

"11" \_\_\_\_\_ 08 2021



**Программа школьного курса  
«Информатика» для 9 класса**

Учитель: Бийболатова Н.Г.

Целевая аудитория: 9 класс

Срок реализации: 1 год

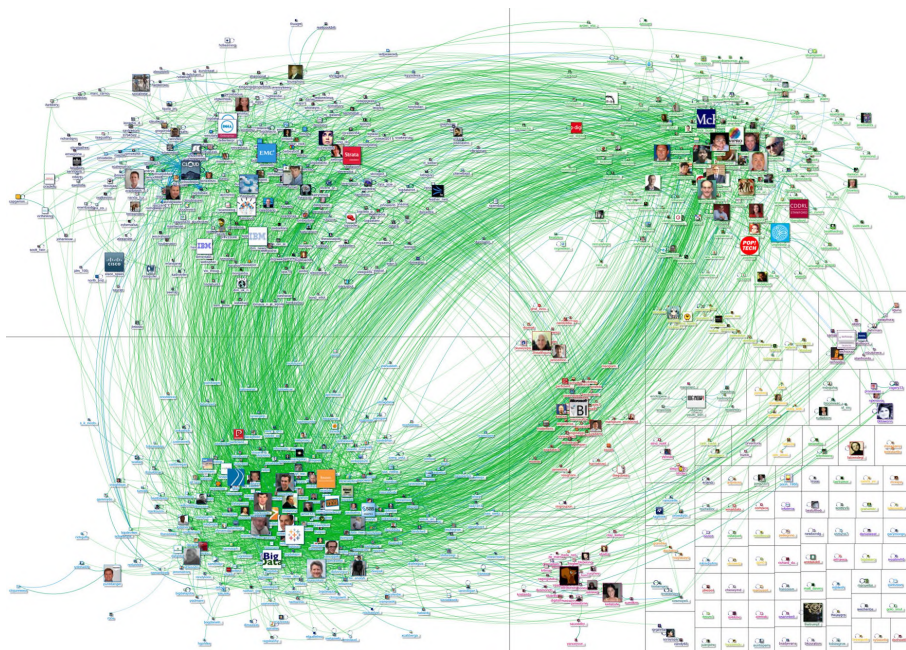
Количество часов: 68

Разработано ФГАУ «Фонд новых форм развития  
образования», г.Москва, 2020г.

# Информатика

# 9

класс



**ТОЧКА РОСТА**  
АДРЕС: МОСКВА, УЛ. ПЕТРОВКА, Д. 10  
ТЕЛЕФОН: +7 (495) 777-77-77  
САЙТ: WWW.TOCHKA-ROSTA.RU



Фонд новых форм  
развития образования  
PLUS ULTRA | ДАЛЬШЕ ПРЕДЕЛА

Общеобразовательная общеразвивающая  
программа технической направленности

*Программа школьного курса  
«Информатика» для 9 класса*

Длительность 68 ак.ч  
Для педагогов и учащихся

ФГАУ «Фонд новых форм развития образования»

Москва, 2020



**Фонд новых форм  
развития образования**  
PLUS ULTRA | ДАЛЬШЕ ПРЕДЕЛА

УДК  
ББК

**Программа школьного курса «Информатика» для 9 класса**

Авторы:

Целевая аудитория:

Срок реализации:

## Оглавление

I.	Пояснительная записка	4
II.	Общая характеристика учебного предмета	5
III.	Место курса в учебном плане	6
IV.	Личностные, метапредметные и предметные результаты освоения конкретного учебного предмета, курса	7
V.	Содержание курса	10
VI.	Поурочное планирование	11
VII.	Планируемые результаты обучения	34

ISBN

(с) ФНФРО 2020

В пособии использованы материалы из открытых источников сети Интернет. Поскольку источники, размещающие у себя информацию, далеко не всегда являются обладателями авторских прав, просим авторов использованных нами материалов откликнуться, и мы разместим указание на их авторство.

