**Структура web сторінки**

**window** – глобальний об’єкт, який є вікном браузера і містить в собі інтерфейс для роботи із компонентами браузера

**Объектная Модель Документа (DOM)** – це програмний інтерфейс API для роботи із HTML i XML сторінками. DOM надає структуроване уявлення документа і визначає те, як ця структура може бути доступна з програм, які можуть змінювати вміст, стиль і структуру документа. Подання DOM складається з структурованої групи вузлів і об'єктів, які мають властивості і методи. DOM - це уявлення HTML-документа у вигляді дерева тегів.

(<https://learn.javascript.ru/dom-nodes>, [https://docs.google.com/presentation/d/1LA4\_b6LrYjZ8WpfwPdMxKPf0IMZ\_nFzHDdjITPUpsgE/edit#slide=id.g5dbb3f8f21\_2\_140](https://docs.google.com/presentation/d/1LA4_b6LrYjZ8WpfwPdMxKPf0IMZ_nFzHDdjITPUpsgE/edit) , <https://developer.mozilla.org/ru/docs/Web/API/Document_Object_Model/Introduction> )

Але на практиці ми в основному працюємо з 4 з них:

* **document** - «вхідні точка» в DOM. document є власним кореневих об'єктом.
* **вузли-елементи (elements)** - HTML-теги, основні будівельні блоки. Element - позначає елемент або вузол типу element, що повертається членом DOM API
* **текстові вузли** - містять текст.
* **коментарі** - іноді в них можна включити інформацію, яка не буде показана, але доступна в DOM для читання JS.

Типи данних:

* **Вузол** – будь-який текст, коментар, елемент в dom
* **Колекція** (HTMLCollection)– перебераємий об’єкт псевдомасив (перебереється for of, не містить доступ до методів масива). Колекція це жива структура, масивоподібне представлення dom елементів, при зміні який в дом структурі, зміни відобразяться і в живій колекції
* **Не жива колекція (static)** (NodeList) **–** колекція, яка не міняється в момент зміни структури dom дерева

**BOM** – інтерфейс для роботи із усім, окрім документа

* **navigator** – об’єкт для отримання інформації про браузер і операційну систему
* **location** – об’єкт для взаємодії із шляхом до сторінки
* **history** – дає доступ до історії браузера, дозволяючи переміщатись по ній
* **screen** – інформація про дісплей (розміри, розширення)
* **frames** – колекція зі всіма вікнами на поточній сторінці

**Методи для роботи із DOM**

**Методи виведення інформацї:**

* **alert(**‘msg’**)** – виведення інформації вспливаючим повідомленням
* **confirm(‘**msg’**) –** виводить повідомлення вспливаючим вікном із кнопками для підтвердження та заперечення. **Повертає true/false** в залежності від обраної відповіді
* **prompt(**‘msg’) – виводить вспливаюче повідомлення в вікні + кнопки + можливість ввести свій текст. **Повертає null – якщо натиснуто ‘скасувати’, текст введений користувачем – якщо підтведжено.**

**Методи отримання точок входу в DOM структуру:**

* **document.documentElement** – повертає тег **html**
* **document.body**– повертає тег **body**
* **document.head**– повертає тег **head**
* **document / element**.**querySelectorAll(`**selector**`) –** повертає **не живу колекцію усіх елементів,** які містять вказаний селектор в межах об’єкта пошуку **(NodeList)**
* **document / element**.**querySelector (`**selector**`) –** повертаєелемент**,** відповідає вказаному селектору, в межає об’єкта пошуку **(NodeList)**
* **document / element**.**getElementById(`**id**`) –** повертає **колекцію усіх елементів,** які містять вказаний id в межах об’єкта пошуку **(HTMLCollection)**
* **document / element**. **getElementsByTagName(`**tag**`) –** повертає **колекцію усіх елементів,** які містять вказаний тег в межах об’єкта пошуку **(HTMLCollection)**
* **document / element**. **getElementsByClassName(`**class**`)** повертає **колекцію усіх елементів,** які містять вказаний клас в межах об’єкта пошуку **(HTMLCollection)**
* **document / element**. **getElementsByName(`**attribute-name-value**`)** повертає **колекцію усіх елементів,** які містять атребут name з вказаним значенням **(NodeList)**
* **element.closest(**‘selctor’**)** – рухається по ланцюгу батьків елемент і шукає той, що задовільняє переданий селектор. **Повертає перший батьківський елемент з ланцюга**, що задовільняє селектор
* **element.matches(**‘selector’**) – Повертає true/false**, якщо елемент співпадає / містить вказаний селектор

