**Лабораторна робота № 9**

**Тема:** Робота з вікнами

**Мета**: Управління вінка на стороні користувача

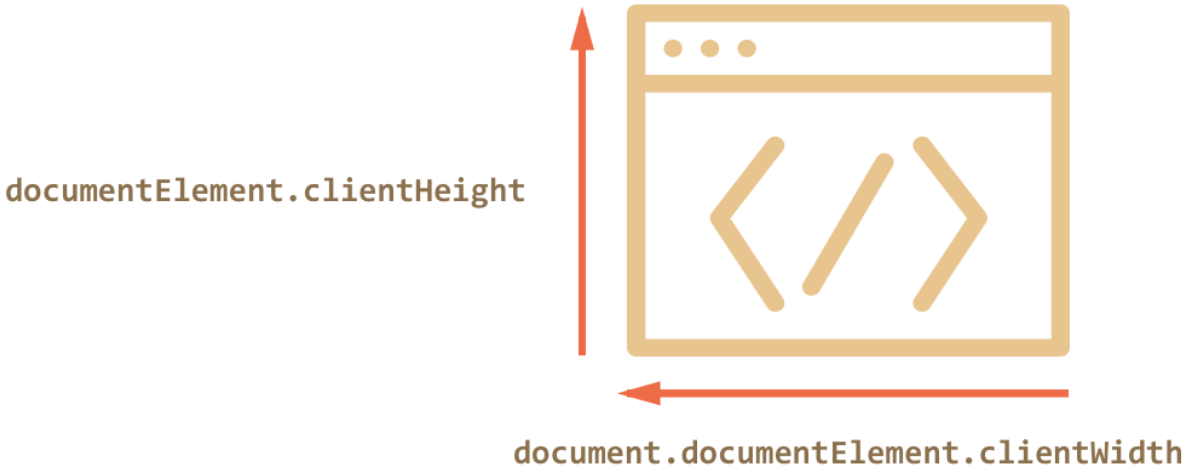
## КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

### Розміри і прокрутка сторінки

З точки зору HTML, документ - це document.documentElement. У цього елемента, відповідного тегу <html>, є всі стандартні властивості і метрики і, в теорії, вони і повинні нам допомогти. Однак, на практиці є ряд нюансів, саме їх ми розглянемо в цьому розділі.

#### *Ширина / висота видимої частини вікна*

Властивості clientWidth/Heightдля елемента document.documentElement- це якраз ширина / висота видимої області вікна.




Наприклад, кнопка нижче виведе розмір такої області для цієї сторінки:

```
alert (document.documentElement.clientHeight)
```

```
alert( window.innerWidth ); // вся ширина вікна
alert( document.documentElement.clientWidth ); // ширина минус прокрутка
```

Зазвичай нам потрібна саме доступна ширина вікна, наприклад, щоб намалювати що-небудь, тобто за вирахуванням смуги прокрутки. Тому використовуємо documentElement.clientWidth.

**Ширина / висота сторінки з урахуванням прокрутки**

Теоретично, видима частина сторінки - це documentElement.clientWidth/Height, а повний розмір з урахуванням прокрутки - за аналогією, documentElement.scrollWidth/scrollHeight.

Це вірно для звичайних елементів.

А ось для сторінки з цими властивостями виникає проблема, коли прокрутка тобто, то немає . У цьому випадку вони працюють некоректно. У браузерах Chrome / Safari і Opera при відсутності прокрутки значення documentElement.scrollHeightв цьому випадку може бути навіть менше, ніж documentElement.clientHeight, що, звичайно ж, виглядає як цілковита нісенітниця і нонсенс.

Ця проблема виникає саме для documentElement, тобто для всієї сторінки.

Надійно визначити розмір сторінки з урахуванням прокрутки можна, взявши максимум з декількох властивостей:

```
var scrollHeight = Math.max(
  document.body.scrollHeight,
document.documentElement.scrollHeight,
  document.body.offsetHeight,
document.documentElement.offsetHeight,
  document.body.clientHeight,
document.documentElement.clientHeight
);
alert( Висота з урахуванням  прокрутки: ' + scrollHeight );
```

**Отримання поточної прокрутки**

У звичайного елемента поточну прокрутку можна отримати в scrollLeft/scrollTop.