Сборник предназначен исключительно для некоммерческого использования.

## 1. Пояснительная записка

Программа данного курса направлена на получение учащимися теоретических и практических знаний в области современной информатики, касающейся работы с данными, сетевыми технологиями, веб-разработкой. Занятия курса направлены на развитие мышления, логики, творческого потенциала учеников. Программа направлена на применение в реальных проектах умения писать на уже изученном ранее языке программирования Python, на расширение знаний учащихся, касающихся программированию на этом языке. Курс содержит большое количество проектов (именуемых Кейсами), часть из которых потребует от учащихся умения работать в команде.

### Цель и задачи обучения

Целью изучения предмета «Информатика» является получение теоретических и практических знаний, умений и навыков в области современной информатики; формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира.

Для достижения поставленной цели необходимо решение следующих задач:

- создание условий для развития интеллектуальных и творческих способностей учащихся, необходимых для успешной социализации и самореализации личности;
- формирование информационной и алгоритмической культуры;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм;
- формирование умений формализации и структурирования

информации, умения выбирать способ представления данных в соответствии с поставленной задачей;

- овладение важнейшими общеучебными умениями и универсальными учебными действиями (формулировать цели деятельности, планировать ее, находить и обрабатывать необходимую информацию из различных источников, включая Интернет и др.).

## 2. Общая характеристика учебного предмета

Программа по предмету «Информатика» предназначена для изучения курса информатики учащимися основной школы. Она включает в себя пять блоков:

- Big Data
- Сетевые технологии. Интернет
- Работа с внешними API сайтов
- Как создаются сайты?
- Видео

Важная задача изучения этих содержательных линий в курсе – добиться систематических знаний, необходимых для самостоятельного решения задач, в том числе и тех, которые в самом курсе не рассматривались. Для изучения программирования используется современный язык программирования Python, а также JavaScript.

### Технологии, используемые в образовательном процессе:

- Технологии традиционного обучения для освоения минимума содержания образования в соответствии с требованиями стандартов; технологии, построенные на основе объяснительно-иллюстративного способа обучения. В основе – информирование, просвещение обучающихся и организация их репродуктивных действий с целью выработки у школьни-

ков общеучебных умений и навыков.

- Технологии компьютерных практикумов.
- Технологии реализации межпредметных связей в образовательном процессе.
- Технологии дифференцированного обучения для освоения учебного материала обучающимися, различающимися по уровню обучаемости, повышения познавательного интереса.
- Технология проблемного обучения с целью развития творческих способностей обучающихся, их интеллектуального потенциала, познавательных возможностей. Обучение ориентировано на самостоятельный поиск результата, самостоятельное добывание знаний, творческое, интеллектуально-познавательное усвоение учениками заданного предметного материала.
- Личностно-ориентированные технологии обучения, способ организации обучения, в процессе которого обеспечивается всемерный учет возможностей и способностей обучаемых и создаются необходимые условия для развития их индивидуальных способностей.
- Информационно-коммуникационные технологии.
- Технология коллективных методов обучения (работа в парах постоянного и сменного состава)

**Формы организации образовательного процесса:** фронтальные, групповые, индивидуальные, индивидуально-групповые, практикумы; урок-консультация, урок-практическая работа, уроки с групповыми формами работы, уроки-конкурсы.

### 3. Место курса в учебном плане

Данная программа предусматривает на реализацию программы по информатике в 9 классе 68 часов. Рабочая программа рассчитана на 34 учебные недели, 2 часа в неделю, общее количество часов — 68. Рабочая программа может реализовываться с использованием электронного обучения (ЭО) и дистанцион-

ных образовательных технологий (ДОТ).

### 4 Личностные, метапредметные и предметные результаты освоения учебного предмета

**Личностными результатами, формируемыми при изучении предмета информатика, являются:**

- формирование ответственного отношения к учению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в мире профессий и профессиональных предпочтений, с учётом устойчивых познавательных интересов;
- формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, общественно полезной, учебно-исследовательской, творческой и других видов деятельности.

