

Ruby Essential

Управляющие структуры. Циклы.

Ruby Essential

Автор курса



Юля Гончаренко

Ruby Essential

После урока обязательно



Повторите этот урок в видео формате на
[ITVDN.com](http://itvdn.com)

Доступ можно получить через руководство вашего
учебного центра



Проверьте как Вы усвоили данный материал
на [TestProvider.com](http://testprovider.com)

Ruby Essential

Тема

Управляющие конструкции.
Циклы.

Ruby Essential

Управляющие структуры. Условия

1. Что такое цикл?
2. Организация циклов : циклы while, until и for
3. Метод loop.
4. Итератор - определяем самодельные конструкции.

Ruby Essential

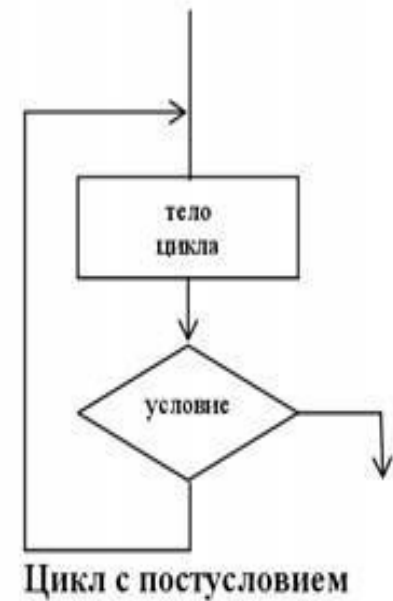
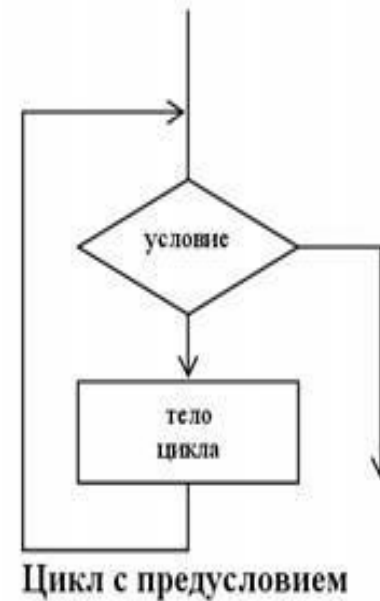
Что такое цикл?

Циклический алгоритм — алгоритм, в котором набор инструкций повторяется заданное количество раз или пока не выполнится заданное условие.

В циклических алгоритмах можно выделить такие обязательные составляющие, как:

- условие
- тело цикла
- счетчик

В зависимости от того, где стоит условие(до тела цикла или после), циклы можно разделить на циклы с постусловием и циклы с предусловием.