**Методи роботи із вузлами DOM**

**Методи навігації по вузлам DOM дерева:**

* **element.firstChild** – повертає тег **перший дочірній вузол** (елемент, текст, комнтар)
* **element.lastChild**– повертає тег **останній дочірній вузол**(елемент, текст, комнтар)
* **element.childNodes – повертає неживу колекцію (**NodeList**) дочірніх вузлів блока** (виглядає, як масив, але не здатний використовувати їх базові методи)
* **element.previousSibling – повертає попередній вузол об’єкта**, що знаходиться на тому ж рівні вложеності
* **element.nextSibling – повертає наступний вузол об’єкта**, що знаходиться на тому ж рівні вложеності
* **element.parentNode – повертає батьківський вузол елемента**

**Методи перевірки DOM елемента на наявність вузлів:**

* **element.hasChildNodes()** – **повертає true/false** в залежності від наявності дочірніх **вузлів** в елемента

**Методи роботи лише з елементами DOM**

**Методи навігації лише по елементам DOM дерева:**

* **element.children** – **повертає живу колекцію** (HTMLCollection) **дочірніх елементів** елемента
* **element.firstElementChild** – повертає **перший дочірній елемент**
* **element.lastElementChild**– повертає **останній дочірній елемент**
* **element.previousElementSibling – повертає попередній елемент об’єкта**, що знаходиться на тому ж рівні вложеності
* **element.nextElementSibling – повертає наступний елемент об’єкта**, що знаходиться на тому ж рівні вложеності
* **element.parentElement – повертає батьківський елемент елемента**

**Методи зміни вмісту елемента:**

* **element.innerHTML**= ‘’ – Без присвоєння повертає html текстовий вміст (вузол) елемента. З присвоєнням дозволяє вписати в елемент будь-яку структуру в вигляді строки. Присвоювати можна і елементи, які перетворяться в dom елементи
* **element.outerHTML – Повертає нам в текстовому вигляді елемент в вигляді dom структури із усім його вмістом. Дозволяє присвоювати нову стурктуру в елемент.** Отримаємо завжди оригінальну версію елемента, навіть при переопреділені даного елемента даною властивістю
* **element.textContent – повертає лише текстовий вміст елемента.** Будь –яка структура dom вписана в елемент переведеться в строку
* **element**.**data** – **Поверне вміст текстового вузла чи коментаря.** Можемо переприсвоїти даний вміст

**Методи створення DOM структури:**

* **document.createElement(**‘tag-name’**)** – **повертає елемент вкзаного тегу.**
* **document.createFragment() –** Створює обгортку для html структури, яка зникає при монтуванні структури в dom
* **document.createTextNode(**‘text’**)** – створює і **повертає текстовий вузол**
* **element.after(**element-new**) – вставляє element-new після element**
* **element.before(**element-new**) - вставляє element-new до elemen**
* **element.prepend(**element-new**) - вставляє element-new в elemen перед усіма дочірніми елементами**
* **element.apend(**element-new**) - вставляє element-new в elemen після усіх дочірніх елементів**
* **element.insertAdjacentHTML(**‘position’, ‘structure’**) – вставляє dom структуру описану в строці в позицію**
* ‘beforebegin’ – вставляє html перед element
* ‘afterbegin’ - вставляє html в початок element перед усіма дочірніми елементами
* ‘beforeend’ - вставляє html в кінець element після усіх дочірніх елементів
* ‘afterend’ – вставляє html після елемет
* **element.cloneNode()** – Копіює вузол. Якщо передати параметр true – буде скопійовано і вміст елемента
* **element.remove()** – видаляємо елемент