**Метапредметные результаты изучения предмета «Информатика»:**

- умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;
- умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;

- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности её решения;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учёта интересов; формулировать, аргументировать и отстаивать своё мнение;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

#### **Предметные результаты изучения предмета «Информатика»:**

- формирование представления об основных изучаемых понятиях курса;
- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие

основных навыков и умений использования компьютерных устройств;

- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для решения конкретной задачи;
- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей, с использованием соответствующих программных средств обработки данных;
- формирование умения работать с данными, использование языка программирования Python для решения различных задач, возникающих в курсе;
- формирование понимания принципов устройства компьютерных сетей, умения работать с внешними API сайтов;
- формирование понимания того, что включает в себя профессия веб-разработчика, умение создавать несложные веб-страницы и приложения, понимание того, что такое frontend и backend-разработка;
- формирование умения работы с видеоредакторами и сервисом YouTube;
- формирование умения создавать реальные приложения, формирование умения применять накопленные знания для решения практических задач;
- использование готовых прикладных компьютерных программ по выбранной специализации
- развитие умений применять изученные понятия, результаты, методы для решения задач практического характера и задач из смежных дисциплин с использованием при необходимости справочных материалов, компьютера, пользоваться оценкой и прикидкой при практических расчётах;
- формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете, умения соблюдать нормы информационной этики и права.



## 5. Содержание курса

### Big Data (9 часов)

Что такое большие данные и как с ними работать? Модель обработки данных MapReduce. Примеры задач, решаемых с помощью парадигмы MapReduce (задача Word Count, обработка логов рекламной системы). Стек технологий Hadoop для работы с большими данными.

### Сетевые технологии. Интернет (9 часов)

История возникновения компьютерных сетей. MAC-адрес. IP-адрес, типы IP-адресов. Адрес сети и адрес узла. Маска подсети. WWW и Интернет – в чем отличие? URL-адреса. Протоколы передачи данных. Внутреннее устройство WWW. Запросы и ответы. Взаимодействие клиент-сервер. Установка веб-сервера. Безопасность в Интернете. Службы и сервисы Интернета.

### Работа с внешними API сайтов (9 часов)

Тестирование API. Форматы передачи данных XML и JSON. Определение IP с помощью api.ipify.org. Структура данных словарь в языке Python. Модуль requests языка Python и http запросы.

### Как создаются сайты (32 часа)

Основы HTML5. Гипертекст. Таблицы. Верстка простых веб-страниц. Вставка изображений. Аудио- и видео- контент на веб-странице. Дизайн веб-страницы. Каскадные таблицы стилей. Верстка страниц с помощью блоков. Формы в HTML. CSS и анимация. Основы программирования на языке JavaScript. Frontend и backend-программирование.

### Видео (9 часов)

Форматы видеофайлов. Простой видеоредактор. Обработка видео: нарезка, создание титров, добавление простых эффек-

тов. Экспорт видео. Продвинутые эффекты. Работа с сервисом YouTube.

## 6. Поурочное планирование

### Модуль 1. Big Data

#### Урок 1. Big Data: большие данные или сложные данные?

Что такое Big Data, почему важно уметь с ними работать? Примеры источников данных, для которых необходимы методы работы с большими данными, обсуждения. Характеристики больших данных, три V (volume, velocity, variety).

#### Урок 2. Большие данные в мире и в России

В каких проектах и задачах возникают большие данные? Еще немного истории и подробностей. Основные принципы работы с большими данными. Как умение работать с большими данными помогает развитию электромобилей Tesla?

#### Кейс 1. Подготовьте презентацию на тему “Что такое Big Data?”

#### Урок 3. Map Reduce

Модель распределенной обработки данных Map Reduce. Стадии Map, Shuffle и Reduce.

