

# Курс веб-разработка

Первые шаги к CMS | PHP и шаблоны



# Что научитесь создавать на курсе

## Часть 1 / База

Промо  
СТРАНИЦА  
А | В  
html + css | анимации

АДАПТИВНЫЕ  
СТРАНИЦЫ  
(мобильники)

ИНТЕРАКТИВНЫЕ  
СТРАНИЦЫ  
(взаимодействие)

Современный  
лендинг

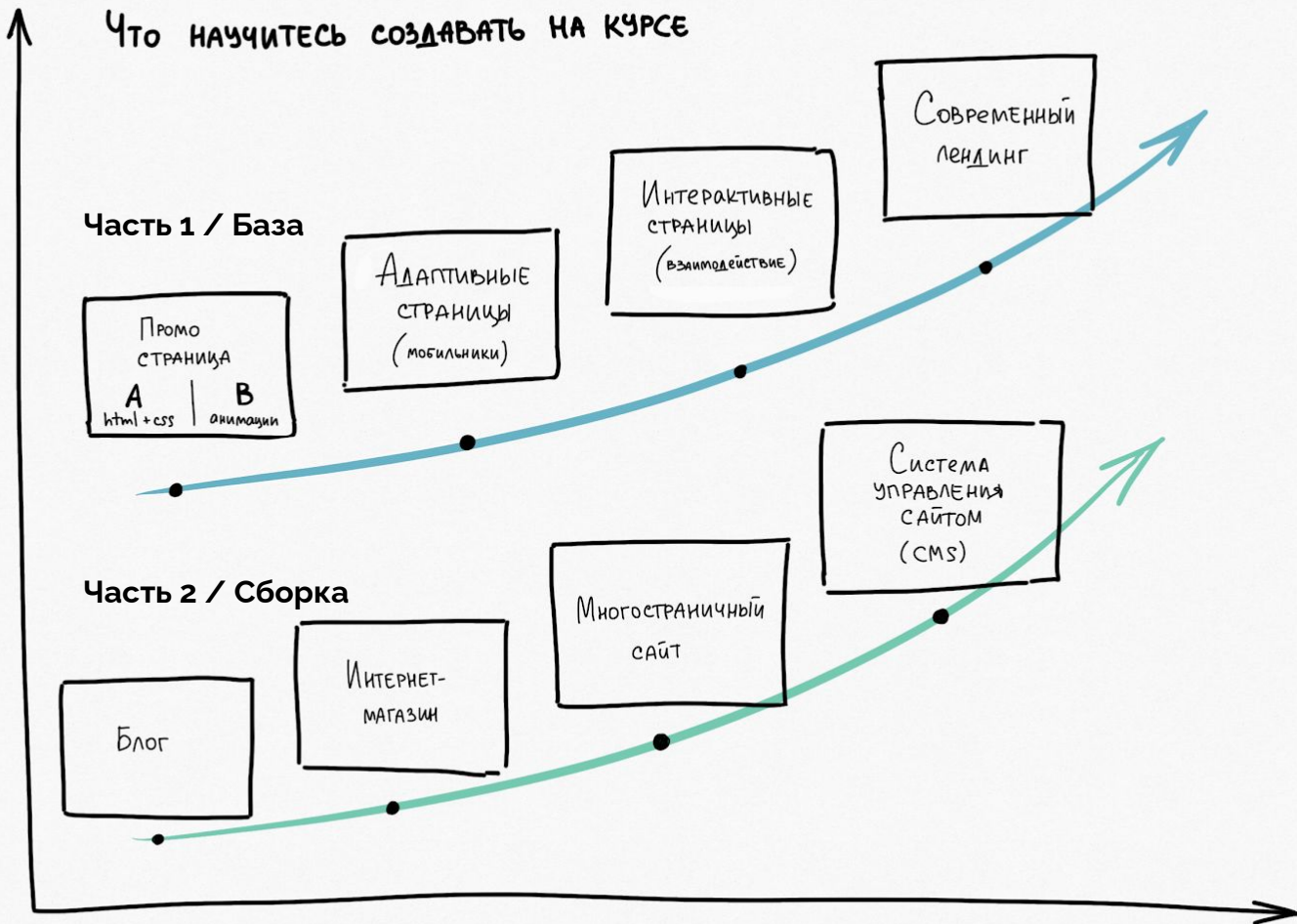
## Часть 2 / Сборка

Блог

Интернет-  
магазин

Многостраничный  
сайт

Система  
управления  
сайтом  
(CMS)



---

## Когда статического сайта бывает недостаточно?

1. Чтобы поменять текст на странице, нам нужно заново заливать ее на сервер.
  2. Добавить новость на сайт сможет только понимающий в верстке человек.
  3. Добавить фотогалерею из 100 фоток - даже страшно представить.
  4. Приходится вручную обрабатывать формы заявок и как-то сохранять данные.
-

...

