

# Регулярные выражения и валидация данных



JavaScript  
Courses

[www.courses.dp.ua](http://www.courses.dp.ua)

# 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

*Регулярное выражение – язык шаблонов которыми проверяется нахождение в строке искомых элементов.*

[https://ru.wikibooks.org/wiki/Регулярные\\_выражения](https://ru.wikibooks.org/wiki/Регулярные_выражения)

[https://uk.wikipedia.org/wiki/Регулярний\\_вираз](https://uk.wikipedia.org/wiki/Регулярний_вираз)

# Тестер регулярных выражения

## Тестер регулярных выражений

Введите Регулярное выражение

Введите проверяемый текст

TRUE

*При помощи него мы можем протестировать регулярное выражение перед использованием.*

*Воспользуйтесь файлом*

***[./source/regexp.html](#)***

# Валидация данных

(проверка на корректность)

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

# 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.*

<https://learn.javascript.ru/regexp-methods>

# 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
```

<https://learn.javascript.ru/regexp-methods>



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  </script>
```

Воспользуйтесь заготовкой

[./source/example\\_01](#)

# Валидация средствами JavaScript

# Проверка вводимых данных (валидация)

Проверка вводимых данных сводится к:

- 1. Заполнено ли поле (выбран ли вариант, для полей выбора)?*
- 2. Соответствуют ли введённые данные шаблону?*

## 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.*

# Алгоритм валидации с JavaScript:

1. Подписываемся на событие клик (***onclick***) кнопки запускающей отправку данных;
2. В обработчике событие проверяем **данные** на соответствие (***.checkValidity()*** для формы);
3. Если данные не корректны сообщаем об это пользователю и **отменяет отправку данных** (Для этого достаточно вызвать метод ***.preventDefault()*** у объекта с информацией о событии);
4. Если данные корректны, запускаем процесс отправки данных вызвав метод ***.submit()*** для формы.

Немного практики #2

# Валидация данных

## Форма регистрации

ФИО	Email	
<input type="text" value="Ваше ФИО"/>	<input type="text" value="Ваш email"/>	
Телефон	ИНН	Номер паспорта
<input type="text" value="Ваш телефон"/>	<input type="text" value="Ваш ИНН"/>	<input type="text" value="Номер вашего паспорта"/>
<input type="button" value="Зарегистрировать"/>		

Воспользуемся заготовкой: [./source/example\\_02](#)

Реализуем вариацию данных в форме



Немного практики #3

# Регулярные выражения на практике

	SSD 240/256 ГБ, SATA	1132 грн.
Корпус и БП	Middle Tower ATX, 450 Вт	1415 грн.
	Сумма для ПК на платформе Intel	18253.5 грн.
	Сумма для ПК на платформе AMD	16980 грн.

При первой же возможности мы вернули в базовые игровые конфигурации чипы **Core i3-8100**. Катализатором здесь стал некоторый откат стоимости процессоров Intel на нашем рынке. Нет, они по-прежнему еще стоят очень дорого, но лишь по этой причине отбрасывать возможность сборки ПК на данной платформе – не оправдано. Итак, самые доступные 4-ядерные чипы Coffee Lake сейчас можно купить в Украине за 4811 грн., то есть на 849 грн.–40 дешевле, чем полтора месяца назад. На зарубежных площадках эта модель предлагается за 3679 грн.–140. Если вас не смущают нюансы с гарантийным обслуживанием и даже с учетом стоимости доставки покупка процессора обойдется дешевле, можно рассмотреть такой вариант дополнительной экономии.

Увы, других способов пока нет. Вернее, есть, но они вам вряд ли понравятся. Самый очевидный – покупка процессора семейства **Pentium Gold**. Это изначально будут откровенно слабые чипы для игровой системы. Двух вычислительных ядер уже недостаточно для тяжелых проектов, несмотря на их достаточно высокую эффективность и поддержку технологии многопоточности Hyper-Threading.

Воспользуемся заготовкой: [./source/example\\_03](#)

Реализуем замену всех долларовых цен на гривневые (с пересчётом)

Домашнее задание  
/узнать

# Разберите детально **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

*Регулярное выражение – шаблон которым проверяется строка, строка может соответствовать шаблону, а может не соответствовать.*

[https://ru.wikibooks.org/wiki/Регулярные\\_выражения](https://ru.wikibooks.org/wiki/Регулярные_выражения)

[https://uk.wikipedia.org/wiki/Регулярний\\_вираз](https://uk.wikipedia.org/wiki/Регулярний_вираз)

Домашнее задание  
/сделать

# Домашнее задание #L.1

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

Воспользуйтесь заготовкой: [./homework/hw1](#)

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

# Домашнее задание #L.2



## PhonesShop

iPhone 8	DKK 654
iPhone 7	USD 64
Apple iPhone 6	SEK 465
Samsung Galaxy S9	EUR 456
Nokia 3310	USD 567
HTC Two M3	HUF7999
Lenovo H6000	MXN 199
Microsoft Lumia 950	CHF 567
Sony Ericsson K750 Gold	USD 456

Воспользуйтесь заготовкой:

[\*\*\*./homework/hw2\*\*\*](#)

**Задача:** Пересчитать  
цены на товары в гривну  
в соответствии с  
актуальными курсами  
валют НБУ.