Лабораторная работа 10

Тема: Работа с формой HTML

Задание 1. Сделать регистрационную форму с вводом пароля и полем для проверки введенного пароля.

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html;charset=utf-8">

</head>

<body>

<H1>Регистрация</H1>

<FORM NAME="Sel3">

<TABLE>

<TR><TD><B>Идентификатор:<B></TD>

<TD><INPUT NAME="Id" SIZE=20

onBlur="this.value=this.value.toUpperCase()"></TD></TR>

<TR><TD><B>Пароль:<B></TD>

<TD><INPUT TYPE="password" NAME="Pwd" SIZE=20

onFocus="this.select();"></TD></TR>

<TR><TD><B>Проверка пароля:<B></TD>

<TD><INPUT TYPE="password" NAME="Pwd1" SIZE=20

onFocus="this.select();"></TD></TR>

</TABLE>

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

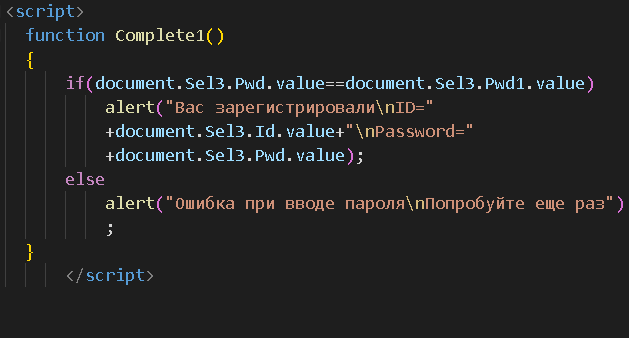
<INPUT TYPE="reset" VALUE="Сброс">

</FORM>

</body>

</html>

Создать скрипт для проверки введённого пароля. В случае успешной операции вывести на экран окно о регистрации. Если пароли не совпадают вывести окно об ошибке.

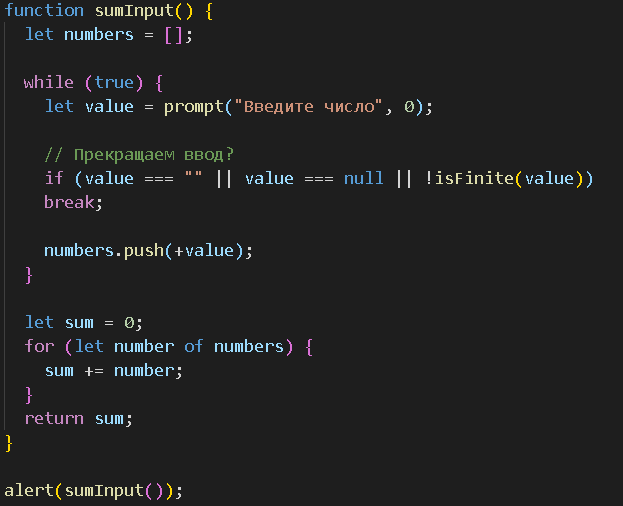


Задача 2. Напишите функцию sumInput(), которая:

* Просит пользователя ввести значения, используя prompt и сохраняет их в массив.
* Заканчивает запрашивать значения, когда пользователь введёт не числовое значение, пустую строку или нажмёт «Отмена».
* Подсчитывает и возвращает сумму элементов массива.

P.S. Ноль 0 – считается числом, не останавливайте ввод значений при вводе «0».

Обратите внимание на малозаметную, но важную деталь решения. Мы не преобразуем value в число сразу после prompt, потому что после value = +value мы не сможем отличить пустую строку (конец записи) от «0» (разрешённое число). Мы сделаем это позже.



Задача 3. Напишите функцию ucFirst(str), возвращающую строку str с заглавным первым символом. Например:

Мы не можем просто заменить первый символ, так как строки в JavaScript неизменяемы.

