

# Ruby Essential

8 уроков

## Урок 1. Введение в Ruby

### Обзор, цель и назначение урока

Целью вводного урока является ознакомление учащихся с языком программирования Ruby, его основными особенностями и отличиями от других языков. К концу урока учащиеся смогут создать свою первую программу на языке Ruby.

### Изучив материал данного занятия, учащийся сможет:

- Кратко излагать основные принципы и историю языка Ruby;
- Понимать для чего предназначенный данный язык программирования, а в каких сферах он применяется;
- Самостоятельно устанавливать и работать с интерпретатором Ruby;
- Создавать проекты в среде разработки RubyMine, настраивать данную среду разработки;
- Писать первые простые программы на Ruby.

### Содержание урока

1. Что такое Ruby?
2. История языка Ruby.
3. Преимущества и недостатки языка.
4. Применение Ruby на примере глобальных проектов.
5. Интерпретатор Ruby. Интегрированная среда разработки: установка, обзор возможностей и начало работы.

## Урок 2. Базовые типы данных. Работа с числами

### Обзор, цель и назначение урока

Целью данного урока является ознакомление учащихся с такими понятиями, как переменные, константы, и типы данных в Ruby. Во время урока учащиеся на практике подробно рассмотрят тип данных Numeric, научатся работать с разными представлениями чисел, а также выполнять операции над ними; попрактикуются преобразовывать объекты из численного в строковый тип данных, и наоборот.

### Изучив материал данного занятия, учащийся сможет:

- Отличать переменные от констант в языке Ruby, и работать с ними.
- Работать с типами данных объектов: определять тип данных, преобразовывать, понимать особенности каждого из типов.
- Выполнять основные операции над числами в зависимости от приоритета операций.
- Писать простые программы для работы с числами.

### Содержание урока

1. Переменные и константы.
2. Типы данных в Ruby. Для чего нужны типы данных ?
3. Работа с числами. Целые числа и числа с плавающей точкой.
4. Основные методы преобразования типов данных.
5. Операторы и приоритеты.

## Урок 3. Строки, работа со строками. Время и дата

### Обзор, цель и назначение урока

Целью данного урока является ознакомление учащихся с такими типами данных, как String и Time в Ruby, а также основными методами для работы с этими типами данных. Во время урока учащиеся на практике рассмотрят способы создания строк, познакомятся с понятием конкатенации строк, научатся основным методам для работы со строками, такими как: получение доступа, управление регистром, сравнение строк, изменение строк и другие методы. А также подробно рассмотрят основные методы для работы со временем и датой в Ruby.

#### **Изучив материал данного занятия, учащийся сможет:**

- Понимать, когда стоит применять тип данных String(Строка).
- Работать со строками в Ruby: создавать строки, изменять их и выполнять различные методы над строками.
- Работать с типом данных Time. Применять основные методы для работы со временем и датой в разных форматах.

#### **Содержание урока**

1. **Строки. Способы создания строк.**
2. **Конкатенация строк.**
3. **Основные методы для работы со строками: получение доступа и сравнение строк, управление регистром, изменения строк.**
4. **Время и Дата в Ruby. Основные методы для работы.**

### **Урок 4. Массивы и хеши**

#### **Обзор, цель и назначение урока**

Целью данного урока является ознакомление учащихся с такими типами данных, как Array-Массивы и Hash-Хеши, а также основными методами для работы с этими типами данных. Во время урока учащиеся на практике рассмотрят способы создания массивов и хешей, получение доступа к элементам данных структур, а также научатся применять основные методы и операторы для работы с данными структурами.