Більшість браузерів коректно обробить запит до documentElement.scrollLeft/Top, проте Safari / Chrome / Opera є помилки (наприклад 157855 , 106133 ), через які слід використовувати document.body.

Щоб взагалі обійти проблему, можна використовувати спеціальні властивості window.pageXOffset/pageYOffset:

```
alert( 'Поточна прокрутка зверху: ' + window.pageYOffset );
alert( Поточна прокрутка зліва: ' + window.pageXOffset );
```

**Зміна прокрутки: scrollTo, scrollBy, scrollIntoView**

важливо:

Щоб прокрутити сторінку за допомогою JavaScript, її DOM повинен бути повністю завантажений.

На звичайних елементах властивості scrollTop/scrollLeft можна змінювати, і при цьому елемент буде прокручуватися.

Ніхто не заважає точно так же надходити і з сторінкою. У всіх браузерах, крім Chrome / Safari / Opera можна здійснити прокрутку установкою document.documentElement.scrollTop, а в зазначених - використовувати для цього document.body.scrollTop. І буде працювати. Можна спробувати прокручувати і так і сяк і перевіряти, подіяла чи прокрутка, буде кросбраузерну.

Але є й інша, просте і універсальне рішення - спеціальні методи пересування між об'єктами window.scrollBy (x, y) і window.scrollTo (pageX, pageY) .

Метод scrollBy(x,y)прокручує сторінку щодо поточних координат.

Наприклад, кнопка нижче прокрутить сторінку на 10px вниз:

```
window.scrollBy (0,10)
```

Метод scrollTo(pageX,pageY)прокручує сторінку до вказаними координатами щодо документа.

Він еквівалентний установці властивостей scrollLeft/scrollTop.

Щоб прокрутити в початок документа, досить вказати координати (0,0).

```
window.scrollTo (0,0)
```

**scrollIntoView**

Для повноти картини розглянемо також метод elem.scrollIntoView (top) .

Метод elem.scrollIntoView(top)викликається на елементі і прокручує сторінку так, щоб елемент виявився вгорі, якщо параметр тордорівнює true, і внизу, якщо тордорівнює false. Причому, якщо параметр торне вказано, то він вважається рівним true.

Кнопка нижче прокрутить сторінку так, щоб кнопка виявилася вгорі:

```
this.scrollIntoView ()
```

А наступна кнопка прокрутить сторінку так, щоб кнопка виявилася внизу:

```
this.scrollIntoView (false)
```

**Заборона прокрутки**

Іноді буває потрібно тимчасово зробити документ «непрокручіваемим». Наприклад, при показі великої діалогового вікна над документом - щоб відвідувач міг прокручувати це вікно, але не документ.

Щоб заборонити прокрутку сторінки, досить поставити document.body.style.overflow = "hidden".

При цьому сторінка замре в поточному положенні.

Спробуйте самі:

```
document.body.style.overflow = 'hidden'
document.body.style.overflow = ''
```

При натисканні на верхню кнопку сторінка замре на поточному положенні прокрутки. Після натискання на нижню - прокрутка відновиться.

Замість document.body може бути будь-який елемент, прокрутку якого необхідно заборонити.

Недоліком цього способу є те, що сама смуга прокрутки зникає. Якщо вона займала певну ширину, то тепер ця ширина звільниться, і вміст сторінки розшириться, текст «стрибне», зайнявши місце, що звільнилося.

Це може бути не дуже красиво, але легко обходиться, якщо обчислити розмір прокрутки і додати такий же за розміром padding.