**Как мы можем упростить  
внесение изменений  
на сайте?**

# CMS - Content Management System

1. Позволяет управлять информацией через административную панель.
2. Автоматизирует такие вещи как добавление фотогалереи или формы на сайте.
3. Чтобы вносить изменения, необходимо иметь доступ в административную панель с помощью логина и пароля.



DokuWiki

concrete5

Drupal



XOOPS  
powered by you



Pligg



Other CMSes

ArticleMS



zikula

modx



Geeklog

AlegroCart  
OPEN SOURCE E-COMMERCE

zenPHOTO  
THE SMALL-SCALE ARTIST'S CHOICE



mambo™  
power in simplicity



Joomla!



osCommerce  
Open Source E-Commerce Solutions

PHPDug



OPENCART  
opensource shopping



PRESTASHOP

phpbb  
creating communities

TYP03



TEXTPATTERN

Plone



NUKE VIEW

powered by

zenCart  
the art of e-commerce



---

## Почему wordpress?

1. Он бесплатный.
  2. Одна из самых популярных CMS, а значит про нее есть куча информации.
  3. Хорошая документация, частично переведена на русский язык (Wordpress Codex)
  4. Много бесплатных шаблонов.
  5. Постоянно обновляется.
-

# Панель администратора

The screenshot shows the WordPress administrator dashboard. At the top, the user is identified as 'Ангелина Буваева' with a profile picture. The dashboard title is 'Консоль'. A sidebar on the left contains navigation links: Главная, Обновления (2), Записи, Медиафайлы, Страницы, Комментарии, Contact Form 7, Внешний вид, Плагины (2), Пользователи, Инструменты, Настройки, SEO, Elementor, and Gutenberg. The main content area features a large announcement card for Gutenberg. The card has a red header 'Консоль' and a 'Настройки экрана' button. The announcement text reads: 'Грядет новый, современный издательский опыт. Направьте ваши слова, мультимедиа и вёрстку в новом направлении с Gutenberg, создаваемым нами, новым редактором для WordPress.' Below the text is a preview of the Gutenberg editor interface. To the right of the preview are two columns of text: 'Попробуйте новый редактор!' and 'Ещё не готовы?'. The first column includes a red button 'Попробуйте Gutenberg' and a link 'Узнайте больше о Gutenberg'. The second column includes a grey button 'Установить Classic Editor'.

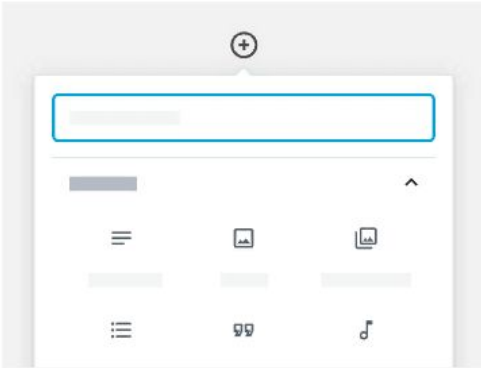
Привет, Angelina Buvaeva

Настройки экрана ▾ Помощь ▾

## Консоль

Грядет новый, современный издательский опыт. ✕ Закрыть

Направьте ваши слова, мультимедиа и вёрстку в новом направлении с Gutenberg, создаваемым нами, новым редактором для WordPress.



### Попробуйте новый редактор!

Вы можете попробовать Gutenberg, установив его как плагин (и если захотите, то воспользоваться обратной связью), до того, как мы официально его выпустим. Вы можете помочь с [тестированием](#), [сообщениями об ошибках](#) или принять участие в разработке в [репозитории GitHub](#).

