

Задание 1. Создать форму, где будет помещен выпадающий список ссылок, которым можно осуществить переход, нажав на кнопку «перейти».

Список «SELECT» необходимо заполнить динамически.

Динамическое заполнение списка

Кроме работы с готовыми списками JavaScript может заполнять список динамически. Для записи нового элемента списка используется конструктор **Option** с четырьмя параметрами, первый из которых задает текст, отображаемый в списке, второй - значение элемента списка, соответствующее значению атрибута **VALUE**, третий соответствует свойству **defaultSelected**, четвертый - свойству **selected**.

Пример html

```
<FORM NAME="Sel">
  <!-- Пустой список ссылок-->
  <SELECT NAME="ListOfLinks">
  </SELECT>
  <!-- Кнопка активизации выбранной ссылки-->
  <INPUT TYPE="button" VALUE="Переход"
    onClick="window.location.href=document.links[Sel.ListOfLinks.selectedIndex];">
</FORM>

<A HREF="https://ru.wikipedia.org"></A>
<A HREF="https://vk.com/"></A>
<A HREF="http://edumaterial.ru/"></A>
```

Пример js

```
<script><!--
  // Цикл по всем ссылкам
  for(i=0; i<document.links.length; i++)
  {   // Создание i-элемента списка и запись в него ссылки
      document.Sel.ListOfLinks.options[i] = new Option(document.links[i], i, false, false);
  }
  // Выделение первого элемента в списке
  document.Sel.ListOfLinks.selectedIndex = 0;
  <!-->
</script>
```

Результат

https://ru.wikipedia.org/ ▼ Переход

Задание 2. Создайте форму, на которой пользователю необходимо ввести свои данные (имя, фамилию, возраст, номер телефона) и две кнопки. «Готово» - данные из формы отображаются в «сплывашке» alert. «Сбросить» - сброс формы

Пример html

```
<!-- Кнопки готовности и сброса -->
<INPUT TYPE="button" VALUE="Готово" onClick="Complete();" >
<INPUT TYPE="reset" VALUE="Сброс">
```

Пример js

```
function Complete()
{
    var Elem="Фамилия: " + document.Sell.Family.value +
        "\nИмя: " + document.Sell.Name.value +
        "\nВозраст: " + document.Sell.Age.value +
        "\nТелефон: " + document.Sell.Phone.value;
    alert(Elem);
}
```

Результат

The screenshot shows a web browser window with a toolbar containing icons for Yandex, VK, Mail, and others. The main content area displays a form titled "Заполните анкету" (Fill out the survey). The form has four input fields: "Фамилия:" (empty), "Имя:" (filled with "ПУК"), "Возраст:" (filled with "18"), and "Телефон:" (filled with "124"). Below the fields are two buttons: "Готово" (Ready) and "Сброс" (Reset). An alert dialog box is open on the right side of the browser, titled "Подтвердите действие" (Confirm action). It displays the collected data: "Фамилия:", "Имя: ПУК", "Возраст: 18", and "Телефон: 124". There is an "OK" button at the bottom right of the dialog.

Задание 3. При нажатии кнопки «Готово» сделайте проверку, на то что заполнены все поля формы. Если хотя бы одно из полей не заполнено, выведите сообщение при помощи alert для пользователя с сообщением о необходимости заполнить все поля.

Задание 4. Создайте форму, где пользователю необходимо ввести логин и пароль при его регистрации.

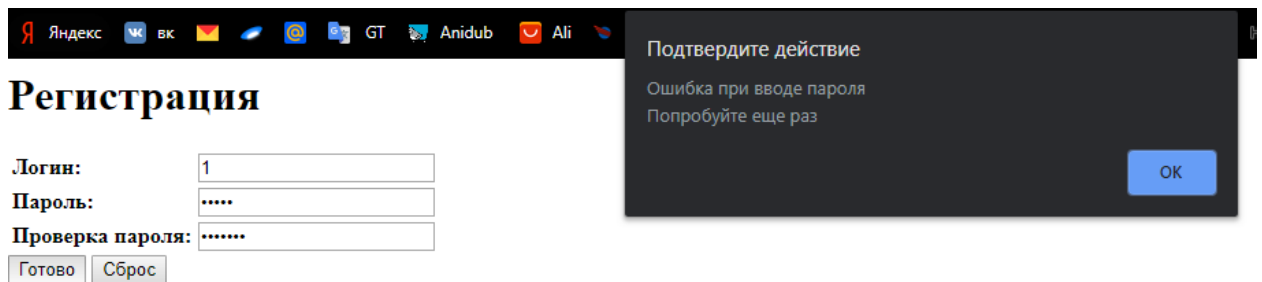
Результат

Регистрация

The screenshot shows a registration form with three input fields: "Логин:" (filled with "1"), "Пароль:" (filled with "*****"), and "Проверка пароля:" (filled with "*****"). Below the fields are two buttons: "Готово" (Ready) and "Сброс" (Reset).