**Методи взаємодії з атребутами елементів:**

* **element.className** – Повертає імена класів елемента. Можемо переопреділити клас через =
* **element.classList.add(**‘className’**)** – додає клас
* **element.classList.remove(**‘className’**)** –видаляє клас
* **element.classList.toggle(**‘className’**)** – тоглить клас
* **element.classList.contains(**‘className’**)** – перевіряє на наявність класу
* **element.style.propertyName** = ‘value’ – задає стилі в атребут style. Властивість більше 2-х слів записують стилем lowerCamelCase. Без присвоєння, поверне значення даної властивості, якщо воно записано в атребуті style
* **element.style.setProperty(**‘property-name’, ‘value’**) –** встановлює css властивості для даного елемента
* **element.style.cssText = `` -** дозволяє записати одразу декілька стилів в css фоматі в вигляді строки. Повністю перезаписує стилі даного елемента
* **getConputedStyle(element)**.**propertyName** – повертає значення стилю, який нас цікавить. Лише отримуємо властивості. Другий необов’язковий параметр даної фукнції строчний, в якому вказуємо певдоелементи, даного елемента, стилі якого ми хочемо отримати **“::before”.** Даний метод використовується лише для питання
* **link.href** – **Повертає шлях вказаний в атребуті**
* **input.value – Повертає значення поля**
* **element.id – Повертає id**
* **element.hasAttribute(**‘name’**) –** перевіряє наявність атребута**. Повертає true/false**
* **element.getAttribute(**‘name’**) – повертає значення атребута**
* **element.setAttribute(**‘name’, ‘value’**) –** задає значення атребута
* **element.removeAttribute(**‘name’**)** – видаляємо атребут
* **element.dataset.attributeName** – Повертає значення data-attrebute. Також даний атребут можна переопреділити через ‘=’.
* **element.tagName** – повертає ім’я тега в uppercase
* **element**.**nodeName** – повертає назву елемента
* **element.nodeType** - повертає тип елемента
* nodeType === 1 – елемент
* nodeType === 3 – Текст
* nodeType === 2 – атребут

**Forms**

**Отримання колекції форм**

**document.forms** – отримання колекції усіх форм на сторінці, **жива колекція (**HTMLCollections**)**

**document.forms.main** – отримуємо конкретну форму по значен атребута name (main – в даному випадку)

**input.form** – знаючи інпут, отримуємо батьківську форму

**Отримати поля форми**

**document.forms[0] || document.forms.main.elements –** поверне живу колекцію усіх елементів, що знаходяться в даній формі (якщо в elements задати значення name - input, то ми отримаємо дане поле **document.forms.main.elements.inputMain**)

**document.form.main.inputMain** – поверне поле, значення атребуту name якого збігається із останній параметром даної конструкції (inputMain в даному випадку)

* При звернені до поля input з типом radio, атребут name, якого повинен бути однаковими, ми отримаємо живу колекцію даних полів

**Взаємодія із полями різного типу**

**Текстовими**

**input.value – отримаємо значення поля.** Значення можна як отримати, так і задати. Може являтись атребутом поля (вручну в html значення поля можна задати лише через цей атребут)

**textarea.value** **- отримаємо значення поля**.

**Radio / Checkbox**

**radio**.**value** - ‘on’ по-замовчуванню – можна задати при потребі

**radio**.**chacked** – true/false в залежності від статусі обраного поля

**File**

**input.value –** поверне ім’я файлу, який ми передали в поле або пусту строку

**Select**

**Для мультивибору** додаємо select атребут **multiply**

**select.options –** колекція options (**HTMLCollection**)

**select.value –** поверне значення вибраного на даний момент option. При присвоєні числового значення, що позначає порядковий номер option, то вибираємо його як selected

**select.selectedIndex –** номер вибраного на даний момент option

**option.selected –** true/false вибрана опція

**select.options[selectedIndex].text –** текстове значенняконкретного option

**option.index –** індекс поточного option

**Attributes**

**multiple** – **задається select** тегу

**selected** – вибирає дефолтний option, **задається option**

**tabindex** = ‘number’ **–** задає чергу по якій ми переключатимемось по полям табом, **задається будь-яким полям**

* **tabindex** = 0 –Зазвичай додається для того, щоб фокус спрацьовував на елементі із даним атребутом, але не міняв порядок переключення tab кнопкою. Будь-який елемент із даним значенням атребута, може прирівнятись до input і стати сприєнятливим до події **focus** і **blur**
* **tabindex** = -1 – фокус можна встановити лише програмно, кнопка tab ігноруватиме дане поле. Будь-який елемент із даним значенням атребута, може прирівнятись до input і стати сприєнятливим до події **focus** і **blur**

