Формы

Раман Крот

Адукар

Повестка дня

1. Навигация по форме
2. Обработка форм
3. Валидация формы
4. Регулярные выражения

Навигация по форме

Навигация по форме осуществляется по клику на tab или мышью.
Ниже представлены различные методы и свойства:

document.forms.formName - получение формы

form.elements - псевдомассив со всеми элементами формы

elem.form - получение формы из элемента

input.value / textarea.value / select.value - получение/запись значения

Навигация по форме

input.checked - значение “выбранности” checkbox’a

option.selected - значение “выбранности” option’a

new Option(text, value, defaultSelected, selected) - создание нового значения option

focus - событие получения фокуса на элементе (не всплывает)

blur - событие исчезания фокуса на элементе (не всплывает)

Обработка формы

change - событие происходящее после изменения поля input или по изменению select и checkbox

input - срабатывает на изменении input моментально

cut, copy, paste - события вырезания, копирования и вставки

Отправка формы

При отправки формы возникает событие **submit**. Отправка происходит после клика на input с типом submit или по нажатию на Enter.

Обычно используется для валидации формы. Если форма валидна вызывается метод **form.submit()**. Если же нет - обработка прекращается вызовом **event.preventDefault()**

Валидация формы

Штатными возможностями мы можем только установить значение required, которое будет проверять “заполненность” или же “выбранность” элемента. Для всех остальных проверок используется следующий алгоритм:

- вешается обработчик на событие submit
- выполнение отправки прерывается и проводится проверка полей и элементов на соответствие установленному паттерну
- в случае прохождения проверки данные отправляются, в противном случае выполнение скрипта прекращается

Практика

1. Создайте селект с несколькими опциями, добавьте одну опцию используя Javascript, сделайте так, чтобы по выбору опции выводилось сообщение с данными этой опции
2. Создайте форму вычисления процентов по вкладу:

Регулярные выражения

Присутствуют в большинстве языков программирования и образуют собой шаблон описывающий строку. Используются для поиска и замены в строке.

```
/^(([^<>()\[\]\.,;:\s@\"]+(\.[^<>()\[\]\.,;:\s@\"]+)*)|(\".+\"))@(([^<>()[\]\.,;:\s@\"]+\.)+[^<>()[\]\.,;:\s@\"]{2,})$/i
```

Паттерны и флаги

Есть два варианта создания регэкспа:

new RegExp(“шаблон”, “флаги”) - через создание нового объекта RegExp, где шаблон - код, способ описания строки, а флаги служебные значения поиска

/шаблон/флаги - более простая и более часто используемая форма

Паттерны и флаги

i - нечувствительность к регистру

g - ищет все совпадения, иначе - первое

m - многострочный режим

Методы регулярок

str.search(reg) - возвращает позицию первого совпадения или -1

str.match(reg) - если не установлен флаг g, возвращает первое совпадение, обёрнутое в массив со свойствами index (позиция совпадения) и input (исходная строка), если в шаблоне встречаются скобки, то содержимое скобок будет еще одним элементом массива. Если флаг g установлен, то возвращается просто массив из всех совпадений. Если совпадений не найдено - null

regexp.test(str) - проверяет наличие совпадений, возвращает true/false

Классы символов

. - любой символ, кроме перевода строки

\ - экранирование символа

\d - любая цифра

\D - не цифра

\s - любой пробельный символ

\S - не пробельный символ

Классы символов

\w - любой текстовый символ латинского алфавита или нижнее подчеркивание “_”

\W - не текстовый символ латинского алфавита или нижнее подчеркивание “_”

\b - граница слова

\B - не граница слова

Квантификаторы

{n} - энное количество элементов (\d{2} - два числа)

{n, n} - диапазон элементов

+ - один или более (тоже самое что и {1,})

? - ноль или один (тоже, что и {0,1})

***** - ноль или более (эквивалентно {0,})

Наборы и диапазоны

[aoe] - ищет в наборе из символов a, o и e

[a-z] - диапазон от a до z

[^] - исключающий диапазон

Скобочные группы

В регулярных выражениях существуют скобочные группы, которые определяются знаками ( ).

Они позволяют более точно настроить поиск по строке (значение ?: удаляет группу из результатов поиска, как отдельный элемент), а также применять квантификаторы ко всей группе.

Все содержимое образует собой группу. Группы могут быть вложенными

Начало и конец строки

^ - ищет совпадения только в начале строки

$ - ищет только в конце строки

Практика

1. Создайте регулярное выражение для поиска трёх точек.
2. Создайте regexp, который ищет все положительные числа, в том числе десятичные.
3. Создайте регулярку, которая ищет цвета в формате #eee, #eeeddd
4. Предложите строку, которая подойдет под выражение ^$
5. Создайте HTML-форму регистрации с email и паролем. По клику провести валидацию пароля и email, где пароль должен содержать хотя бы одну цифру, один спецсимвол и одну букву, а так же быть длиннее 6 знаков. При прохождении валидации выводить приветственное сообщение, в противном случае - ошибку.

Внеклассное чтение

https://learn.javascript.ru/forms-controls

https://habr.com/en/post/242695/

https://learn.javascript.ru/regular-expressions

https://regexr.com