[Попробуйте Gutenberg](#)


[Узнайте больше о Gutenberg](#)

### Ещё не готовы?

Новый редактор будет включен по умолчанию в следующем значимом выпуске WordPress. Если вы не уверены насколько он совместим с вашей темой и плагинами, то у нас есть запасной план. Установите плагин [Classic Editor](#), чтобы продолжить использовать текущий редактор до тех пор, пока вы не будете готовы перейти на новый.

[Установить Classic Editor](#)

### Elementor на виду

 Elementor v2.2.7 [+ Создать новую страницу](#)

### Быстрый черновик



...

**Что нужно для работы  
Wordpress?**

---

## Что нужно для запуска WP?

1. Сервер (он будет обрабатывать запросы пользователей)
  2. PHP (язык для обработки данных на сервере)
  3. MySQL (база данных, в ней хранятся тексты статей, логины и пароли пользователей и много чего еще)
-

# MySQL - базы данных

1. На сайте много информации: тексты статей, данные пользователей, результаты заполнения форм, комментарии к новостям. Все это надо где-то хранить.
2. MySQL предоставляет удобный способ хранения информации: в таблицах.
3. Wordpress создает и редактирует таблицы автоматически, нам нужно только настроить подключение к MySQL.



# PHP - язык, распространенный в веб-приложениях.

1. Должен быть настроен на сервере.
2. Используется вместе с HTML для создания гибких шаблонов веб-страниц.
3. У языка есть хорошая документация на русском языке:

<http://php.net/manual/ru/>



# Сервер

1. Для разработки на домашнем ПК устанавливается локальный сервер.
2. Настройка локального сервера может занять много времени, особенно у новичков.
3. Вместо этого мы воспользуемся готовым решением:  
<https://laragon.org/download/>

...

# ОСНОВЫ РАБОТЫ С PHP

# ОСНОВЫ PHP

Код на PHP заключен в такую конструкцию:

```
<?php
```

```
    /* Здесь будет код */
```

```
?>
```

# ОСНОВЫ PHP

Для вывода текста на странице  
есть оператор **echo**:

```
<?php
```

```
    echo 'Hello!';
```

```
?>
```



# ОСНОВЫ PHP

Переменные в PHP начинаются со знака **\$**. Если не присвоить им значение сразу, то они будут равны **null**.

```
<?php
```

```
    $greeting = 'Hello there!';
```

```
    echo $greeting;
```

```
?>
```

# ОСНОВЫ PHP

Еще одна полезная функция в PHP - это вывод информации из переменной: **var\_dump**

```
<?php
```

```
    $greeting = 'Hello there!';
```

```
    var_dump($greeting);
```

```
?>
```

...

**Пора сделать что-то  
интересное?**

# ОСНОВЫ PHP

Данные из GET-запросов хранятся в переменной `$_GET`, и мы можем получить их следующим образом:

```
<?php
```

```
    $greeting = $_GET['greeting'];
```

```
    var_dump($greeting);
```

```
?>
```

# ОСНОВЫ PHP

Если параметра нет в GET-запросе, PHP ругается. Чтобы этого не происходило, мы можем сначала *проверить* существование значения, а только потом его *использовать*.

```
<?php
    if (isset($_GET['greeting']))
    {
        $greeting = $_GET['greeting'];
        var_dump($greeting);
    }
?>
```

# ОСНОВЫ PHP

Как вы успели заметить, условие в PHP пишется так же, как и в JavaScript. И кстати, для сложения строк здесь используется не +, а точка:

```
<?php
    if (isset($_GET['greeting']))
    {
        echo 'Hello, ' . $_GET['greeting'] . '!';
    }
else
{
    echo 'Sorry, no greeting!';
}
?>
```

...

**Как это использовать  
на странице?**

Давайте скопируем сайт Selly Hotels в папку с нашим PHP и переименуем **index.html** в **index.php**

Ссылка на сайт:

<https://drive.google.com/open?id=1vdaml18QDgai7j4NlcCK98i-eUGUNMz8>



# PHP и шаблоны

Писать html-код внутри **echo** не очень удобно. Поэтому для условий внутри html-шаблона есть специальные, более удобные конструкции:

```
<?php if (isset($_GET['email'])): ?>
    <form action="/" method="get" class="order_form">
        <input id="emailField" type="text"
            name="email" placeholder="Введите email"
            class="form_field">
        <button class="submit">Подписаться</button>
    </form>
<?php endif; ?>
```

# PHP и шаблоны

Само слово **echo** тоже можно сократить, если мы используем его в шаблоне:

```
<?php if (isset($_GET['email'])): ?>
```

```
    <p class="promo_subtitle">
```

```
        Теперь на почту <?=$_GET['email'] ?> мы будем  
        присылать информацию о наших акциях!
```

```
    </p>
```

```
<?php endif; ?>
```

А теперь попробуем использовать  
переменную прямо внутри HTML-кода!  
И проверим как работает условие **if**.