**Для динамічного створення нового option**

new Option(text, value, defaultSelected(true/false), selected(true/false)) –

**text** – текст в середині options

**value** – вставляє значення value

**defaultSelected** – true/false – при true ставиться html атребут selected

**selected** – true/false – true елемент option буде вибраний

**Події Форм**

**focus** – спрацьовує **на фокус** поля

**blur** – спрацьовує **на розфокус** поля

**focusin** – подія, яка завішується на форму і по замовчуванню вспливає, коли **focus** спрацьовує на полі

**focusout** – подія, яка завішується на форму і по замовчуванню вспливає, коли **blur** спрацьовує на полі

**change** – спрацює, коли ми розфокусовуємо поле, в яке вводили дані. На **checkbox, radio, select, file** - спрацює одразу після зміни даних.

**input** – спрацює, коли ми вводимо дані в поле

**cut** – спрацює при вирізанні тексту із поля

**copy** – копіювання тексту із поля

**paste** – спрацює при вставлені тексту в поле

**submit** – спрацює при відправці даних (input / button з типом submit, )

**Методи для роботи із розмірами проктрутки і координатами елементів**

<http://www.wisdomweb.ru/JSd/scrawidth.php>

[https://developer.mozilla.org/ru/docs/Web/API/Element/scro HYPERLINK "https://developer.mozilla.org/ru/docs/Web/API/Element/scrollHeight" HYPERLINK "https://developer.mozilla.org/ru/docs/Web/API/Element/scrollHeight" HYPERLINK "https://developer.mozilla.org/ru/docs/Web/API/Element/scrollHeight"llHeight](https://developer.mozilla.org/ru/docs/Web/API/Element/scrollHeight)

<https://developer.mozilla.org/ru/docs/Web/API/Element/clientHeight>

<https://developer.mozilla.org/en-US/docs/Web/API/Window/pageYOffset>

**Методи отримання розмірів елементів :**

* **window.innerWidth/innerHeight – поверне ширину/висоту сторінки в видимій області вікна з полосою прокрутки**
* **element.scrollWidth/scrollHeight – поверне повну ширину/висоту елементаочей,** включно із тією частиною, яка не видима через прокрутнку, але без border, padding, scrollbar

**(element.scrollHeight = element.clientHeight + element.scrollTop)**

* **element.offsetWidth/offsetHeight – поверне повну ширину/висоту елемента,** включно із padding, border і полосою прокрутки, але виключаючи margin. Якщо елемент прихований display:none – отримаємо 0

Якщо це html, то поверне весь розмір, навіть прихованої частини екраном

* **element.clientWidth/clientHeight – поверне ширину/висоту елемента** включно із padding, але виключаючи border, margin і полосу прокрутки.

Якщо html тег, то поверне його розміри в видимій області вікна браузера, без полоси прокрутки

**Метод для взаємодії із параметрами скролу:**

* **window.pageYOffset/pageXOffset – повертає значення прокручених пікселів** по ширині / по висоті. Значення не динамічне
* **window.scrollBy(**x, y**)** – прокручує елемент з прокруткою на вказані параметри. Додає значення до вже поточної позиції скролу сторінки, при повторному виклику функції
* **window.scrollTo(**x, y**)** – прокручує елемент з прокруткою на вказані параметри, без додавання значення до поточної позиції скролу. Містить в собі об’єкт налаштувань **{top: num, left: num, behavior: ‘smooth || instant’}** – плавна || різка (не працює в сафарі)
* **element.scrollIntoView(**{ } || false**)** – **без параметрів** верхня межа браузера буде вирівняна по верху видимої частини елемента до якого ми прокручуємо, з параметром false – нижня частина вікна браузера буде вирівняна по низу елемента до якого прокручуємо, з параметрами можна налаштувати плавну прокрутку

**{**

* **block:** ‘куди - вертикальне вирівнювання блока до якого проскролюємо в межах видимої частини вікна ’ (start, center, end, nearest )**,**
* **inline:** ‘горизонтальне вирівнювання вирівнювання блока до якого проскролюємо в межах видимої частини вікна’(start, center, end, nearest )**,**
* **behavior:** ‘тип анімації’ ( smooth, auto )**,**

