

Элементы ввода, регулярные выражения и обработка данных

JS
COURSE
ORT DNIPRO

ORT**DNIPRO**.ORG/**JS**

1. Регулярные выражения

Регулярные выражения

Короткие записи популярных символьных классов

Короткая запись	Полная запись	Описание
\s	[\f\t\n\r]	Пробельный символ
\S	[^\f\t\n\r]	Любой символ, кроме пробельного
\d	[0-9]	Цифра
\D	[^0-9]	Любой символ, кроме цифры
\w	[a-zA-Z0-9_]	Латиница или цифра
\W	[^a-zA-Z0-9_]	Любой символ, кроме латиницы или цифры
.	[^\n\r]	Любой символ, кроме перевода строки
\b		Граница слова
\B		Не граница слова
\A		Начало строки
\Z		Конец строки

Регулярные выражения – язык шаблонов которыми проверяется нахождение в строке искомых элементов.

https://ru.wikibooks.org/wiki/Регулярные_выражения
https://uk.wikipedia.org/wiki/Регулярний_вираз

Тестер регулярных выражения

Тестер регулярных выражений

Введите Регулярное выражение

...

Введите проверяемый текст

Hello

TRUE

При помощи него мы можем протестировать регулярное выражение перед использованием.

В репозитории занятия: [./src/regexp-tester](#)

2. Работа с регулярными выражениями в JavaScript

Регулярные выражения в JavaScript

`RegExp.test(string)`

`String.match(RegExp)`

`String.search(RegExp)`

`String.replace(RegExp, Function)`

*Методы строк и методы объекта RegExp
которые нам могут пригодиться.*

Регулярные выражения в JavaScript

```
2
3   let str = 'HTML & CSS & JavaScript is a frontend techonology.';
4   let re = /javascript/i;
5
6   let result = re.test();
7   console.log('test():', result);
8
9   //-----//
10
11  result = str.search(re);
12  console.log('search():', result);
13
14  //-----//
15
16  result = str.match(re);
17  console.log('match():', result);
18
19  //-----//
20
21  result = str.replace(re, 'JS');
22  console.log('replace():', result);
23
```

3. Немного практики #1

Работы с регулярными выражениями в JavaScript

```
1  <script>
2
3  let data = 'Lorem ipsum dolor sit amet, AE4574CM consectetur adipiscing elit. Integer varius dictum turpis sit
4  amet dictum. Cras aliquet quis dui vitae suscipit. Mauris felis lorem, TE3589EP dapibus ut libero id, consequat
5  malesuada nunc. Etiam purus tellus, pharetra eu AC45876KH consectetur at, tristique non leo. Maecenas egestas
   eros ut lacus posuere viverra. Etiam vel ultrices urna. Vestibulum tempus, ligula AH7893Ab eget hendrerit luctus,
   est nisi BB5686TB dignissim nibh, in euismod ligula mauris eu lorem. Etiam eget pulvinar nisi, ac semper massa.
   Phasellus ultrices venenatis BB1200CA eleifend. Cras tincidunt suscipit elit a blandit. Nunc AA7744 sodales nisl
   quis eros dignissim condimentum.';
6
7  </script>
```

Воспользуйтесь шаблоном в репозитории
занятия [./src/regexp-template](#)

4. DOM и валидация данных

Валидация данных

(проверка на корректность)

Валидация данных сводится к проверке соответствуют ли введённые данные шаблону. Что считать корректным, а что нет – определяет разработчик.

JavaScript, формы и элементы ввода

*Для JavaScript'а элементы ввода это текстовые поля. Получить (или установить) информацию из них можно обратившись к свойству **value** элемента.*

*Но нам не обязательно проверять значение, мы можем «спросить» браузер считает ли он значение валидным, обратившись к методу **.checkValidity()** элемента ввода.*

Свойство .validity элементов ввода

Validity Properties

The **validity property** of an input element contains a number of properties related to the validity of data:

Property	Description
customError	Set to true, if a custom validity message is set.
patternMismatch	Set to true, if an element's value does not match its pattern attribute.
rangeOverflow	Set to true, if an element's value is greater than its max attribute.
rangeUnderflow	Set to true, if an element's value is less than its min attribute.
stepMismatch	Set to true, if an element's value is invalid per its step attribute.
tooLong	Set to true, if an element's value exceeds its maxLength attribute.
typeMismatch	Set to true, if an element's value is invalid per its type attribute.
valueMissing	Set to true, if an element (with a required attribute) has no value.
valid	Set to true, if an element's value is valid.

Свойство **.validity** доступное у элементов ввода позволяет получить детальную информацию и проблемах при валидации данных в элементе ввода.

Подробнее: https://www.w3schools.com/js/js_validation_api.asp

На следующем занятии...

На следующем занятии...

Концепция

Single Page Application

Домашнее задание
/сделать

Домашнее задание #К.1

```
1 <script>
2
3   let data = "Pellentesque ex eros, +380665433321 porttitor eu bibendum ac, aliquam tincidunt urna. Mauris
4   tristique lobortis orci, nec varius magna convallis interdum. Etiam 5363 4567 8765 3454 pharetra tempor ex, vel
5   eleifend (067) 678 44 21 odio lacinia (0562) 35-30-38 eget. Morbi maximus libero vitae aliquet facilisis. Vivamus
6   5674-2346-8945-0012 vitae quam nisi. Quisque 12/45 quis venenatis 5192722517688913 lacus. Sed ac lorem (050)
7   567-45-33 nec leo pharetra 4556796335044346 dapibus sed eu +38067432112 ex. In hac 4913-8185-2881-4543 habitasse
8   platea dictumst. In dignissim 5461158320267908 suscipit rutrum. Ut 4916849417542904 luctus sapien in risus 56th
9   street auctor, ac placerat 067-678-44-21 quam malesuada. Pellentesque (056) 7783322 bibendum justo 5363 4567
10  87653 3455 tempus purus convallis, a viverra nunc ullamcorper. Nulla 5213 9203 2475 5355 eget lectus gravida,
11  porta eros vitae, semper erat +39-926-1234567. Aenean volutpat vehicula dui ut pharetra.";
12
13  /*
14  Необходимо выбрать !!валидные!! номера банковских карт и вывести их в формате 4 блока по 4 цифры.
15
16  Например:
17
18  5475 0330 3843 6453
19  4539 1799 4927 2557
20  ...
21  */
```

Задача: Выбрать из строки корректные, **валидные**, номера банковских карт и вывести их в формате 4 блока по 4 цифры.

Воспользуйтесь шаблоном в репозитории занятия: [./src/homework-template](#)