# Задача №1 | Отзывчивая форма

У тебя новое письмо! Давай посмотрим что там?

[https://drive.google.com/open?id=1y9znh7nElSRKizQXFCs3x07\\_19xQtsCUbg6eLZKpRLI](https://drive.google.com/open?id=1y9znh7nElSRKizQXFCs3x07_19xQtsCUbg6eLZKpRLI)

...

**Что если данных  
очень много?**

# PHP и шаблоны

Данные могут храниться не только в переменных. Множество однотипных данных можно хранить в массиве. Он выглядит так:

```
$fruits = ['apple', 'apricot', 'orange', 'banana'];
```

Количество элементов массива можно посчитать:

```
$fruits = ['apple', 'apricot', 'orange', 'banana'];  
$fruitsNumber = count($fruits);
```

# PHP и шаблоны

Чтобы вывести данные из массива в цикле, мы можем воспользоваться циклом **foreach**:

```
foreach ($fruits as $fruit)
{
    echo $fruit;
}
```

# PHP и шаблоны

Массивы бывают не только простые, но и ассоциативные. Это похоже на объект в Javascript. Мы можем обращаться к значению по ключу. А ключом, например, может быть слово.

```
$user = [  
    'firstName' => 'Ace',  
    'lastName' => 'Ventura',  
    'post' => 'detective',  
    'hobby' => 'animals'  
];  
  
echo $user['firstName'] . ' ' . $user['lastName'];
```



# PHP и шаблоны

Разумеется, вывод цикла в шаблоне тоже можно упростить, чтобы было удобнее. Например, выведем список фруктов в виде списка html:

```
<?php $fruits = ['apple', 'banana', 'orange']; ?>
<ul>
    <?php foreach ($fruits as $fruit): ?>
        <li><?= $fruit ?></li>
    <?php endforeach; ?>
</ul>
```

# PHP и шаблоны

Как насчет массива ассоциативных массивов? Например, набор статей. Для каждой у нас есть заголовок и краткое описание.

```
<?php
    $articles = [
        [
            'title' => '13 причин любить инфографику',
            'text' => 'Инфографика повсюду. Почему она так популярна? Пытаясь ответить на
вопрос, мы перевели ряд научных исследований...',
        ],
        [
            'title' => 'Что такое педагогический дизайн?',
            'text' => 'Педагогический дизайн – относительно новое понятие в современной
системе образования...',
        ]
    ];
?>
```

```
<?php foreach ($articles as $article): ?>
    <div class="wrapper">
        <h2 class="title"><?= $article['title'] ?></h2>
        <p class="text"><?= $article['text'] ?></p>
    </div>
<?php endforeach; ?>
```

# Задача N°2 | Вывод данных из файла

У тебя новое письмо! Давай посмотрим что там?

<https://drive.google.com/open?id=1lJoSakgvTetCqLhyUX-3hqiQwNf7FX9GcDkqtNRIPq8>

...

**Еще не забыли  
про jQuery?**

# PHP и jQuery могут взаимодействовать друг с другом

## Get started with a free account

Find your people. Engage your customers. Build your brand. Do it all with Mailchimp's marketing tools. Already have an account? [Log in](#)

Email

Username

Choose a username that contains only letters and numbers, or use your email address.

Password

 Show

- One lowercase character
- One uppercase character
- One number
- One special character
- 8 characters minimum

Get Started!

By clicking this button, you agree to Mailchimp's [Anti-spam Policy & Terms of Use](#).

Рассмотрим нашу форму на сайте.  
Сейчас она перезагружает страницу.

Но форма может работать лучше!

