# 

# ЗМІСТ

[ЗМІСТ 1](#_Toc429643658)

[1. Мета роботи 2](#_Toc429643659)

[2. Виконання прикладів 2](#_Toc429643660)

[3. Завдання за варіантом. 5](#_Toc429643661)

[4. Виконання індивідуальних завдань 5](#_Toc429643662)

[Завдання 1. 5](#_Toc429643663)

[Завдання 2. 6](#_Toc429643664)

[Завдання 3. 6](#_Toc429643665)

[5. Висновки 7](#_Toc429643666)

# Мета лекції

* Ознайомитись з метою та історією створення JavaScript;
* Розглянути основні засоби розробки і запуску програм на JavaScript;
* Висвітлити відомості про основні конструкції, оператори, ключові слова, структури даних мови JavaScript;
* Розглянути основні довідкові ресурси для JavaScript;
* Надати студентам можливість створювати найпростіші JavaScript-програми.

# План лекції

1. Введення у мову JavaScript.
   1. Що таке JavaScript і де він застосовується.
   2. Що вміє і не вміє JavaScript.
   3. Унікальні особливості JavaScript та альтернативні технології.
   4. Де шукати допомогу.
   5. Засоби розробки програм JavaScript.
   6. Засоби налагоджування і запуску JavaScript.
2. Основи JavaScript
   1. Тег <script> в *html.* Вправа HelloWorld.
   2. Порядок виконання скриптів.
   3. Структура коду: коментарі, команди.
   4. Змінні. Іменування змінних. Типи даних.
   5. Оператори в JavaScript.
   6. Взаємодія з користувачем. Умовні і логічні оператори.
   7. Цикли. Конструкція switch
   8. Функції, функціональні вирази. Рекурсія.
   9. Налагоджування у Chrome. Поради по стилю коду.
3. Вбудовані типи і структури даних.
   1. Методи і властивості. Числа і робота з ними.
   2. Рядки.
   3. Об'єкти як асоціативні масиви. Перебор властивостей. Передача за посиланням.
   4. Масиви. Черга. Методи масивів.
   5. Ітерація по масивам.
   6. Масив аргументів функцій.
   7. Дата і час.
4. Замикання та область видимості.
5. Об'єкти і методи.
   1. Методи в об'єктів. Перетворення об'єктів.
   2. Створення об'єктів через *new.*
   3. Дескриптори, геттери і сеттери.
   4. Статичні методи. Фабричні методи.
   5. Виклики методів: *call, apply.*
   6. Прив’язка контексту
   7. Декоратори.

# Тези лекції

## Введення у мову JavaScript

### а. Що таке JavaScript і де він застосовується.

JavaScript - це мова за допомогою якої сторінки роблять "живими"(тобто вони можуть взаємодіяти із користувачем. наприклад, переміщення картинки чи блоку або зміну кольору екрану можна реалізувати через функції написану цією мовою). Програми написані цією мовою називаються скриптами. В браузері вони підключаються в HTML на пряму. Як тільки сторінка завантажена, скрипт виконується.

JavaScript на початку мав називатись LiveSctipt, але на час маркетологи вирішили, що так як мова Java популярна, то LiveScript можна розпіарити назвавши на його честь.

Ця мова має незалежна, має свою специфікацію - EcmaScript(мова винайдена Бренданом Айком. Об'єктна мова(побудована на взаємодії об’єктів).

Програми на JavaScript виконуються не лише в браузері, а й на сервері. Будь-де, потрібно аби лиш інтерпретатор цієї мови був. Процес виконання коду називається інтерпретацією.

### b. Що вміє і не вміє JavaScript.

Це сучасна мова, яка не надає низькорівневих ресурсів для роботи із памяттю(тобто розмітити її вручну не вийде як в С++).

Використовуючи її можна:

1. Створити нові HTML теги, видаляти існуючі, змінювати стилі елементів, ховати та показувати елементи

2. Реагувати на дії користувача, тобто обробка кліків мишкою, переміщення курсору, натиски клавіатури

3. Надсилати серверу запити і завантажувати дані без оновлення сторінки(ця технологія має назву "ajax")

4. Отримувати і встановлювати куки, виводити повідомлення.

Мінуси JavaScript:

1. Він не має прямого доступу до ОС. Тобто не може зчитати/записати файли на жорсткий диск, копіювати їх чи викликати програми.

2. Обмеження безпеки: незручний запит на інший домен

3. Він не може із однієї вкладки працювати з іншими вкладками

### c. Унікальні особливості JavaScript та альтернативні технології.

* 1. Повна інтеграція із HTML/CSS
  2. Прості речі робляться просто
  3. Підтримується більшістю браузерів

Жодна браузерна технологія не має такого синтезу можливостей.