**ОТЖЕ**

розміри:

Для отримання розмірів видимої частини вікна: document.documentElement.clientWidth/Height

Для отримання розмірів сторінки з урахуванням прокрутки:

```
var scrollHeight = Math.max( document.body.scrollHeight,
document.documentElement.scrollHeight,
document.body.offsetHeight,
document.documentElement.offsetHeight,
document.body.clientHeight,
document.documentElement.clientHeight);
```

**Прокрутка вікна:**

Прокручування вікна можна отримати як window.pageYOffset(для горизонтальної - window.pageXOffset) всюди, крім IE8-. Про всяк випадок - ось самий крос-браузерні спосіб, що враховує IE7- в тому числі:

```
var html = document.documentElement;
var body = document.body;
var scrollTop = html.scrollTop || body && body.scrollTop ||
0;
scrollTop -= html.clientTop; // в IE7- <html> Зміщено
відностно (0,0)
alert( "Поточна прокрутка: " + scrollTop );
```

Встановити прокрутку можна за допомогою спеціальних методів:

window.scrollTo(pageX,pageY) - абсолютні координати,

window.scrollBy(x,y) - прокрутити щодо поточного місця.

elem.scrollIntoView(top)- прокрутити, щоб елемент elem стало видно.

## *ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ*

**Завдання 1. Висота вікна**

Дана кнопка. При натисканні на цю кнопку виведіть висоту вікна браузера.

**Завдання 2. Прокручування**

Дана кнопка. При натисканні на цю кнопку перейдіть вікно браузера до самого низу.

**Завдання 3. Наявність прокрутки**

Дана кнопка. При натисканні на цю кнопку дізнайтеся, чи є у вікна браузера вертикальна прокрутка.

**Завдання 4. Відкрити нове вікно**

Створи HTML-сторінку з кнопкою **«Відкрити сайт»**. При натисканні на неї за допомогою JavaScript відкривається нове вікно з сайтом https://example.com, розмірами **600x400 пікселів**.

Підказка: використовуй window.open(url, name, features).

**Завдання 5. Закрити вікно через певний час**

Після відкриття нового вікна (як у завданні 1), автоматично **закрити його через 5 секунд**.

Підказка: setTimeout(() => { newWindow.close(); }, 5000);

**Завдання 6. Отримати розміри поточного вікна**

Виведи в консоль **ширину** та **висоту** поточного вікна користувача при завантаженні сторінки.

Підказка: window.innerWidth та window.innerHeight.

**Завдання 7. Перемістити вікно в певну позицію**

Після натискання кнопки перемістити вікно браузера в позицію **(100, 100)** на екрані.

Підказка: window.moveTo(x, y). (Працює лише для вікон, відкритих через window.open, не завжди доступно в браузерах через безпеку.)

**Завдання 8. Змінити розмір вікна**

Напиши скрипт, який змінює розмір поточного вікна до **800x600** пікселів після натискання кнопки.

Підказка: window.resizeTo(width, height) (Також працює лише для нових вікон, не вкладок.)

**Завдання 9. Відстеження зміни розміру вікна**

Додай обробник події resize, щоб у консоль виводилось: **«Розмір вікна змінено: ширина - XXX, висота - YYY»** кожного разу, коли змінюється розмір вікна.

Підказка:

```
window.addEventListener('resize', () => {
  console.log(`Ширина: ${window.innerWidth}, Висота: ${window.innerHeight}`);
});
```

**Завдання 10. Прокрутка сторінки вниз**

Зроби сторінку з великим вмістом. Додай кнопку **«Прокрутити вниз»**, яка переміщує вікно на **300 пікселів вниз**.

Підказка: window.scrollBy(0, 300);

**Завдання 11. Відстеження прокрутки**

Під час прокрутки сторінки виводь у консоль позицію по осі Y (scrollY).

Підказка:

```
window.addEventListener('scroll', () => {
  console.log(window.scrollY);
});
```

**Завдання 12. Діалогові вікна**

1. Попроси ім’я користувача через prompt().
2. Запропонуй підтвердити введення через confirm().
3. Виведи вітання через alert().

Підказка:

```
let name = prompt("Введіть ваше ім’я:");
if (confirm(`Ви ввели: ${name}. Продовжити?`)) {
  alert(`Привіт, ${name}!`);
}
```

**Завдання 13. Передача даних між вікнами**

**Завдання:**

1. Відкрий нове вікно.
2. Передай у нього дані через window.open() (наприклад, query-параметр або через window.opener).
3. У новому вікні оброби ці дані й виведи повідомлення.

Підказка:

- Передача:

```
let newWin = window.open('child.html?msg=Привіт');
```

- Отримання:

```
let params = new URLSearchParams(window.location.search);
alert(params.get("msg"));
```

.