

Введение в основы JavaScript

Junior Frontend Developer

Цель урока:

Вы научитесь работать с самыми базовыми вещами, на которых строится язык программирования JavaScript. Например, математические выражения, логические конструкции (if else, switch case), циклы.

Содержание урока:

1. Математика в JavaScript: <https://learn.javascript.ru/operators>
2. Операторы сравнения в JS: > < >= <= == ===
3. if else, switch case, ? :
4. Логические операторы: || (ИЛИ), && (И) и ! (НЕ), ??
5. Циклы. for, while, do while

Дополнительные материалы:

1. Оператор объединения с null: <https://learn.javascript.ru/nullish-coalescing-operator>
2. Циклы while и for: <https://learn.javascript.ru/while-for>
3. Условное ветвление if else, ? : : <https://learn.javascript.ru/ifelse>
4. Логические операторы: <https://learn.javascript.ru/logical-operators>
5. Операторы сравнения в JS: <https://learn.javascript.ru/comparison>

Задание #1

Форма авторизации и регистрации - очень популярная и неотъемлемая часть многих веб-сайтов. Сейчас вы разработаете ее на JavaScript.

Представим, что на нашем сайте есть пользователь, у которого логин - это "the_best_user", а пароль - "12345678". Сохраните эти значения в переменные и назовите их existedUserLogin и existedUserPassword соответственно.

Вам требуется с помощью функции prompt вывести 2 модальных окна на экран. 1-е будет с надписью "Введите логин", 2-е - "Введите пароль". Сохраните введенные пользователем значения в переменные userLogin и userPassword. Также учтите, что в итоговых данных пользователя не должно быть лишних пробелов в начале и в конце строки. Используйте метод trim, чтобы убрать лишние пробелы.

Если введенные пользователем логин и пароль совпадают с уже существующими данными в existedUserLogin и existedUserPassword, то выведите в модальном окне при помощи alert сообщение "Добро пожаловать, userLogin!" (замените userLogin на логин пользователя). Иначе же отобразите сообщение "Логин и (или) Пароль введены неверно!".

Примечание: никогда в разработке веб-приложений не используйте нестрогое сравнение “==”. Оно является инициатором большого количества багов, так как производит преобразование типов. Используйте исключительно строгое сравнение “===”. Оно не преобразует типы и уменьшает шанс возникновения ошибок. В последующих заданиях также используйте строгое сравнение “===”.

Задание #2

Вы когда нибудь проходили тесты? Думаю ваш ответ - да, так как они повсюду. Сейчас почти все тесты были перенесены в онлайн формат и размещаются на веб-сайтах, которые разрабатывают веб-разработчики. Сейчас вы создадите небольшой тест по математике для учеников начальных классов.

В тесте всего 5 вопросов:

1. Сколько будет $2 + 2$?
2. Сколько будет $2 * 2$?
3. У Пети было 5 яблок. 3 из них он съел, 1 отдал другу. Сколько яблок у Пети осталось?
4. У Маши было 10 конфет. 2 она съела, 1 отдала другу. После мама дала Маше еще 5 конфет. Сколько в итоге конфет осталось у Маши?
5. Сколько будет $2 + 2 * 2$?

Вам необходимо помочь ученикам начальных классов, поэтому сделаем для них тест, который проверит их знания математики.

Каждый вопрос и заранее подготовленный правильный ответ (ответы должны быть типом данных number) сохраняйте в переменные. Вопросы выводите с помощью функции prompt и также сохраняйте значение введенные пользователем в отдельные переменные.

Если ответ был дан правильно, то отобразите сообщение “Ответ Верный” при помощи alert, иначе - “Ответ Неверный”. Если даже ответ был дан неправильно, то пользователь все равно должен быть перенаправлен на следующий вопрос.

Также вам необходимо вести счет правильных и неправильных ответов. Сохраняйте эти данные в переменные **correctAnswers** и **incorrectAnswers** соответственно. После выполнения теста с помощью alert выведите сообщение “Конец теста! Правильные ответы - correctAnswers; Неправильные ответы - incorrectAnswers.” (замените correctAnswers и incorrectAnswers на итоговые значения).

Примечание: учтите, что prompt возвращает тип данных string, а значения, которые должны храниться в переменных с ответами - это тип данных number.

Задание #3

Сейчас вы поработаете с новой [функцией confirm](#). Она выводит модальное окно с 2-мя кнопками: “OK” и “Cancel”. Данная функция идеально подходит для того, чтобы задавать вопросы пользователю. Давайте этим и займемся.

Для начала создайте новую папку под названием “confirm”. В эту папку добавьте 2 файла: index.html и index.js. Сделайте базовую разметку HTML и подключите index.js файл в index.html.

Сейчас ваша задача состоит в том, чтобы задавать вопросы с помощью confirm и выводить ответы через alert. То есть сначала идет вопрос, а затем после нажатия одной из кнопок (“OK” либо “Cancel”) выводится ответ.

Синтаксис функции confirm выглядит следующим образом:

```
confirm('Ваш текст');
```

Используйте данные вопросы: JavaScript появился в 1995 году? Спецификация JavaScript называется ECMAScript? JavaScript был создан за 1 месяц?

В случае, если ответ на вопрос утвердителен, то выводите в alert строку “Верно”, иначе выведите правильный ответ.

Откройте index.html файл в браузере и посмотрите на ваш результат.