**}**

**Методи метрики (визначаючі позицію елемента на сторінці):**

* **element.offsetParent** – **отримаємо найближчий батьківський елемент, відносно якого позиціонується наш об’єкт**, якщо:
* даний батьківський елемент задовільняє умови – являється css – позиціонованим (relative, absolute, fixed, stratch),
* тегом <td>, <th>, <table>
* <body> – відносно якого зазвичай позиціонуються всі елементи
* Або отримуємо null – якщо поточний елемент прихований display: none; чи відсутній в документі, для елементів body i html
* **element.offsetLeft/offsetTop** – **повертає значення зміщення** поточного елементу відносно його батьківського елемента (**offsetParent**) зверху і зліва
* **element.clientLeft/clientTop** – повертає значення відступів внутрішньої частини елемента (content) від зовнішньої скрол, border і padding
* **event.pageY/pageX** - властивість працює відносно документа і **повертає відстань від лівого верхнього кута чи верхнього лівого кута документа до елемента по вертикалі/горизонталі**.
* **event.clientY/clientX** - властивість працює відносно видимого вікна і **повертає відстань від лівого верхнього кута до елемента по вертикалі/горизонталі**.
* **element.scrollLeft** / **scrollTop –** призначені не лише для читання, а й можна задавати початково прокручені пікселі. Повертає к-сть прокручених пікселів елемента
* **element.getBoundingClientRect()** – повертає об’єкт із парметрами елемента. Значення можуть бути від’ємні, якщо елемент. **(**

**{ height:,**

**width:,**

**left:,**

**right:**’позиція від правого кута сторінки до лівого краю вікна браузера’**,**

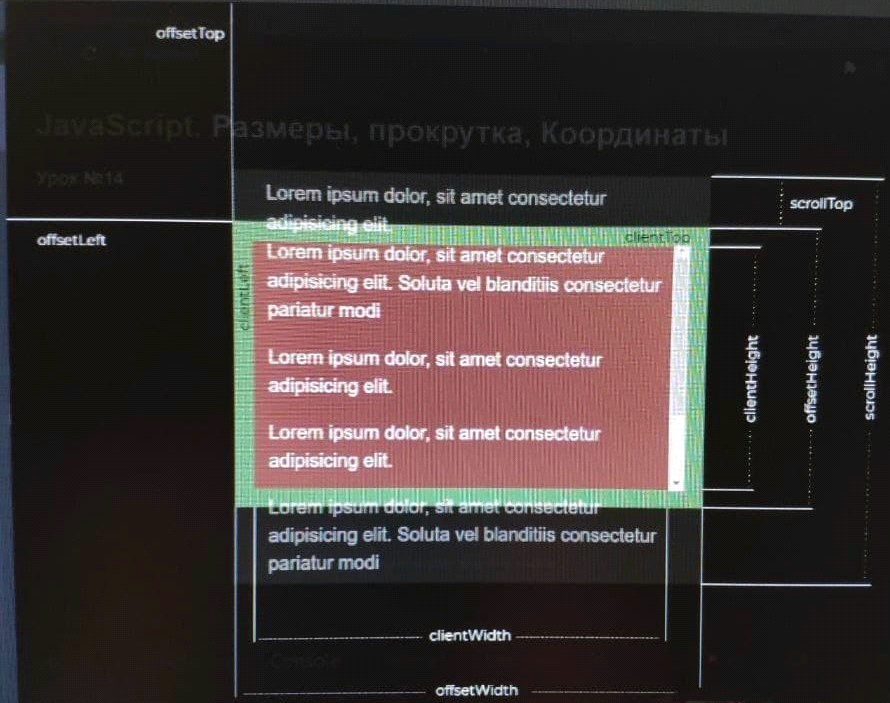
**top:,**

**bottom:** ‘відстань від нижньої границі елемента до верхнього краю вікна браузера’**,**

**x: left,**

**y:top}**) всі параметри відносно вікна браузера

* **document.elementFromPoint()** – повертає об’єкт, що знаходиться за заданими координатами. Всі координати шукаються відносно вікна браузера.