#### Урок 4. Задача Word Count

Классическая задача, решаемая с помощью парадигмы MapReduce – задача Word Count (для каждого слова, хотя бы раз встречающегося в наборе документов вычислить сколько именно раз это слово встретилось). Вспомнить, как решить подобную задачу в простом варианте (есть текстовый файл, нужно определить частоту появлений каждого слова или каждого символа) – идея сортировки подсчетом, словари в языке Python.



### Урок 5. Map Reduce и сложная задача Word Count

Как Map Reduce может помочь решить задачу Word Count в трудном варианте. Другие примеры задач, решаемых с помощью парадигмы MapReduce (например, задача обработки логов рекламной системы).

### Урок 6. Стек технологий Hadoop для работы с большими данными

История. Что такое Hadoop, основные компоненты. Пример запуска Map Reduce-задачи на Hadoop.

### Кейс 2. Составьте интеллект-карту на тему “Map Reduce и Word Count”.

### Урок 7. Реализация на Python задачи подсчета количества слов в простом варианте. Подготовка

Словари в языке Python. Простые задачи на словари.

### Урок 8. Реализация на Python задачи подсчета количества слов в простом варианте. Подготовка

Словари в языке Python. Задача подсчета количества символов в тексте. Идея сортировки подсчетом.

### Урок 9. Кейс 3: “Подсчет частоты встречающихся слов в произведениях уроков литературы” (реализация в простом варианте, с использованием словарей).

#### Полезные ресурсы:

1. <https://habr.com/ru/company/dca/blog/267361/>
2. <https://in-scale.ru/blog/big-data>
3. <https://pythonworld.ru/typy-dannyx-v-python/slovari-dict-funkcii-i-metody-slovarej.html>
4. <https://habr.com/ru/company/yandex/blog/332688/>
5. Марк Лутц, Изучаем Python, 4ое издание (2011 год).

### Что получит ученик по окончании модуля:

- познакомится с понятием BigData, проблемами, связанными с обработкой больших данных и способами решения этих проблем
- научится пользоваться продвинутыми возможностями языка python (словари) для решения различных задач
- потренируется в практическом программировании, написав программу подсчета частоты встречающихся слов в тексте

## Модуль 2. Сетевые технологии. Интернет.

### Урок 10. Что такое Интернет?

История возникновения компьютерных сетей. Для чего нужно соединять компьютеры в сети? Как соединить в сеть несколько компьютеров (по кругу? каждый с каждым? все компьютеры подключены к одному серверу? недостатки этих идей). Сетевая карта. Принципы работы сетевого хаба и свитча. Интернет = объединение нескольких сетей.

### Урок 11. Как «подписывают» устройства в сети?

MAC-адрес: уникальность, где выдается, почему нельзя общаться по MAC-адресам. IP-адрес, как способ «пронумеровать» устройства. Сколько памяти выделяется на один IP-адрес? Сколько адресов можно в принципе записать? IPv4 и IPv6 - зачем нужно увеличивать длину IP-адреса?

### Урок 12. Структура IP-адреса

Типы IP-адресов: публичные/частные; статические/динамические, etc. Белый IP и адреса внутри сети. Как устройства ищут друг друга по IP-адресам? Адрес сети и адрес узла. Маска подсети. Два способа записи маски: префиксный и десятичный. Кейс 4. Создайте интеллект-карту на тему “Структура IP-адреса”

### Урок 13. Как путешествует информация?

Передача информации из сети через основной шлюз. ipconfig.

Трассировка маршрута с помощью tracer. Передача пакетов с помощью ping. Для чего нужны DNS-серверы. ipconfig с параметрами.

#### Урок 14. URI и URL-адреса

Домены разного уровня. Доменные регистраторы (например, reg.ru). В чем отличие Интернета и WWW? WWW - всемирная паутина, страницы, связанные гиперссылками, Интернет - объединение сетей, протоколы и т.п.