#### **Изучив материал данного занятия, учащийся сможет:**

- Понимать и различать типы данных Массивы и Хеши.
- Создавать массивы, получать доступ к элементам массива, и работать с основными методами массивов
- Создавать хеши, получать доступ к ключам и их значениям, применять основные методы для работы с хешами

#### **Содержание урока**

1. **Понятие массива. Создание массива и доступ к элементам .**
2. **Основные методы для работы с массивами.**
3. **Многомерные массивы.**
4. **Понятие хеша. Создание и доступ к парам значение-ключ.**
5. **Методы для работы с хешами.**

### **Урок 5. Управляющие структуры. Часть 1. Условия**

#### **Обзор, цель и назначение урока**

Целью данного урока является ознакомление учащихся с такими понятиями , как алгоритмы и условные конструкции. Учащиеся на практике познакомятся с понятием алгоритма и разновидностями алгоритмов, изучат условные конструкции if, if-else, if-elsif-else, unless, а также case и тернарный оператор.

#### **Изучив материал данного занятия, учащийся сможет:**

- Понимать и различать структуру алгоритмов.
- Применять на практике условные конструкции типа `if`, `if-else`, `if-elsif-else`, `unless` и `case`.
- Использовать условные операторы `if` и `unless` в качестве модификаторов.
- Применять тернарный оператор.

## Содержание урока

1. Что такое условные конструкции ?
2. Инструкции `if-else-if`, `if-elsif-else`.
3. Работа `if` в качестве модификатора.
4. Инструкция `unless`.
5. Инструкция `Case`.
6. Условный оператор `?:`.

## Урок 6. Управляющие структуры. Часть 2. Циклы

### Обзор, цель и назначение урока

Целью данного урока является ознакомление учащихся с такими понятиями, как циклы и итераторы. Учащиеся на практике рассмотрят такие циклические конструкции, как **`while`**, **`until`** и **`for`**, познакомятся с методом `loop`, и научатся определять итераторы-самоделные циклические конструкции.

### Изучив материал данного занятия, учащийся сможет:

- Понимать и применять циклические алгоритмы.
- Применять в своих программах различные циклы.
- Пользоваться методом `loop`.
- Определять самоделные циклические конструкции.

## Содержание урока

1. Что такое цикл?
2. Организация циклов: циклы **`while`**, **`until`** и **`for`**.
3. Метод `loop`.
4. Итератор — определяем самоделные циклические конструкции.

## Урок 7. Методы и блоки

### Обзор, цель и назначение урока

Целью данного урока является ознакомление учащихся с понятием методов и блоков. На уроке будут подробно рассмотрены понятия определение метода, работы с аргументами методов, вызов метода и возвращаемые значения метода, а также особенная разновидность методов — синглтон методы. На практике учащиеся в течении урока будут создавать методы, различные по функционалу, работать с аргументами этих методов, а также вызывать их. Учащиеся на практике научатся использовать блоки и работать с аргументами блока. Также познакомятся с `proc` и `lambda` объектами, рассмотрят отличия между ними и научатся применять их в своих программах.

### Изучив материал данного занятия, учащийся сможет:

- Создавать собственные методы.
- Работать с аргументами методов.
- Работать с методами, определенными для единственного объекта.
- Вызывать методы.
- Работать с конструкциями блоков.
- Применять `proc` и `lambda` объекты.

## Содержание урока

1. Что такое методы?
2. Определение простых методов.
3. Работа с аргументами метода.
4. Вызов метода.
5. Определение синглтон-методов.
6. Блоки – определение и применение.
7. Создание и работа Proc и Lambda объектов.

## Урок 8. Классы и объекты.

### Обзор, цель и назначение урока

Целью данного урока является ознакомление учащихся с таким понятием, как классы и объекты. Данный урок является введением в объектно-ориентированное программирование в Ruby. Учащиеся на практике научатся создавать собственные классы, работать с объектами классов и методами классов. Также научатся работать с различными переменными, и правильно их применять, рассмотрят область видимости методов и посмотрят, чем они отличаются.

### Изучив материал данного занятия, учащийся сможет:

- Создавать собственные классы.
- Работать к методами классов(get, set, initialize и другие).
- Создавать экземпляры классов и работать с ними.
- Задавать область видимости переменных и методов.

## Содержание урока

1. Понятие ООП.
2. Что такое класс?
3. Создание классов.
4. Объекты класса и методы.
5. Область видимости переменных и методов.