---

## Что такое AJAX?

1. AJAX - Asynchronous Javascript and XML
  2. Это механизм асинхронного обмена данными с сервером
  3. Асинхронный - это значит что обмен данными происходит в фоне. А в это время пользователь продолжает пользоваться страницей.
  4. Когда приходит ответ от сервера - мы можем его прочесть и обработать.
-

# jQuery ajax()

В jQuery есть специальная функция для ajax-запроса: <http://api.jquery.com/jquery.ajax/>

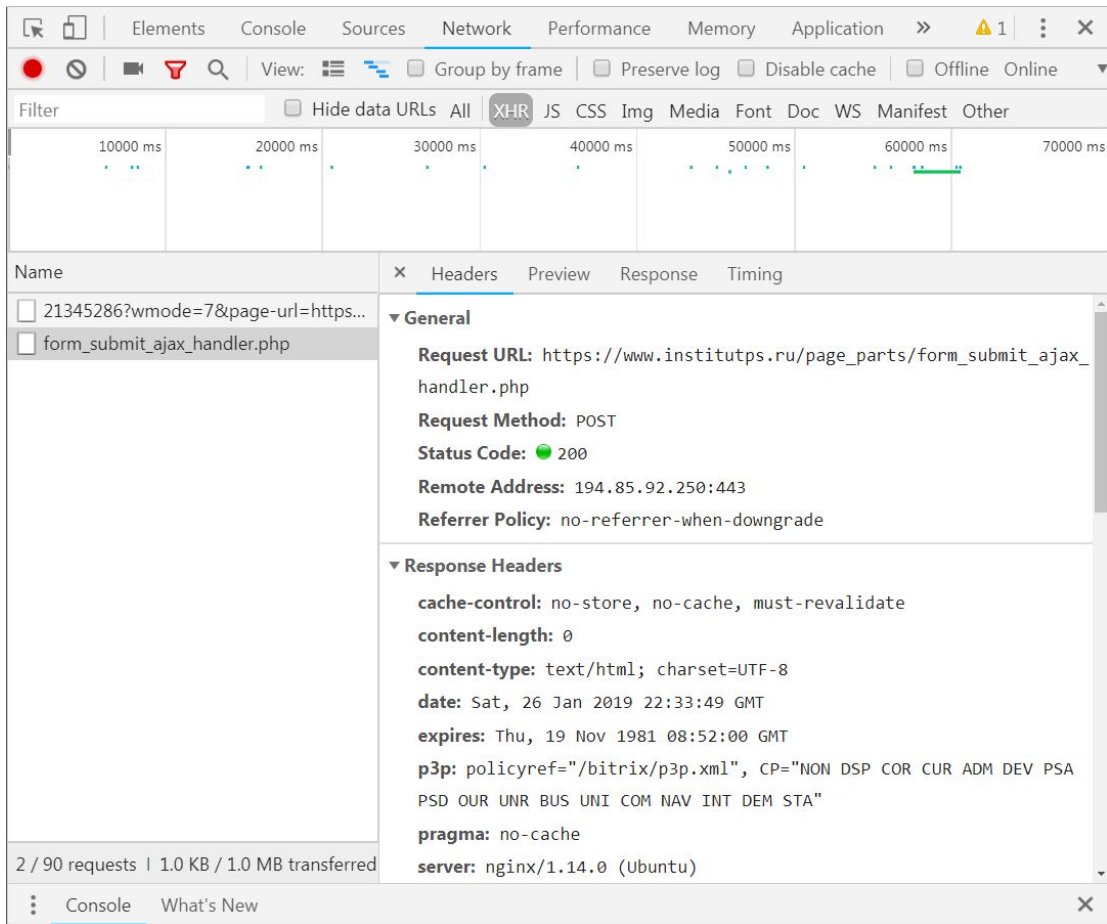
```
$.ajax({
  method: 'POST',
  url: '/php/handler.php',
  data: {
    name: 'John',
    location: 'Boston',
  },
})
.done(function(response) {
  console.log('Response from server: ' + response);
})
```



