Школа инженера
Основы веб-разработки: JavaScript

Постников Илья

www.digdes.ru

JavaScript История создания и стандарты

- Netscape 1995 LiveScript/JavaScript
- 1997 Ecma International первая стандартизация ECMAScript – ES1
- ES5 (2009)
- ES6 (2015) стандарт для современных браузеров

Что может JavaScript?

- фронт-разработка браузерных web-приложений
- приложения для мобильных устройств
- бекенд-приложения под NodeJS
- приложения под Windows 8 / 8.1 / 10
- smart-контроллеры (Internet Of Things)

Подключение и выполнение (01_run_js.html)

- На странице в тегах <script>…</script>
- Подключение внешнего файла <script src="*имя_файла.js*">
- Последовательное исполнение тегов html и js-скриптов

Отладочная информация

- Вывод в модальное окно alert
- Вывод в консоль браузера console.log
- Печать на страницу document.write

Комментарии

- Однострочные //…
- Многострочные /*…*/

Переменные и типы данных (02_typeandvar.html)

Объявление глобальных и локальных переменных

- var
- let, const – создают новую область видимости переменной

Типы

- string – строки как “…” или ’…’
- number – целые или с плавающей точкой -2^{53} до 2^{53}
- boolean – true или false
- undefined – неопределенное значение (объявили но не присвоили)
- null – пустое значение
- object – { имя_свойства: значение_свойства }

Массивы - new Array или [знач1, знач2, … ,значN] (тип – object, проверка Array.isArray)

Слабая типизация – переменная может менять тип динамически

typeof – проверка типа

Замыкания и функции IIFE (03_ClosureAndIIFE.html)

Замыкание - функция, созданная в одной области видимости, запоминает свое лексическое окружение даже в том случае, когда она выполняется вне своей области видимости.

Компоненты замыкания:

- внешняя функция – определяет свой скоуп из набора переменных
- переменные (лексическое окружение) - определеные во внешней функции
- вложенная функция - использует эти переменные

Само вызывающиеся функции IIFE (Immediately Invoked Function Expression)

- вызов функции сразу при определении

Классы (ООП) (04_class.html)

- **class** - представляет описание объекта
- объект – экземпляр реализации

Описание

- class Student {}
- **constructor** – метод создания объекта
- **this** – для обращения к свойствам объекта

Создание

- оператор **new**

Наследование

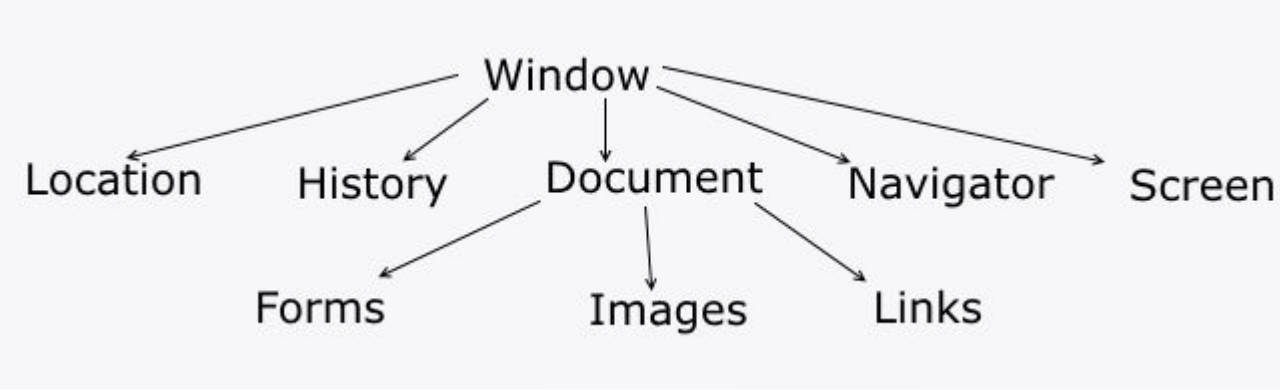
- **extends** – для указания базового класса
- **super** – обращение к конструктору и методам базового класса

Статические методы

- **static** – объявление
- через **this** обращаться нельзя!
- обращаемся к классу, а не к конкретному объекту

Browser Object Model - BOM (05_bom.html)