#### Кейс 5. Подготовьте презентацию “Интернет и WWW. История: факты и выдумки.”

#### Урок 15. Как работает WWW?

Браузер. Веб-сервер. Взаимодействие браузера и веб-сервера: запросы (requests) и ответы (responses). Языки для веб-программирования: php, python, java, ruby. Взаимодействие клиент-сервер. Протокол HTTP. Коды возврата.

#### Кейс 6. Составьте интеллект-карту на тему: “Как работает WWW?”

#### Урок 16. Методы передачи данных

Как устроены и чем отличаются GET и POST запросы. Как установить себе веб-сервер?

#### Урок 17. Безопасность передачи данных

Безопасность в интернете. SSL и HTTPS (SSL over HTTP). Авторизация и cookies. Сетевые угрозы. Мошенничество. Правила личной безопасности в Интернете.

#### Кейс 7. Составьте интеллект-карту на тему: “Данные в интернете. Передача и безопасность.”

#### Урок 18. Службы Интернета

Электронная почта. Обмен файлами (FTP). Форумы. Общение в

реальном времени. Интернет-магазины. Электронные платёжные системы.

#### Полезные ресурсы:

1. <https://habr.com/ru/post/253803/>
2. <https://fb.ru/article/367974/post-get-chem-otlichayutsya-zaprosyi-drug-ot-druga>

Что получит ученик по окончании модуля:

- Разберется с тем, чем отличаются Интернет и WWW.
- Познакомится с устройством сетей и принципами передачи информации по сетям.
- Узнает, что такое IP-адреса, какие они бывают и для чего нужны.
- Познакомится с тем, как следят за безопасностью в Интернете.

### Модуль 3. Работа с внешними API сайтов.

#### Урок 19. Что такое API?

Тестирование API. Анализ запроса с помощью сервиса apitester.com, например, на примере запросов <https://ru.wikipedia.org/wiki/Python>, <https://ru.wikipedia.org/wik/Python>, <https://ru.wikipedia.org/wiki/python>.

Коды ответа сервера. Обсудить, каковы коды ответа для каждого из приведенных выше запросов. Ошибки клиента и ошибки сервера. Что записано в строке <https://yandex.ru/search/?text=Python&lr=213> после знака вопроса? Параметры запросов.

#### Урок 20. Форматы передачи данных

Формат данных XML и формат данных JSON. Аналогия с правильными скобочными последовательностями.

#### Урок 21. Сервис ipify.org

Сервис ipify.org. Определение IP с помощью api.ipify.org. Опре-

деление геолокации по IP с помощью geo.ipify.org. Регистрация на сервисе для того, чтобы иметь возможность получать данные геолокации.

### Урок 22. Структура данных словарь

Для того чтобы удобно обрабатывать запросы, нужно знать, как работать со словарями в Python. Ключи и значения. Создание словаря в Python. Добавление и удаление элемента. Перебор элементов словаря.

### Урок 23. Модуль requests языка Python и http-запросы

Установка модуля requests. Запись запроса с помощью requests. Функция get(). Создание консольного приложения, отправляющего запрос на сервис <https://api.ipify.org> и получающего IP-адрес вашего компьютера.

### Урок 24. Кейс 8: Создание приложения «Определение данных геолокации по IP»

Создание консольного приложения, по IP-адресу компьютера определяющее страну, область, город, почтовый индекс, временную зону, широту и долготу. Запись запроса с параметрами в библиотеке requests.

Начало работы над программой может быть, например, таким:

```
import requests
url_1 = "https://api.ipify.org"
answer_1 = requests.get(url_1)
ip_address = answer_1.text
print(ip_address)
url_2 = "https://geo.ipify.org/api/v1"
params = {
    "apiKey": "тут должен быть ваш apiKey, полученный при
регистрации на сервисе",
    "ipAddress": ip_address
}
answer_2 = requests.get(url_2, params = params)
print(answer_2.json())
```

### Урок 25. Кейс 8: Создание приложения «Определение данных геолокации по IP»

Превращение консольного приложения, созданного на прошлом уроке, в графическое.

