяти не могут иметь одно имя).

Переменные

Переменную можно перезаписывать просто вызывая её по имени и присваивать новое значение с помощью оператора присвоения =

```
let inno=1  
inno=2  
alert(inno)
```

Alert выведет 2

Можно создать переменную и записать её значение в другую:

```
let inno = 1  
let dom = inno  
alert(dom)
```

Alert выведет 1

Как называть переменные

- ◆ Имя переменной не должно начинаться с цифры
- ◆ Имя может содержать буквы, цифры, а также символы \$ и _
- ◆ Регистр имеет значение (innodom, INNODOM - будут разными переменными)
- ◆ Называть не латинскими буквами переменные разрешено, но так делать не советуем
- ◆ Нельзя называть зарезервированными словами (например, переменную let назвать let, class, id и т.д.)

Название в **CamelCase** - это когда слова пишутся слитно и каждое новое слово начинается с заглавной буквы.

Snakecase - написание слов через нижнее подчеркивание.

Константы

Неизменная переменная - имеет постоянное значение, которое нельзя изменить (перезаписать). Объявляется с помощью слова **const**

```
const innodom = 'hi';  
innodom = 'hello';  
// будет ошибка, константа нельзя перезаписывать
```

Иногда слишком длинные и не запоминающиеся данные, их нужно записать как постоянную, для более легкого к ним обращения. Такие постоянные принято называть в **верхнем регистре через нижнее подчеркивание**. Например, запомнить название постоянной легче, чем каждый раз искать это длинное значение.

```
const KEY = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9';
```


Операторы

- ◆ Операторы присваивания
- ◆ Операторы сравнения
- ◆ Арифметические операторы
- ◆ Битовые (побитовые) операторы
- ◆ Логические операторы
- ◆ Строковые операторы
- ◆ Условный (тернарный) оператор
- ◆ Оператор запятая
- ◆ Унарные операторы

Операторы

◆ Оператор присваивания

Основным является оператор `=` (перезванивает значение операндам)
Также существуют сокращённые операторы арифметических действий (это рассмотрим в уроке с арифметическими действиями JS).

◆ Арифметические операторы

Операторы сложения `(+)` вычитания `(-)` и остальные арифметические действия, о них будет рассказываться тоже дальше.

Операторы

- ◆ **Логические операторы**

Обычно возвращают ответ типом true или false, не учитывая операторов || (или) && (и) они будут возвращать одно из значений.

- ◆ **Строковые операторы**

Оператор + (объединяет две строки).

- ◆ **Операторы сравнения**

Больше, меньше, равно и т.д.