# jQuery ajax()

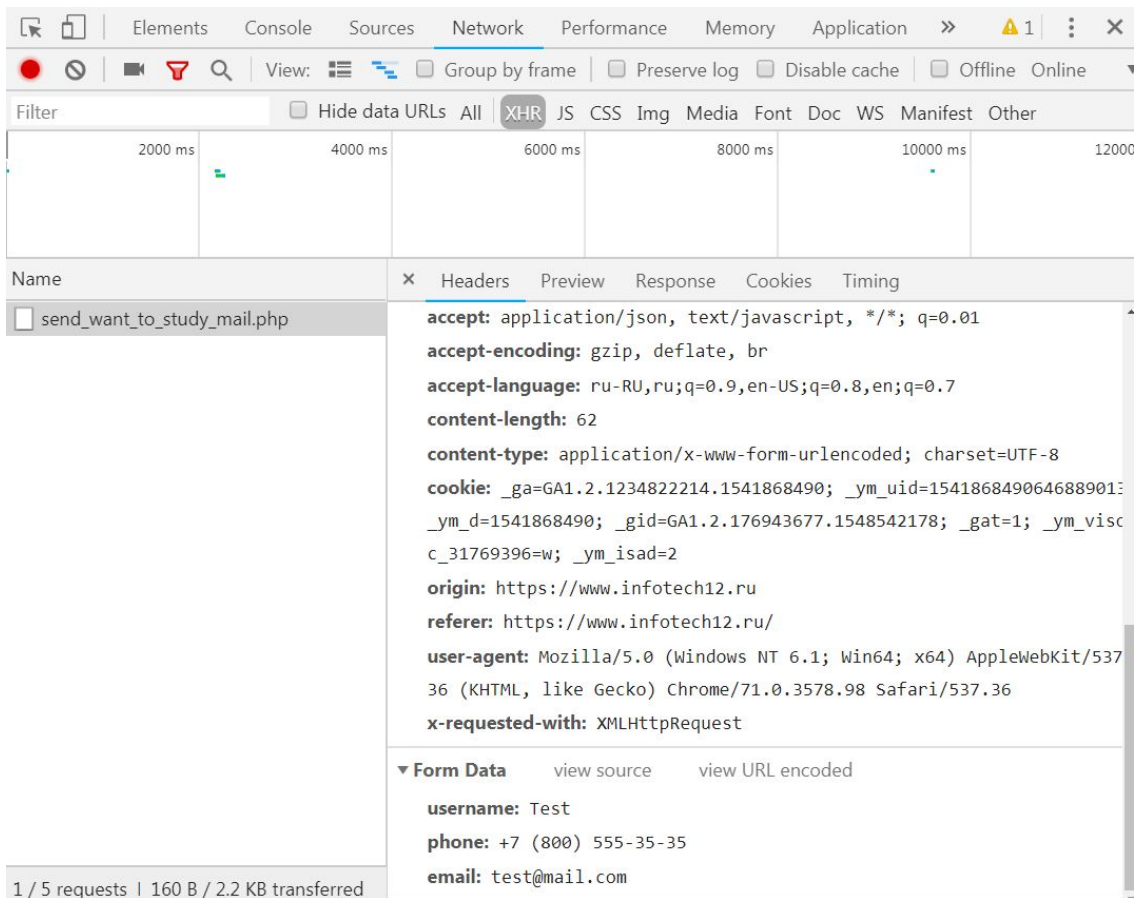
Асинхронные запросы  
можно отслеживать через  
консоль.

Для этого открываем  
вкладку **network** и  
ставим фильтр на **XHR**



# jQuery ajax()

В самом низу вкладки **Headers** есть все данные, которые мы отправляем на сервер.



The screenshot shows the Network tab in a browser's developer tools. The selected request is 'send\_want\_to\_study\_mail.php'. The 'Headers' sub-tab is active, displaying the following information:

- accept:** application/json, text/javascript, \*/\*; q=0.01
- accept-encoding:** gzip, deflate, br
- accept-language:** ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
- content-length:** 62
- content-type:** application/x-www-form-urlencoded; charset=UTF-8
- cookie:** \_ga=GA1.2.1234822214.1541868490; \_ym\_uid=154186849064688901; \_ym\_d=1541868490; \_gid=GA1.2.176943677.1548542178; \_gat=1; \_ym\_viscc\_31769396=w; \_ym\_isad=2
- origin:** https://www.infotech12.ru
- referer:** https://www.infotech12.ru/
- user-agent:** Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36
- x-requested-with:** XMLHttpRequest