Задание #4

Данный код 3 раза запрашивает имена новых студентов и, если имя студента было введено, то его приветствуют сообщением “Добро пожаловать, newStudent!” (newStudent - имя студента).

Вам необходимо переписать данный цикл for на цикл с while и do while, чтобы логика выполнения кода и результат остались без изменения.

```
for (let i = 0; i < 3; i += 1) {  
    let newStudent = prompt('Введите имя нового студента!');  
    newStudent = newStudent.trim();  
    if (newStudent) {  
        alert(`Добро пожаловать, ${newStudent}!`);  
    }  
}
```

Задание #5

Продолжим работать с циклами.

Сейчас ваша задача состоит в следующем. Вам дано число 100. Необходимо посчитать сумму от 0 до 100 (либо наоборот, от 100 до 0). В итоге у вас должно получиться значение 5050. Сохраните итоговое значение в переменную sum и выведите его на экран при помощи alert.

Задание #6 (дополнительное)

Представим, что мы разрабатываем сайт для ресторана, в котором действует программа скидок. Ресторан подсчитывает сумму оплаты каждого клиента и поощряет самых частых посетителей.

На данный момент, у ресторана есть клиент по имени Игорь, который за все время потратил 110\$ в данном заведении. Сохраните эти данные в переменные **clientName** и **clientSpentForAllTime**.

Программа скидок работает следующим образом:

1. Если клиент потратил от 100\$ до 300\$, то скидка 10%;
2. Если клиент потратил от 300\$ до 500\$, то скидка 20%;
3. Если клиент потратил от 500\$, то скидка 30%;

Клиент по имени Игорь решил посетить еще раз ресторан. Как мы видим, ему полагается скидка в 10%, так как он потратил 110\$. Сейчас Игорь закупается в заведении на 25\$ и ему необходимо предоставить скидку. Сохраните значение 25 в переменную **clientSpentToday**.

Для начала, реализуйте логику предоставления скидок. Используйте **if else** и логические операторы. Итоговую скидку сохраните в переменную **discount**. Выведите в модальном окне через **alert** текст “Вам предоставляется скидка в **discount**%!” (**discount** замените на процент скидки).

После этого необходимо выставить клиенту по имени Игорь счет на оплату (учитывая скидку). Для начала обновите переменную **clientSpentForAllTime**. Затем выведите в модальном окне через **alert** сообщение “Спасибо, **clientName**! К оплате **clientSpentToday**\$”. За все время в нашем ресторане вы потратили **clientSpentForAllTime**\$.” (замените **clientName**, **clientSpentToday** и **clientSpentForAllTime** на соответствующие значения переменных).

Задание #7 (дополнительное)

Поздравляем! Вы разработали логику предоставления скидки клиентам ресторана. Вы оптимизировали большую задачу, потому что раньше скидки считались вручную.

Вы работали уже с заранее определенными данными **clientName**, **clientSpentToday** и **clientSpentForAllTime**. Но клиентов у ресторана много и каждый покупает блюда по разной стоимости и каждому необходимо предоставлять разные скидки. Поэтому сейчас мы будем запрашивать пользователя ввести нужные данные в текстовое поле.

Измените код из задания 6

Воспользуемся уже известной нам функцией prompt. В каждую из следующих переменных **clientName**, **clientSpentToday** и **clientSpentForAllTime** присвойте значение, которое введет пользователь в текстовое поле. Итоговые данные **clientSpentToday** и **clientSpentForAllTime** должны быть типом данных **number**.

Для **clientName** сообщение в **prompt** должно быть “Введите имя клиента”, **clientSpentToday** - “Сколько клиент потратил сегодня?”, **clientSpentForAllTime** - “Сколько клиент потратил за все время?”.

Если пользователь ввел неверные данные для clientSpentToday и clientSpentForAllTime (например, "hello"), то выведите в модальном окне текст "Сумма, которую клиент потратил за все время и которую потратил сегодня, должна быть числом! Перезагрузите страницу, чтобы повторить попытку.". При неверных данных не позволяйте пользователю идти дальше к подсчету скидки. Используйте if else, чтобы реализовать данную логику

Задание #8 (дополнительное)

Когда мы проходим регистрацию, то в большинстве случаев у нас запрашивают пароль. И всегда для него есть некоторые ограничения по длине либо по символам. Сейчас вы напишете программу, которая будет проверять на валидность пароль, введенный пользователем.

Для начала вам необходимо запросить пароль у пользователя при помощи prompt. Отобразите сообщение "Введите пароль".

Пароль должен удовлетворять следующим условиям:

1. Длина пароля должна быть минимум 3 символа и не больше 20 символов (включая 20).
2. Пароль должен содержать минимум 1 прописную (большую) букву.
3. Пароль должен содержать минимум 1 цифру.

Если все условия удовлетворены, то выведите при помощи alert сообщение "Пароль валидный. Добро пожаловать в аккаунт!", иначе - "Пароль не удовлетворяет условиям! Перезагрузите страницу и попробуйте ввести его еще раз."

Посмотрите на несколько возможных результатов:

1. "1234f" - не удовлетворяет условиям
2. "123456" - не удовлетворяет условиям
3. "1234F" - удовлетворяет условиям
4. "12" - не удовлетворяет условиям
5. "JavaScript" - не удовлетворяет условиям
6. "JavaScript123" - удовлетворяет условиям