### d. Де шукати допомогу

У довідниках. Є три основні довідники, вони написані англійською мовою:

1. Mozilla Developer Network. Також вона має розширення для Firefox.

2. MSDN - довідник від Макрасофта. Також вона має інфо для IE.

3. Safari Developer Library - менш відомий, але також напрочуд корисний.

Ось наведені довідники не від розробників браузерів:

1. http://javascript.ru/manual

2. http://caniuse.com (ресурс про підтримку браузерами найновіших можливостей HTML/CSS/JavaScript)

3. http://help.dottoro.com тут інфа про HTML/CSS/JavaScript.

Усі ці посилання можна знайти на веб-сторінці лекції.

### e. Засоби розробки програм JavaScript.

Для розробки потрібен хороший редактор.

Вибраний вами редактор мусить мати:

1. Підсвічування синтаксису

2. Автодоповнення

3. Фолдінг - можливість приховати/знов вивести блок коду

IDE (Integrated Development Environment). Всім відома інтегрована середа розробки із багатьма витребеньками. Вміє працювати із системами, як багтрекерами, контролю версій і багато чого ще.

Як правило, IDE завантажує проект цілком, тому може надавати автодоповнення функцій по всьому проекту, зручну навігацію по файлам.

Окрім IDE, можна використовувати WebStorm, продукти Eclipse, Komodo IDE та Netbeans.

Плюси WebStorm: має LiveEdit(можливість під час редагування без оновлення сторінки одразу бачити результат змін у коді), підтримує HTML5, є підтримка Emmet.

Для роботи із JavaScript доками в Eclipse треба плагін встановити. Komodo IDE має гарну підсвітку синтаксису(яку можна налаштувати за своїм смаком - ось http://mintmem.com/page/komodo-edit\_4 стаття в якій детально описано це). Netbeans має багатомовний редактор: Java, C/C++, Ruby, Groovy, PHP, JavaScript, CSS, XML, HTML, RHTML, JSP. Редактор перевіряє правильність слів і скобок, підсвічує синтаксис вихідного коду, перевіряє на помилки код, під час введеня цього ж коду, також є автозавершення коду та фрагментів документації. Редактор генерує і вставляє у висхідний код стандартні фрагменти на Джава чи інших мовах. Будь-яку вкладку робочого простору можна перемістити у бажане місце. Можна групувати пов'язані проекти.

Також є легкі редактори. Це:

* Sublime Text (крос-платформений, shareware).
* Atom (крос-платформений, free).
* SciTe простий, легкий и швидкий (Windows, безкоштовний).
* Notepad++ (Windows, безкоштовний).

Легкі редактори не такі могутні, як IDE. Вони мало вісять чи прості, швидко стартують.

### f. Засоби налагоджування і запуску JavaScript.

Аби подивитись, яких ви помилок наробили в коді, в:

* Google Chrome треба зайти на сторіночку bug.html. В її JavaScript-коді є помилка. Звичайний користувач її не побачить. Для того аби її помітити треба відкрити інструменти розробника (F12 под Windows, а для Mac, то Cmd+Opt+J). Після натиску зявиться вкладка Console, в ній і буде помилка.
* Firefox треба встановити розширення Firebug.
* Safari треба активувати меню розробника.

1. **Основи JavaScript**

### Тег <script> в html. Вправа HelloWorld.

Програми на JavaScript вставляються в будь-яке місце HTML-коду за допомогою тега SCRIPT. Тег

<script> ... </script>

містить виконуваний код. Коли браузер бачить цей тег, він:

* 1. Починає показувати сторінку до тегу
  2. Бачить тег, виконує скрипт
  3. Закінчивши виконання, повертається назад в HTML-режим і тоді показує решту документу.

alert(повідомлення) – показує текст повідомлення.

### Порядок виконання скриптів.

Коли коду багато, його виносять в окремий файл, який можна підключити в HTML.

<script src="/path/to/script.js"></script>

Тут так само не показується решта сторінки до повного виконання скрипта, а скрипти можуть грузитись довго – тому є атрибути *async* та *defer.*

* *Async –* синхронна загрузка скриптів
* *Defer –* послідовна загрузка скриптів.

### Структура коду: коментарі, команди.

Команди бажано розділяти крапкою з комою. Коментарі мають вигляд такий самий, як і в Java. Підтримується документуючий коментар.

## *Змінні. Іменування змінних. Типи даних.*

Змінна об’являється так:

var message;

Після об’явлення, в змінну можна записувати дані.

Для іменування змінних використовують camelCase, змінні називають англійською мовою.

Є 5 примітивних типів даних: number, string, boolean, null, undefined і 6 тип – object.

### Оператори в JavaScript.

* Для складання рядків +
* Точна рівність ===
* Логічні оператори &&, ||, !
* Звичайні арифметичні оператори, інкремент і декремент.

### Взаємодія з користувачем. Умовні і логічні оператори.

* alert(повідомлення)
* prompt(title, default)
* confirm(question)
* if(), else, else if.

### Цикли. Конструкція switch

Як в Java.