### Урок 26. Кейс 8: Создание приложения «Определение данных геолокации по IP»

Превращение консольного приложения, созданного на прошлом уроке, в графическое.

### Урок 27. Кейс 8: Создание приложения «Определение данных геолокации по IP».

Дополним созданное приложение возможностью сохранить найденные данные в файл.

#### Что получит ученик по окончании модуля:

- Познакомится с тем, что такое внешние API сайтов, и научится с ними работать с помощью соответствующих библиотек языка Python.

## Модуль 4. Как создаются сайты?

### Урок 28. Как пишут веб-сайты?

Содержание, оформление, работа с данными. Первые примеры на языке HTML. Средства разработки: онлайн-инструменты и IDE. Инструменты разработчика в Google Chrome.

### Урок 29. Структура HTML-документа

Структура html-документа. Основные теги. Отличия HTML5 и HTML4.

### Урок 30. Работа с текстом и гиперссылками

Текстовые веб-страницы. Простейшая веб-страница. Заголовки. Абзацы. Специальные символы. Списки.

### Урок 31. Кейс 9: Создание простой веб-страницы.

### **Урок 32. Графика на веб-страницах**

Вставка изображений. Форматы графических файлов и их особенности, важные для веба.

### **Урок 33. Таблицы в html**

Как создавать таблицы. Форматирование таблицы. Верстка простой html-страницы с использованием таблиц.

### **Урок 34. Звук, видео и встраиваемые объекты на веб-страницах**

Форматы аудиофайлов для веб. Вставка аудио на веб-страницы. Видеоконтент на веб-странице. Встраиваемые объекты.

### **Урок 35. Займемся дизайном**

В чем заключается работа дизайнера? Намечаем структуру страниц. Создаем основные элементы. Шапка (header). Подвал (footer). Меню сайта.

### **Урок 36. Кейс 10: проектируем небольшой сайт.**

Парная работа: ребята разбиваются на пары, один “заказчик”, другой “дизайнер и веб-программист”. Заказчик должен составить техзадание веб-программисту. Веб-программист должен описать возможности сайта, рассказать, что и как в нем будет работать.

### **Урок 37. Каскадные таблиц стилей (CSS)**

Что такое CSS и зачем он нужен. Общая структура css-таблиц. Элементы, классы, id. Основные селекторы. Инструменты для работы с css.

### **Урок 38. Вставка css в html-документ**

Какие бывают шрифты и как их правильно использовать. Интересные примеры использования CSS.

**Кейс 11. Найдите в интернете 5-8 примеров необычного использования CSS и расскажите о них классу.**

### **Урок 39. Верстка структуры страницы с помощью блоков**

Плавающие блоки. Верстка шапки. Верстка подвала. Верстка меню.

### **Урок 40. Формы в HTML**

POST и GET запросы. Основные форматы полей форм. Кнопки. Скрытые поля.

### **Урок 41. CSS и анимация**

### **Урок 42. Кейс 12. Анимированные кнопки на вашем сайте.**

Парная работа: школьники разбиваются на группы из 2-3 человек. Каждая группа придумывает дизайн и реализует “самую необычную кнопку в мире”. В конце урока – конкурс кнопок.

### **Урок 43. Язык JavaScript. Введение**

JavaScript или ECMAScript? Где применяется JavaScript: frontend-программирование, backend-программирование, скрипты. Трансляторы и компиляторы. JavaScript – интерпретируемый язык. Где писать? Три способа связать скрипт с html-файлом. Первые простые примеры, теги `<script>`.

### **Урок 44. Язык JavaScript. Введение**

Типы данных в JavaScript (number, string, boolean, null, undefined). Динамическая типизация. Переменные. Комментарии.

### **Урок 45. Условный оператор в JavaScript**

Операторы сравнения. Условный оператор. Преобразование типов. Условный оператор в одну строку. Оператор switch.

### **Урок 46. Циклы for и while**

Простые примеры программ с циклами. Бесконечные циклы. Операторы break и continue.