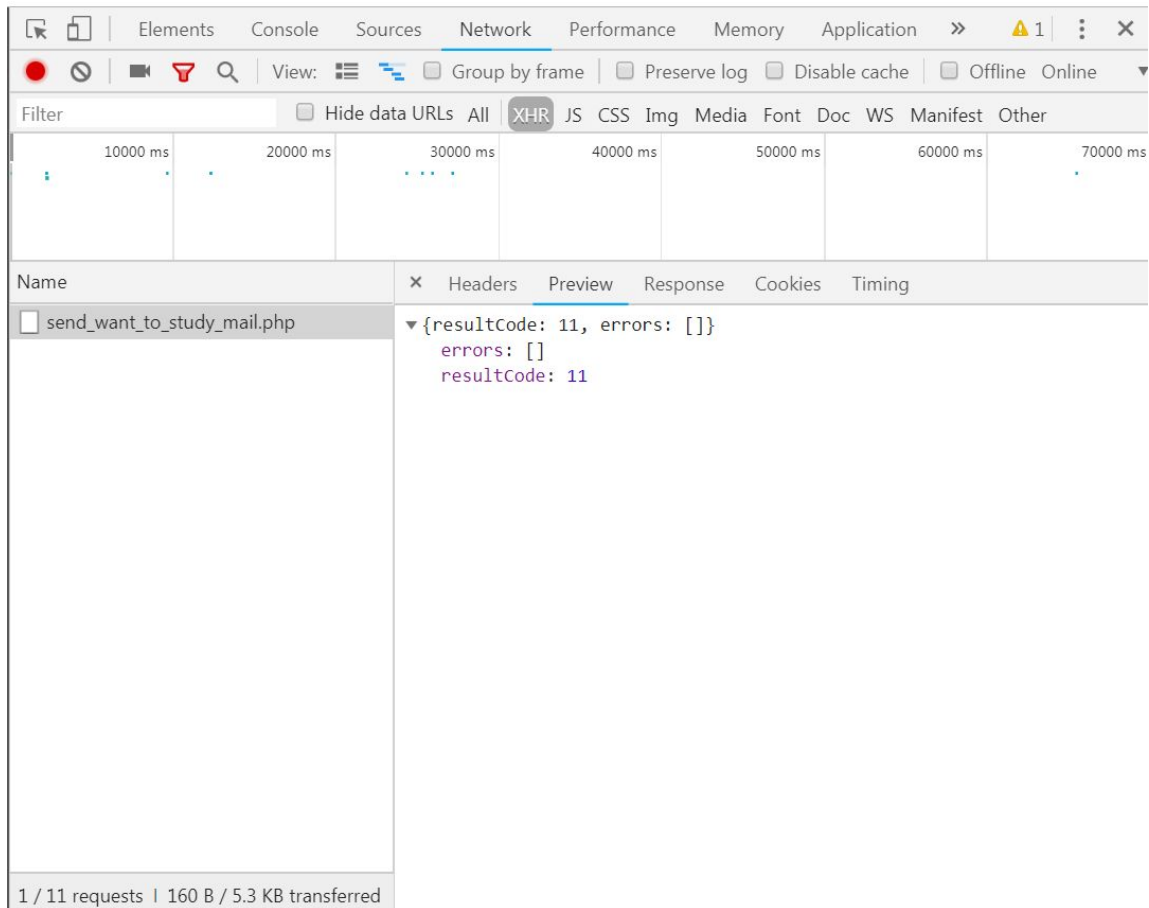
Below the headers, the 'Form Data' section is expanded, showing the following data:

- username:** Test
- phone:** +7 (800) 555-35-35
- email:** test@mail.com

At the bottom of the network panel, it indicates '1 / 5 requests | 160 B / 2.2 KB transferred'.

# jQuery ajax()

А на вкладке **Preview**  
можно посмотреть что мы  
получили от сервера.



The screenshot shows the Chrome DevTools Network tab. The top bar includes navigation icons and tabs for Elements, Console, Sources, Network, Performance, Memory, and Application. The Network tab is active, showing a list of requests. The selected request is 'send\_want\_to\_study\_mail.php'. The 'Preview' tab is selected, displaying the response data: 

```
{resultCode: 11, errors: []}
```

. The status bar at the bottom indicates '1 / 11 requests | 160 B / 5.3 KB transferred'.

# Задача №3 | Интерактивная форма (ајах)

У тебя новое письмо! Давай посмотрим что там?

<https://drive.google.com/open?id=1dY2kqDzujuW2Pa0DazRM4jxXd7YoSfkiK1tMSovVm5I>