**Події в JS**

* [https HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events":// HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events"learn HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events". HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events"javascript HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events". HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events"ru HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events"/ HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events"introduction HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events"- HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events"browser HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events"- HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events" HYPERLINK "https://learn.javascript.ru/introduction-browser-events"events](https://learn.javascript.ru/introduction-browser-events)
* [https HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/":// HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"frontender HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/". HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"info HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"/ HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"an HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"- HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"introduction HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"- HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"to HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"- HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"dom HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"- HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"events HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/" HYPERLINK "https://frontender.info/an-introduction-to-dom-events/"/](https://frontender.info/an-introduction-to-dom-events/)
* Події мишкою - [https HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave":// HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"learn HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave". HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"javascript HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave". HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"ru HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"/ HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"mousemove HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"- HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"mouseover HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"- HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"mouseout HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"- HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"mouseenter HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"- HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"mouseleave HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"# HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"sobytiya HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"- HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"mouseover HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"- HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"mouseout HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"- HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave" HYPERLINK "https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave"relatedtarget](https://learn.javascript.ru/mousemove-mouseover-mouseout-mouseenter-mouseleave)
* [https HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14":// HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14"karmazzin HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14". HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14"gitbook HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14". HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14"io HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14"/ HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14"eloquentjavascript HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14"\_ HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14"ru HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14"/ HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14"chapter HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14" HYPERLINK "https://karmazzin.gitbook.io/eloquentjavascript\_ru/chapter14"14](https://karmazzin.gitbook.io/eloquentjavascript_ru/chapter14)
* Список всіх подій -[https://docs.google.com/presentation/d/1o2wkXfqEIEv7FArGBAQbrIlVe-UBJUeccl703iQMym4/edit#slide=id.g5dbb70b8be\_2\_65](https://docs.google.com/presentation/d/1o2wkXfqEIEv7FArGBAQbrIlVe-UBJUeccl703iQMym4/edit)
* <https://www.w3schools.com/jsref/dom_obj_event.asp>
* Глибоке занурення в події - <https://frontender.info/an-introduction-to-dom-events/>
* Події для мобільних пристроїв - <https://habr.com/ru/post/118318/>
* Події для мобільних пристроїв - <https://www.manhunter.ru/webmaster/807_obrabotka_sobitiy_tachskrinov_na_javascript.html>
* Події для мобільних пристроїв - <http://youon.ru/%D0%90%D0%BD%D0%B4%D1%80%D0%BE%D0%B8%D0%B4/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0/touch-sobytiya-na-javascript-multitach-realizatsiya>
* <https://docs.google.com/presentation/d/1o2wkXfqEIEv7FArGBAQbrIlVe-UBJUeccl703iQMym4/edit#slide=id.g5dbb70b8be_2_95>

**Делігірування подій** – опрацювання подій на батьківському елементі з провіркою чи відбулась подія на конкретному дочірньому вузлі

addEventListener(event, handler(event), options)

Перевагою даного методу перед прописуванням атребута подій в елемента html є те, що ми можемо його гнучко налаштовувати і навішувати на один елемент декілька різних подій.

**options** = {

* **‘capture’**- фаза на якій спрацює функція
* **‘once’** – видалення функції події після першого спрацювання
* **‘passive’** – якщо true – ніколи не викличемо preventDefault в методі

}

**event** – аргумент, який передається в метод опрацювання подій, являється об’єктом і має такі властивості:

e.**type** – **поверне тип події**, строкою

e.**target – поверне елемент на якому спрацювала подія**

e.**currentTarget – поверне безпосередній елемент на якому завішана подія** і навішаний данийобробник подій, без жодних внутрішніх елементів

**Вспливання** - коли при кліку на дочірній елемент, по черзі вгору по батьківським елементам спрацьовують їх події, якщо типи цих подій співпадають. Для зупинки ефекту вспливання використовується властивість об’єкта подій **event.stopPropagation()**

**Погруження –** перш ніж всплити, подія спершу занурюється до дочірніх елементів, а тоді вспливає. **Для активації ефекту погруження чи всплиття**, третім параметром задаємо addEvent.Listener – об’єкт із параметром ‘capture’: true.

**Делігірування подій** – принцип навішування подій одного типу і реалізації, не на кожен елемент, а на батьківський блок, що зменшує навантаження на систему меншою к-стю оброботчиків подій.