### **Урок 47. Функции в языке JavaScript**

Три способа объявления функций. Вызов функций. Ключевые слова this и arguments.

#### **Урок 48. Объекты как ассоциативные массивы в JavaScript**

Ключи и значения. Сравнение ассоциативных массивов. Присваивание ассоциативных массивов.

#### **Урок 49. Объекты как массивы в JavaScript**

Использование объектов как обычных массивов. Список данных. Операции с массивами. Практические задачи.

#### **Урок 50. Объекты в JavaScript: ООП**

Представления об ООП, методы и свойства.

**Кейс 13. Создайте презентацию “Основы ООП в нескольких слайдах”.**

#### **Урок 51. DOM-модель веб-страницы. Использование скриптов**

Использование JavaScript для создания frontend’а. Как его соединить с веб-страницей.

Работа с DOM. Узлы DOM-модели. Атрибуты и свойства. Стили. Размеры и координаты. Основы работы с событиями.

#### **Урок 52. Кейс 14. Разработка и создание одностраничного приложения. Командная работа.**

Ребята разбиваются на группы. Одна группа проектирует сайт, другая его разрабатывает.

#### **Урок 53. Завершение работы над приложением.**

#### **Урок 54. Обработка данных на стороне сервера**

Что такое backend-программирование? Введение в node.js. Хостинг с использованием node.js.

#### **Урок 55. Кейс 15. Создание первого веб-сервера**

#### **Урок 56. Кейс 16. Проектная работа по созданию сайта.**

Ребятам предлагается разбиться на группы из 2-4 человек. В каждой группе нужно выбрать дизайнера, человека отвечающего за подбор информации, веб-программистов. Ребята могут сами

выбрать тему для небольшого сайта, который им предстоит создать (возможно, этот сайт будет посвящен какому-либо школьному предмету, любимому виду спорта, или любимому музыкальному направлению).

#### **Урок 57. Продолжение работы над сайтом.**

#### **Урок 58. Продолжение работы над сайтом.**

#### **Урок 59. Презентация и защита своего проекта перед классом.**

#### **Полезные ресурсы:**

1. <http://htmlbook.ru/html>
2. <https://www.w3schools.com/html/>
3. <http://html.net/tutorials/html/>

Что получит ученик по окончании модуля:

- Познакомится с языком разметки HTML и каскадными таблицами стилей CSS
- Разберется с тем, что такое frontend и что такое backend в веб-разработке
- Изучит основы языка JavaScript и использует его как для работы над frontend’ом, так и для работы над backend’ом
- Создаст несколько сайтов разного уровня сложности

#### **Модуль 5. Видео**

#### **Урок 60. Форматы видеофайлов**

Особенности видеоформатов AVI, MPEG, WMV, 3gp, FLV и других.

#### **Урок 61. Программы для обработки видеофайлов. Простой видеоредактор**

Обзор программ. Установка Movavi Видеоредактор 15. Рабочие области программы

## Урок 62. Изучаем возможности Movavi

Нарезка видео. Создание титров. Добавление простых эффектов. Экспорт видео в формат AVI.

## Урок 63. Более продвинутые видеоредакторы

Видеоредактор Lightworks и его возможности. Интересные примеры обработки видео, например, картина в рамке <https://videosmile.ru/lessons/read/kak-pravilno-obrabatyivat-video-luchshie-programmyi-dlya-obrabotki-video.html>.

**Урок 64. Кейс 17. Научи учителя.** Ученикам заранее (дома) предлагается найти в интернете один-два необычных видео-эффекта, научиться их реализовывать, а на уроке показать свою работу учителю и классу. Авторы самых интересных эффектов обучают одноклассников.

## Урок 65. Знакомство с сервисом YouTube.

История YouTube. Особенности YouTube. Форматы файлов, которые можно загрузить на YouTube. Как скачать файл с YouTube напрямую?

**Кейс 18. Создайте презентацию: “Расскажи бабушке про YouTube”.** Создайте презентацию, в которой доступно опишите возможности и достоинства YouTube.

