Чит-лист по Web UI контролам

**Содержание**

* 1 Контролы
  + 1.1 Textbox
  + 1.2 Text area
  + 1.3 Password field
  + 1.4 Email
  + 1.5 Search field
  + 1.6 Числовое поле ввода
  + 1.7 Ввод времени
  + 1.8 Ввод даты
  + 1.9 Интервал (дат или времени)
  + 1.10 Файлы
* 2 Возможные значения для типов данных
  + 2.1 Тексты
  + 2.2 Действия (Cut, Copy, Paste, etc.)
* 3 Security
  + 3.1 Прочее

Контролы

Textbox

* #Тексты (кроме раздела перенос строки)
* #Security
* Enter срабатывает, как Submit

Text area

* #Тексты
* #Security
* Enter обрабатывается, как перенос строки

Password field

* Большая часть за обработку отображения звездочек зависит от браузера (устройства)
* Русские пароли
* Длина пароля
* Разнообразие символов в пароле
* Совпадение 2-х паролей в форме регистрации
* #Тексты

Email

!!!TBD!!!

Search field

Уточнить язык поисковых запросов?

* И ИЛИ НЕ
* Строгое соответствие
* Поиск по подстроке (в результатах поиска должны быть документы, начинающиеся с или заканчивающиеся тем, что указал пользователь при поиске)
* Какие символы игнорируются?
* Wildcard и прочие regexp'ы (.\*, \*., ?, \*)
* Разделители слов
* #Тексты
* Ищется ли по цифрам. Являются ли цифры разделителями слов (не должны являться)
* Ищутся ли слова с различными "сложными" символами вроде '+' (например, c++)
* Ищется ли по словоформам (например, "систему" и "система" - поиск может считать за одно слово)
* Поиск по нескольким словам, разбросанным по записи (то есть не идущим последовательно друг за другом)
* Поиск с учетом порядка слов ( "Вася сказал" и "сказал Вася" - результат должен быть одинаковый)
* Вести текст в другой кодировке (например, Óèêèïåäèÿ å ìíîãîåçè÷íà åëåêòðîííà)
* Е и Ё
* Вместо " может искаться quot

Числовое поле ввода

* Корректное значение
* Граничные значения
* Значения выходящие за заданный диапазон
* Пустое значение
* Не должно принимать буквы и спецсимволы
* 0
* Отрицательные числа
* Дробные числа с разным разделителем (0.0001 или 0,0001)
* Числа с запятыми или пробелами (1,234,567 или 1 000 000)
* Европейский стиль (1.234.567,89)
* #Продвинутые числа

Ввод времени

Популярные виды контролов

* HH:MM:SS
* HH:MM
* AM PM или 24h

Лучшая практика для реализации форм времени – минимум проверок и валидаций, максимум подсказок (хинтинга)

* Корректное время (12:00:04)
* 00:00:00
* Пусто
* Пусто после редактирования
* Указано только одно из полей (например, указаны только часы 00:MM:SS)
* Отрицательное время -12
* Буквы
* Пробелы
* Copy-Paste некоторый текст
* Больше или меньше цифр, чем нужно
* Разное время между машинами
* Различные временные зоны
* Сброс времени вперед или назад

Ввод даты

* Високосный день
* Некорректная дата (30 февраля)
* 29 февраля в не високосном году
* Различные форматы (Июнь 5, 2001; 60/05/2001; 06/05/01; 05-05-01; 6/5/2001 12:34)
* Перевод на летнее и зимнее время
* Перевод даты вперед-назад

Интервал (дат или времени)

* Иногда оба календаря должны разрешать ставить даты только ДО сегодняшнего дня включительно, а иногда даты могут быть любыми относительно сегодняшнего дня
* Иногда требуется валидация на то, что дата «с» раньше, чем дата «по»:

- если пользователь сперва указал дату «с», то даты для «по» должны ограничиваться датой «с» снизу (включительно) - если пользователь сперва указал дату «по», то даты для «с» должны ограничиваться дату «по» снизу (включительно) - если пользователь несколько раз редактирует даты, то они должны работать адекватно! Каждый из контролов в отдельности должен обрабатываться, как отдельный контрол для ввода даты или времени.

* Другие популярные проверки для интервалов времени и дат:

- A < B

- A = B

- A > B

- Только А

- Только B

- Ни А, ни B

*Здесь А - это начальное время, B - конечное время интервала.*

Файлы

* Длинное имя файла (> 255 знаков)
* Название с русскими буквами
* Специальные символы в имени (пробел, \* ? / \ | < > , . ( ) [ ] { } ; : ' " ! @ # $ % ^ &)
* Не существующий
* Уже существующий
* Нет свободного места на диске
* Доступность хранилища куда грузим файл
* Минимальное место на диске
* Защищенное на запись
* Запись на несуществующий диск
* Запись на заблокированный диск
* Запись на удаленную машину
* Запись на испорченный диск или флешку

Возможные значения для типов данных

Тексты

* Длинные строки (255, 256, 257, 1000, 1024, 2000, 2048 или более знаков)
* Символы со знаками ударения (àáâãäåçèéêëìíîðñòôõöö, etc.)
* Общие разделители и специальные символы (" ' ` | / \ , ; : & < > ^ \* ? Tab « »)
* «Достаточно плохое» строковое значение: "[|]'~<!--@/\*$%^&#\*/()?>,.\*/\
* "Плохие символы" (♣ ☺ ♂)
* Пустая строка
* Пустая строка после редактирования
* Пробелы
* Одиночный пробел
* Только пробелы
* Строка, начинающаяся и заканчивающаяся с пробелов
* Чаще всего лишние пробелы будут удаляться (стрипяться)
* Конец строки (^M, \n, \r)

Действия (Cut, Copy, Paste, etc.)

* Перенос строки
* Один перенос строки
* Только переносы строк
* Начало и окончание текста с переноса строки
* Чувствительность к регистру символов
* Иероглифы
* Продвинутые числа
* Степени двойки

- 32768 (2^15)

- 32769 (2^15 + 1)

- 65536 (2^16)

- 65537 (2^16 + 1)

- 2147483648 (2^31)

- 2147483649 (2^31 + 1)

- 4294967296 (2^32)

- 4294967297 (2^32 + 1)

* Научная запись чисел (1Е-16)
* Вычисляемые выражения (2+3)
* В шестнадцатеричной системе счисления (F0)

Security

* SQL иньекции

- Кавычка: '

- ' OR '1'='1'

- ' or '1'='1' -- '

- ' or '1'='1' ({ '

- ' or '1'='1' /\* '

- !!!Найти!!!

* XSS

<http://ha.ckers.org/xss.html>  
- <script>alert("xss!")</script>

- <script>document.getElementByID("...").disabled=true</script>

- <input onclick="javascript:alert('xss');">

- <b onmouseover="alert('xss!')">Hello</b>

* HTML инъекции

- </body> или </body></body>

- <textarea />

- <input></input>

- <form action="http://live.hh.ru"><input type="submit"></form>

- !!!Найти!!!

* Корректный SQL запрос

- DROP TABLE user;

- SELECT \* FROM blog WHERE code LIKE 'a%';

- Корректный Ruby код, Сфинкс

- !!!Найти!!!

Прочее

* Комментарий <!--
* "${code}";
* '';!--"<CSS\_Check>=&{()}