Но можно пересоздать строку на основе существующей, с заглавным первым символом:

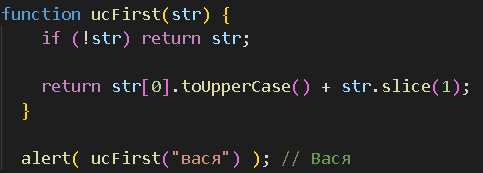
let newStr = str[0].toUpperCase() + str.slice(1);

Однако есть небольшая проблемка. Если строка пуста, str[0] вернёт undefined, а у undefined нет метода toUpperCase(), поэтому мы получим ошибку.

Выхода два:

1. Использовать str.charAt(0), поскольку этот метод всегда возвращает строку (для пустой строки — пустую).
2. Добавить проверку на пустую строку.

Вот второй вариант:



Задача 4. Из заданного массива вывести список имен на экран пользователей проживающих в городе McKenziehaven.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Всякий раз когда вам надо отфильтровать массив в JavaScript — встроенный метод для отфильтровки вашего массива это хороший выбор. filter() даёт вам сделать callback для каждого элемента и вернуть отфильтрованный массив.

Главная разница между forEach() и filter() в том, что forEach() просто пускает цикл по массиву и выполняет callback, в то время как filter() выполняет callback и проверяет возвращаемое значение. Если значение верное, то оно будет положено в новый массив, если же нет, то оно будет из него исключено.

Заданный массив:

let arr = [

{

"id": 1,

"name": "Leanne Graham",

"username": "Bret",

"email": "Sincere@april.biz",

"address": {

"street": "Kulas Light",

"suite": "Apt. 556",

"city": "Gwenborough",

"zipcode": "92998-3874",

"geo": {

"lat": "-37.3159",

"lng": "81.1496"

}

},

"phone": "1-770-736-8031 x56442",

"website": "hildegard.org",

"company": {

"name": "Romaguera-Crona",

"catchPhrase": "Multi-layered client-server neural-net",

"bs": "harness real-time e-markets"

}

},

{

"id": 2,

"name": "Ervin Howell",

"username": "Antonette",

"email": "Shanna@melissa.tv",

"address": {

"street": "Victor Plains",

"suite": "Suite 879",

"city": "Wisokyburgh",

"zipcode": "90566-7771",

"geo": {

"lat": "-43.9509",

"lng": "-34.4618"

}

},

"phone": "010-692-6593 x09125",

"website": "anastasia.net",

"company": {

"name": "Deckow-Crist",

"catchPhrase": "Proactive didactic contingency",

"bs": "synergize scalable supply-chains"

}

},

{

"id": 3,

"name": "Clementine Bauch",

"username": "Samantha",

"email": "Nathan@yesenia.net",

"address": {

"street": "Douglas Extension",

"suite": "Suite 847",

"city": "McKenziehaven",

"zipcode": "59590-4157",

"geo": {

"lat": "-68.6102",

"lng": "-47.0653"

}

},

"phone": "1-463-123-4447",

"website": "ramiro.info",

"company": {

"name": "Romaguera-Jacobson",

"catchPhrase": "Face to face bifurcated interface",

"bs": "e-enable strategic applications"

}

},

{

"id": 4,

"name": "Patricia Lebsack",

"username": "Karianne",

"email": "Julianne.OConner@kory.org",

"address": {

"street": "Hoeger Mall",

"suite": "Apt. 692",

"city": "South Elvis",

"zipcode": "53919-4257",

"geo": {

"lat": "29.4572",

"lng": "-164.2990"

}

},

"phone": "493-170-9623 x156",

"website": "kale.biz",

"company": {

"name": "Robel-Corkery",

"catchPhrase": "Multi-tiered zero tolerance productivity",

"bs": "transition cutting-edge web services"

}

},

{

"id": 5,

"name": "Chelsey Dietrich",

"username": "Kamren",

"email": "Lucio\_Hettinger@annie.ca",

"address": {

"street": "Skiles Walks",

"suite": "Suite 351",

"city": "Roscoeview",

"zipcode": "33263",

"geo": {

"lat": "-31.8129",

"lng": "62.5342"

}

},

"phone": "(254)954-1289",

"website": "demarco.info",

"company": {

"name": "Keebler LLC",

"catchPhrase": "User-centric fault-tolerant solution",

"bs": "revolutionize end-to-end systems"