## Урок 66. Создание и настройка своего YouTube-канала.

Создаем свой YouTube-канал. Настройки канала. Загрузка видео на канал. Создаем и настраиваем плейлисты. Как удалить плейлист с канала.

**Кейс 19 Создайте презентацию: “Самые крутые образовательные каналы на YouTube”.** Какой школьный предмет вам больше всего нравится? Найдите пять интересных каналов, посвященных этому предмету, и расскажите про них в презентации.

**Урок 67. Кейс 20. Работа в группе. Создание рекламного ролика “Необычный взгляд на обычные вещи”.**

Школьникам предлагается прорекламировать что-то, связанное со школой, например, школьную библиотеку (или городскую библиотеку), конкретный предмет, и сделать это с юмором. Для работы над роликом ребята разбиваются на группы по 2-3 человека. Видео нужно обработать с помощью любого видеоредактора и записать на свой канал.

**Урок 68. Завершение работы над роликом. Презентация проектов, конкурс.**

## Полезные ресурсы:

1. [https://www.movavi.ru/videoeditor/?admitad\\_uid=c992c13512ce65caba8700737f1ca220&tagtag\\_uid=c992c13512ce65caba8700737f1ca220](https://www.movavi.ru/videoeditor/?admitad_uid=c992c13512ce65caba8700737f1ca220&tagtag_uid=c992c13512ce65caba8700737f1ca220)
2. <https://lifehacker.ru/besplatnye-videoredactory/>
3. <https://ru.wikipedia.org/wiki/YouTube>
4. <https://iklife.ru/youtube/kak-sozdat-kanal.html>

## Что получит ученик по окончании модуля:

- Познакомится с простой программой для обработки видеоизображений Windows Movie Maker
- Научится обрабатывать видеоизображения, добавляя титры, несложные эффекты и переходы
- Познакомится с сервисом YouTube
- Создаст свой канал на YouTube, настроит его и добавит созданное видео на этот канал

## 7. Планируемые результаты обучения.

### Важнейшими умениями/знаниями являются следующие:

- умение пользоваться персональным компьютером и его периферийным оборудованием;
- умение следовать требованиям техники безопасности, гигиены, эргономики и ресурсосбережения при работе со средствами информационных и коммуникационных технологий;



- умение осуществлять взаимодействие посредством электронной почты, чата, форума;
- умение искать информацию с применением правил поиска (построения запросов), в компьютерных сетях, некомпьютерных источниках информации (справочниках и словарях, каталогах, библиотеках) при выполнении заданий и проектов по различным учебным дисциплинам;
- знакомство с понятием BigData, проблемами, связанными с обработкой больших данных и способами решения этих проблем;
- умение пользоваться продвинутыми возможностями языка Python (словари) для решения различных задач;
- умение реализовывать алгоритмы на языке программирования Python;
- понимание принципов работы компьютерных сетей, в том числе сети Интернет;
- умение пользоваться сервисами Интернета, а также понимание основ безопасности при работе в Интернете
- понимание того, что такое внешние API сайтов, и умение с ними работать с помощью соответствующих библиотек языка Python
- понимание того, как происходит процесс создания сайтов, что такое backend и frontend
- умение писать на языке разметки HTML5, пользоваться каскадными таблицами стилей
- умение создавать веб-страницы
- знание языка программирования JavaScript и его применение для работы с backend'ом
- знакомство с процессом разработки сайтов, умение работать в команде, разрабатывать и реализовывать идеи в рамках технического задания
- умение обрабатывать видеоизображения, добавляя титры, несложные эффекты и переходы
- умение работать с каналами на сервисе YouTube
- умение выбирать способ представления своего проекта с использованием соответствующих программных средств.



[www.roskvantorium.ru/fond](http://www.roskvantorium.ru/fond)



**Фонд новых форм  
развития образования**

PIUS ULTRA | ДАЛЬШЕ ПРЕДЕЛА