**Скасування базової поведінки браузера** – для скасування базової поведінки браузера (перезагрузка сторінки, …) викликаємо ще одну властивість об’єкта подій **event.preventDefatult()**

Для того, щоб браузер не тратив час на перевірку чи є **event.preventDefatult()** в оброботчику подій, ми можемо в третій парметр об’єкт options передати **‘passive’**: **true**

**Подї миші**

**Прості:**

**mousedown / mouseup** – кнопка миші нажата / відпущена

**mouseover / mouseout** – курсор миші наведений на елемент / відведений із нього. Дані події вспливають, тобто наведення на дочрні елементи враховуються, потрібно, коли важливо взаємодіяти із дочірніми елементами можемо використати делігірування

**mouseenter / mouseleave –** тіж події, що вище, але переходи між дочірніми елементами не рахуються, тобто вспливання не відбувається

**Об’єкт подій:**

**event.target –** елемент на який курсор зайшов / з якого пішов

**event.relatedTarget -** елемент з якого курсор зійшов / на який курсор перейшов

**mousemove** – кожен рух над елементом генерує дану подію

**Об’єкт подій:**

**event.clientX/clientY – координати курсора вдносно вкна браузера**

**contextmenu** – Викликає контекстне меню

**Комплексні:**

**click** – Викликається при mousedown, а після mouseup над одним і тим же елементом мишки

**dblclick** Викликається при спрацюванні двічі на одному елементі події click з коротким інтервалом

**Об’єкт подій:**

**event.which – поверне номер кнопки** на який відбувся клік

**Події клавіатури**

**keydown –** подія натиску на кнопку

**keyup –** подія відтиску кнопки

**Об’єкт подій:**

**event.code –код кнопки** (популярне для роботи з гарячими клавішати, так як не враховує регістр і мову)

**event.key – назва кнопки** (включає регістр і розкладку клавіатури)

**event.repeat – поверне true,** якщо кнопка зажата

**Події скролу**

**scroll** – скрол сторінки

**Об’єкт подій:**

**window.scrollY –** властивість, яка виведек-сть просткроленихпікслів зверху сторінки

**Події загрузки сторінки**

**DOMContentLoaded** – загрузка сторінки, спрацює на **document.**(Спрацює при загрузці dom)

**Load –** спрацює на **window** (варто використовувати, якщо функціонал залежить від зображень чи зовнішніх ресурсів, так як вміст спрацює при повній загрузці сторінки)

**document.readyState –** поверне нам статус заргрузки сторінки

**loading**

**iterative –** загружена видима частина( уже тоді спрацює DOMContentLoaded )

**complete –** загружена видима частина і всі файли (лише тоді спрацює load)

**Події перезагрузки сторінки сторінки**

**beforeunload** – спрацює при спробі покинути сторінку

В оброботчику подій прописати **event.returnValue** = ‘’ і **event.preventDefault()**

**unload** – спрацює коли користувач покинув сторінку, а в фоновому режимі

**Події форми**

**focus** – подія при кліку на поле і готовність введення даних

**blur** – подія розфокосування

Події можна реалізовувати і **input**.**focus()/blur()**

**change –** спрацьовує, при внесені змін в поле (для текстових полів спрацьовує після втрати фокусу, для select, file, radio, checkbox - одразу)

**input** – спрацьовує одразу кожного разу при внесені змін в поле

**copy** – спрацює при копіюванні даних із поля

**paste** – спрацює при вставлені даних в поле

**cut** – спрацює при копіюванні із поля

Зупинити дані зміни в полі можна вже відомим event.preventDefault()

**submit** – спрацює при відправці форми

Можна дану властивість викликати для форми **form.submit()**

**Метод видалення іменованого метода для обробки подій:**

**element**.**removeEventListener(**‘typeEvent’, ‘nameOfEventFuncion’**)** – видяляє події з не анонімними функціями реалізації, для цього функцію, яка реалізує якусь подію, треба об’явити явно і дати їй ім’я

**Вспливання події**

**Вспливання події** – це коли на елементі відбувається опрацювання подій спершу на ньому, потім на його батьку і так вверх по ланцюгу батьківських елементів. Якщо на них ми ловимо події того ж типу, то навіть без безпосередньої взаємодії із ними ми отримаємо спрацювання події

Якщо в батьківського і дочірнього елемента тип події одинаковий, то при виклику події для дочірнього елемента, реалізується подія і батьківського елементу: від дочірнього до батьківського

Для рішення використовуємо для найглибшого дочірнього елемента метод **e.stopPropogation()**

**Занурення:**

Для задання **навмисного знурення третім** параметром addEventListener - прописуємо true або об’єкт: {capture: true}, але тоді не працюватиме вспливання, для того, щоб працювали обидві фішки, додаємо такі ж оброботчики подій, без третього параметра