}

},

{

"id": 6,

"name": "Mrs. Dennis Schulist",

"username": "Leopoldo\_Corkery",

"email": "Karley\_Dach@jasper.info",

"address": {

"street": "Norberto Crossing",

"suite": "Apt. 950",

"city": "South Christy",

"zipcode": "23505-1337",

"geo": {

"lat": "-71.4197",

"lng": "71.7478"

}

},

"phone": "1-477-935-8478 x6430",

"website": "ola.org",

"company": {

"name": "Considine-Lockman",

"catchPhrase": "Synchronised bottom-line interface",

"bs": "e-enable innovative applications"

}

},

{

"id": 7,

"name": "Kurtis Weissnat",

"username": "Elwyn.Skiles",

"email": "Telly.Hoeger@billy.biz",

"address": {

"street": "Rex Trail",

"suite": "Suite 280",

"city": "Howemouth",

"zipcode": "58804-1099",

"geo": {

"lat": "24.8918",

"lng": "21.8984"

}

},

"phone": "210.067.6132",

"website": "elvis.io",

"company": {

"name": "Johns Group",

"catchPhrase": "Configurable multimedia task-force",

"bs": "generate enterprise e-tailers"

}

},

{

"id": 8,

"name": "Nicholas Runolfsdottir V",

"username": "Maxime\_Nienow",

"email": "Sherwood@rosamond.me",

"address": {

"street": "Ellsworth Summit",

"suite": "Suite 729",

"city": "Aliyaview",

"zipcode": "45169",

"geo": {

"lat": "-14.3990",

"lng": "-120.7677"

}

},

"phone": "586.493.6943 x140",

"website": "jacynthe.com",

"company": {

"name": "Abernathy Group",

"catchPhrase": "Implemented secondary concept",

"bs": "e-enable extensible e-tailers"

}

},

{

"id": 9,

"name": "Glenna Reichert",

"username": "Delphine",

"email": "Chaim\_McDermott@dana.io",

"address": {

"street": "Dayna Park",

"suite": "Suite 449",

"city": "Bartholomebury",

"zipcode": "76495-3109",

"geo": {

"lat": "24.6463",

"lng": "-168.8889"

}

},

"phone": "(775)976-6794 x41206",

"website": "conrad.com",

"company": {

"name": "Yost and Sons",

"catchPhrase": "Switchable contextually-based project",

"bs": "aggregate real-time technologies"

}

},

{

"id": 10,

"name": "Clementina DuBuque",

"username": "Moriah.Stanton",

"email": "Rey.Padberg@karina.biz",

"address": {

"street": "Kattie Turnpike",

"suite": "Suite 198",

"city": "Lebsackbury",

"zipcode": "31428-2261",

"geo": {

"lat": "-38.2386",

"lng": "57.2232"

}

},

"phone": "024-648-3804",

"website": "ambrose.net",

"company": {

"name": "Hoeger LLC",

"catchPhrase": "Centralized empowering task-force",

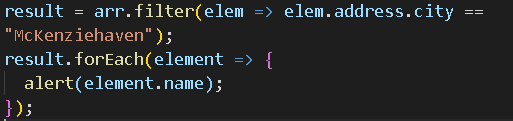
"bs": "target end-to-end models"

}

}

]

Пример программы для вывода пользователей проживающих в городе McKenziehaven

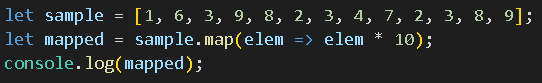


Самостоятельно вывести все элементы массива в виде таблицы в окне браузера.

Задача 5. Задан массив чисел [1,6,3,9,8,2,3,4,7,2,3,8,9]. Необходимо умножить каждый элемент массива на 10, не меняя исходный.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Метод map() как и filter() с foreach() берет callback и пускает его по всем элементам массива, но то, что делает его уникальным — это то что он генерирует новый массив, основываясь на существующем массиве.



Самостоятельно вывести все элементы массива на экран браузера.