Объектная модель броузера

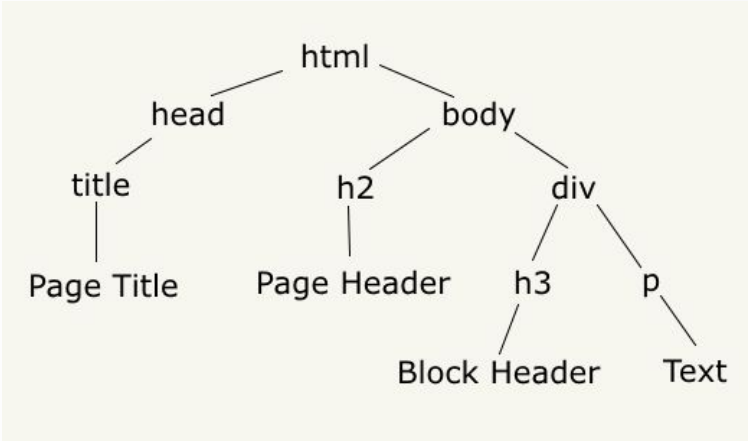


- **window** – основной объект браузера;
  - управление окнами
  - глобальные переменные
- **document** – представляет html-страницу в виде DOM
- **location** – полная информация о расположении текущей страницы
- **history** – предоставляет историю браузера
- **navigator** – информация о браузере, текущей ОС и геолокации

Document Object Model (DOM) (06_dom.html)

Html страница доступна в JS в виде дерева объектов (DOM)

Каждый элемент в иерархии – узел



Некоторые виды узлов:

- **Element** - html-элемент
- **Attr** - атрибут html-элемента
- **Document** - корневой узел html-документа
- **Text** - текст элемента
- **Comment** - элемент комментария

Основные манипуляции с помощью document:

- поиск элементов
- создание добавление удаление
- изменение стиля

Обработка событий (07_event.html)

Некоторые типы событий в JavaScript

- События мыши, события клавиатуры
- События жизненного цикла элементов (например, событие загрузки веб-станицы)
- События элементов форм (нажатие кнопки на форме, выбор элемента в выпадающем списке и т.д.)
- События, возникающие при изменении элементов DOM
- События, возникающие при возникновении ошибок

Типы обработчиков

- встроенные обработчики (inline event handler) – в html элементе
- обработчики как свойства .onclick=*функция_обработчика*
- слушатели событий .addEventListener('click', *функция_обработчика*)

Параметр обработчика – объект Event

- **target**: указывает на элемент, на котором было вызвано событие
- **currentTarget**: определяет элемент, к которому прикреплен обработчик события
- **defaultPrevented**: возвращает true, если был вызван у объекта Event метод preventDefault()
- **timeStamp**: хранит время возникновения события
- **type**: указывает на имя события
- **cancelable**: возвращает true, если можно отменить стандартную обработку события
- **bubbles**: возвращает true, если событие является восходящим

Распространение события: восходящее (bubbles), нисходящее

Хранение данных в браузере (08_cookiewebstor)

Куки (максимальный размер – 4 кб)

- имя куки
- значение
- **expires** – дата/время окончания действия (сеанс по умолчанию)
- **max-age** – кол-во секунд с момента установки куки до удаления
- **path** – доступность куки только для определенного пути (www.site.com/home - path=/home;)
- **domain** – установка для определенного поддомена на мультидоменном сайте
- **secure** – доступность из SSL-соединения (протокол https), false по-умолчанию

document.cookie="имя1=значение1; expires=; path=/; domain=example.com; ..."

WebStorage

- **session storage** – временное хранилище на период веб-сессии
- **local storage** – постоянное хранилище (5мб – Chrome, FF; 10мб - IE)

AJAX – запросы (09_ajax.html)

Asynchronous JavaScript And XML

- асинхронное взаимодействие клиента JS и сервера
- http-протокол (версия 1.1)- методы Get, Post, Put, Delete, Options

Объект XMLHttpRequest

- инициализация - метод open с параметрами:
  - тип запроса – GET, POST, HEAD, PUT
  - url запроса
  - true/false – асинхронный/синхронный запрос (не обязательный)
  - имя и пароль – для авторизации (при необходимости)
- отправка – метод send

Домашнее задание

На свою html-страничку внедрите скрипт, выводящий асинхронным запросом по кнопке или через интервалы – 1час (можно использовать - setInterval) на выбор:

- Курс доллара или евро. Использовать можно, к примеру, сайт

https://www.cbr-xml-daily.ru/. При этом удобнее использовать GET запрос с URL: https://www.cbr-xml-daily.ru/daily_json.js

- Текущую погоду. Например: https://openweathermap.org/api
- Любой другой on-line сервис (расписания, эл. табло, и т.д.) предоставляющий ответы на запросы в формате JSON или XML.

Спасибо за внимание

Контакты

www.digdes.ru

info@digdes.com

Санкт-Петербург
наб. реки Смоленки, д. 33
телефон: +7 812 346 58 33

Москва
Варшавское шоссе, д. 36, стр. 8
телефон: +7 499 788 74 94