**Для того, щоб позбутись обох властивостей**, пишемо обробку подій без вказування третьго параментра option, таким чином занурення не активовується, а на найглибшому дочірньому елементі e.stopPropogation()

**Приклад на занурення і винурення:**

[http HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output":// HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output"jsbin HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output". HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output"com HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output"/ HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output"valasomeli HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output"/ HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output"edit HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output"? HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output"html HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output", HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output"js HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output", HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output" HYPERLINK "http://jsbin.com/valasomeli/edit?html,js,output"output](http://jsbin.com/valasomeli/edit?html,js,output)

**Збереження змін на сайті**

**LocalStorage SessionStorage Cookies**

<https://learn.javascript.ru/localstorage>

**localStorage, sessionStorage –** об’єкти веб-сховища, для **збереження даних в вигляді пари ключ-значення**. Дані, які в них записані зберігаються після перезагрузки сторінки в випадку **sessionStorage** і навіть після перезагрузки браузера **localStorage.**

**Їх відмінність від cookie**

* об’єкти веб сховища не відправляються на сервер при кожному запиті, тому в них ми можемо зберігати в різи більше даних, в середньому від 2-х мегабайт.
* Сервер не може маніпулювати об’єктами сховища через http-заголовки
* Сховище прив'язане до джерела (домен / протокол / порт). Це означає, що різні протоколи або піддомени визначають різні об'єкти сховища, і вони не можуть отримати доступ до даних один одного.

**Методи взаємодії із localStorage i sessionStorage:**

* **Local/sessionStorage.getItem(**‘key’**) –** отримати значення за ключем
* **Local/sessionStorage**.**setItem(**‘key’, ‘value’**) –** задати значення за ключем
* **Local/sessionStorage**.**removeItem(**‘key’**)** – видалити значення за ключем
* **Local/sessionStorage**.**clear() –** видаляє всі дані
* **Local/sessionStorage.key(**index**)** – получити ключ по заданій позиції
* **Local/sessionStorage.length** – отримати кількість елементів сховища

**localStorage**

**Значення ключ-значення задаються строкою**

* Цей об'єкт один на все вкладки та вікна в рамках джерела (один і той же домен / протокол / порт). Нам досить перебувати на тому ж джерелі (домен / протокол / порт), при цьому URL-шлях може бути різним. Об'єкт localStorage доступний всім вікнам з одного джерела, тому, якщо ми встановлюємо дані в одному вікні, зміни стають видимими в іншому.
* Дані не мають терміну давності, за яким закінчуються і видаляються. Зберігаються після перезапуску браузера і навіть ОС.

**sessionStorage**

**Значення ключ-значення задаються строкою**

1. sessionStorage існує тільки в рамках поточної вкладки браузера.

a. Інша вкладка з тієї ж сторінкою матиме інше сховище.

b. Але воно розділяється між іфреймамі на тій же вкладці (за умови, що вони з одного і того ж джерела).

2. Дані продовжують існувати після перезавантаження сторінки, але не після закриття / відкриття вкладки.

!!Але якщо в цих методах нічого немає, то вони не виведуть помилку: в першому випадку null, во втором underfind

///const user = {name: ‘denis’}

///localStorage.setItem(‘user’, **JSON.stringify(user)**) – об’єкт переходить в форматі JSON

**Щоб розпарсити цей об’єкт:**

/// const user = **JSON.parse(localStorage.**user**)**

!!Якщо в localStorage **–** немає даного об’єкту**, то повернеться помилка, тому**

const user = **JSON.parse(localStorage.**getItem(‘user’)**);**

**Local Storage / Sesion Storage**

**Sesion Storage** – зберігає зміни до перезагрузки браузера чи сторінки

**Local Storage** – зберігає зміни навіть при перезагрузці сторінки чи браузера, але вразливий, до чистки даних, кеша…

**Методи для роботи з Storages:**

**! в LocalStorage – ми можемо передавати масиви, або об’єкти в форматі JSON**

* localStorage.setItem(“keyName”, JSON.stingify(object)) – **вписуєму змінні в локальну історію**
* JSON.parse(localStorage.getItem(‘key’))– **отримуємо значення змінної локальної сесії без помилки**
* LocalStorage.remove(“keyName”) – **видаляє з локальної історії змінну із цим іменем**
* LocalStorage.clear() – повністю очищає локал сторедж