

## Лабораторна робота № 5

**Тема:** Зміна властивостей елемента

**Мета:** навчитися проводити зміну властивостей елемента

### ТЕОРЕТИЧНІ ВІДОМОСТІ

#### className і classList

Зміна класу є одним з найбільш часто використовуваних дій в скриптах.

Колись давно в JavaScript існувало обмеження: зарезервоване слово типу "class" не могло бути властивістю об'єкта. Це обмеження зараз відсутня, але в той час було неможливо мати властивість elem.class.

Тому для класів було введено схоже властивість "className": elem.className відповідає атрибуту "class".  
наприклад:

```
<body class="main page">

  <script>

    alert(document.body.className); // main page

  </script>

</body>
```

Якщо ми присвоюємо щось elem.className, то це замінює всю рядок з класами. Іноді це те, що нам потрібно, але часто ми хочемо додати / видалити один клас.

Для цього є інша властивість: elem.classList.

elem.classList - це спеціальний об'єкт з методами для додавання / видалення одного класу.

наприклад:

```
<body class="main page">

  <script>

    // додавання класу

    document.body.classList.add('article');

    alert(document.body.className); // main page article

  </script>

</body>
```

Так що ми можемо працювати як з рядком повного класу, використовуючи className, так і з окремими класами, використовуючи classList. Вибираємо той варіант, який нам зручніше.

методи `classList`:

`elem.classList.add/remove("class")` - додати / видалити клас.

`elem.classList.toggle("class")` - додати клас, якщо його немає, інакше видалити.

`elem.classList.contains("class")`- перевірка наявності класу, повертає `true/false`.

Крім того, `classList` перебирати, тому можна перерахувати всі класи за допомогою `for..of`:

```
<body class="main page">
  <script>
    for (let name of document.body.classList) {
      alert(name); // main, потім page
    }
  </script>
</body>
```

## Element style

Властивість `elem.style`- це об'єкт, який відповідає тому, що написано в атрибуті `"style"`. Установка стилю `elem.style.width="100px"` працює так само, як наявність в атрибуті `style` рядки `width:100px`.

Для властивості з декількох слів використовується `camelCase`:

`background-color` => `elem.style.backgroundColor`

`z-index` => `elem.style.zIndex`

`border-left-width` => `elem.style.borderLeftWidth`

наприклад:

```
document.body.style.backgroundColor = prompt('background color?', 'green');
```

## Скидання стилів

Іноді нам потрібно додати властивість стилю, а потім, пізніше, прибрати його.

Наприклад, щоб приховати елемент, ми можемо поставити `elem.style.display = "none"`.

Потім ми можемо видалити властивість `style.display`, щоб повернутися до первісного стану. Замість `delete elem.style.display` ми повинні привласнити йому порожню

Якщо ми встановимо в `style.display` порожній рядок, то браузер застосує CSS-класи і вбудовані стилі, як якщо б такого властивості `style.display` взагалі не було.

## ***ЗАВДАННЯ ДЛЯ САМОСТІЙНОЇ РОБОТИ***

### ***РОБОТА З CSS***

#### **Завдання 1 . Робота з CSS**

Запрограмуйте поведінку кнопки після натискання на неї: вона змінює колір тексту в `input`.

#### **Завдання 2 . Блокування полів введення**

Запрограмуйте поведінку кнопок після натискання:  
на них одна з них блокує `input` за допомогою атрибута `disabled`, а інша розблокує.

#### **Завдання 3. Зміна картинки кнопкою**

Запрограмуйте поведінку кнопок після натискання: змінюється одна картинка на іншу. Підібрати картинки однакового розміру

#### **Завдання 4. Зміна картинки наведення на миші та при виводі вказівника миші**

Запрограмуйте таку поведінку картини : при наведенні вказівника миші змінюється перша картинка на другу. При виводі вказівника миші друга змінюється на першу

Підібрати картини однакового розміру

Застосування *this* та *parseInt()*

#### **Завдання 5. Підрахунок кількості кліків на кнопці**

Запрограмуйте поведінку кнопки після натискання на неї:

Проводиться підрахунок кліків. Значення виводиться у саму кнопку.