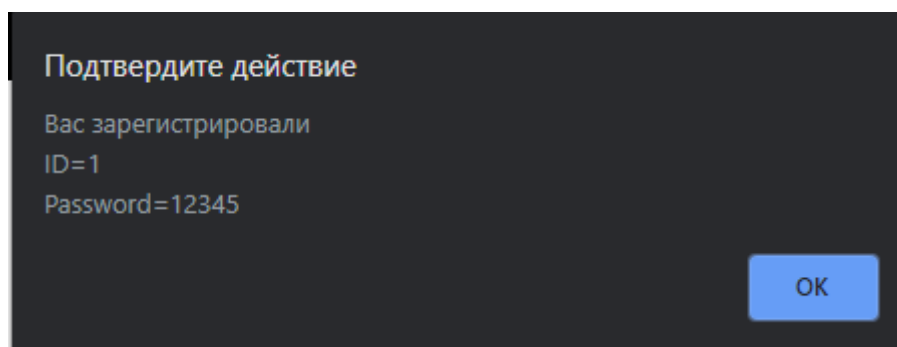
Необходимо предусмотреть проверку пароля на соответствие. Если пароль в области «Пароль» и «Проверка пароля» не совпадают, то с помощью alert вывести сообщение об ошибке.

Результат



The screenshot shows a web browser window with a registration form titled "Регистрация". The form has three input fields: "Логин:" with the value "1", "Пароль:" with masked characters ".....", and "Проверка пароля:" with masked characters ".....". Below the fields are two buttons: "Готово" and "Сброс". An alert dialog box is open on the right side of the browser window. The dialog has a title "Подтвердите действие" and contains the text "Ошибка при вводе пароля" and "Попробуйте еще раз". There is an "OK" button at the bottom right of the dialog.

Если же пароли совпадают с помощью alert вывести следующее сообщение:



The screenshot shows a confirmation alert dialog box. The dialog has a title "Подтвердите действие" and contains the text "Вас зарегистрировали", "ID=1", and "Password=12345". There is an "OK" button at the bottom right of the dialog.

При нажатии на кнопку «Сброс» форма очищается.

- *focus()*. Передает полю фокус ввода.
- *blur()*. Отбирает у поля фокус ввода.
- *select()*. Выделяет содержимое поля.

Пример js

```
<script><!--
function Complete1()
{
    if(document.Sel3.Pwd.value==document.Sel3.Pwd1.value)
        alert("Вас зарегистрировали\nID="+document.Sel3.Id.value+"\nPassword="+document.Sel3.Pwd.value);
    else
        alert("Ошибка при вводе пароля\nПопробуйте еще раз");
}
//-->
</script>
```

Пример html

```
<INPUT TYPE="button" VALUE="Готово" onClick="Complete1();">
<INPUT TYPE="reset" VALUE="Сброс">
```

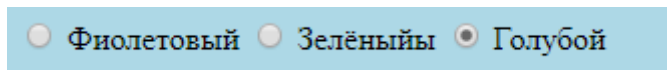
В отличие от большинства элементов документа HTML, поля форм могут получать фокус ввода клавиатуры. При щелчке или выборе их другим способом они становятся активными, т.е. главными приёмниками клавиатурного ввода.

Если в документе есть текстовое поле, то набираемый текст появится в нём, только если поле имеет фокус ввода. Другие поля по-разному реагируют на клавиатуру. К примеру,

пытается перейти на вариант, содержащий текст, который вводит пользователь, а также отвечает на нажатия стрелок, передвигая выбор варианта вверх и вниз. Управлять фокусом из JavaScript можно методами `focus` и `blur`. Первый перемещает фокус на элемент DOM, из которого он вызван, а второй убирает фокус. Значение `document.activeElement` соответствует текущему элементу, получившему фокус.

Задание 5. Создайте форму с тремя полями `RadioButton`. При переключении между ними будет происходить смена фона страницы.

Результат



Пример html

```
<input type="radio" name="color" value="mediumpurple"> Фиолетовый
<input type="radio" name="color" value="lightgreen"> Зелёный
<input type="radio" name="color" value="lightblue"> Голубой
```

Пример js

```
<script>
  var buttons = document.getElementsByName("color");
  function setColor(event) {
    document.body.style.background = event.target.value;
  }
  for (var i = 0; i < buttons.length; i++)
    buttons[i].addEventListener("change", setColor);
</script>
```

Метод `document.getElementsByName` выдаёт все элементы с заданным атрибутом `name`. Пример перебирает их (посредством обычного цикла `for`, а не `forEach`, потому что возвращаемая коллекция – не настоящий массив) и регистрирует обработчик событий для каждого элемента. Помните, что у объектов событий есть свойство `target`, относящееся к элементу, который запустил событие. Это полезно для создания обработчиков событий – наш обработчик может быть вызван разными элементами, и у него должен быть способ получить доступ к текущему элементу, который его вызвал.