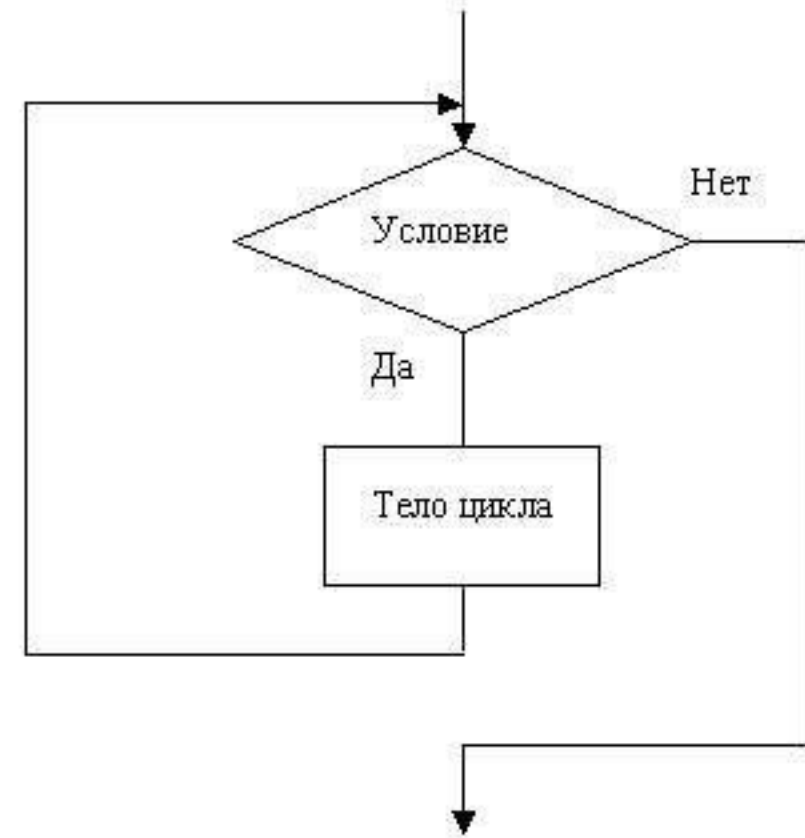


Ruby Essential

Цикл while

Цикл while : выполняется тело цикла пока(while) определенное условие вычисляется в истину (true).

Если в виде условия данному циклу задать параметр – true, цикл будет бесконечным (выполняться бесконечное количество раз).



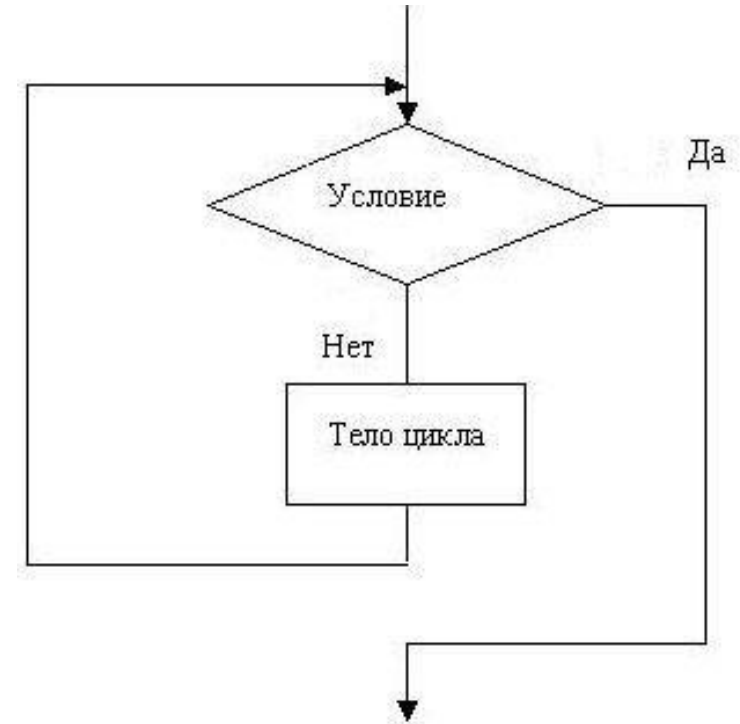
Ruby Essential

Цикл Until

Цикл `until`: выполняется тело цикла до тех пор(`until`), пока условие не станет истинным(то есть вычисляется изначально как `false` или `nil`).

Условие должно размещаться между ключевыми словами `until` и `do`.

В большинстве случаев используется цикл `while`. `Until` в основном используют в тех случаях, когда это существенно улучшает читаемость кода.



Ruby Essential

While и Until в виде модификаторов

Для компактности кода можно использовать циклы `while` и `until` в виде модификаторов.

Условие, ключевое слово цикла и тело цикла при использовании модификатора располагаются на одной строке. Не смотря на особенность записи модификатора, изначально проверяется условие, а потом уже выполняется тело цикла, как и в обычных циклах `while` или `until`.

Особенность модификатора: если тело цикла все таки является составным выражением, используются ключевые слова `begin` и `end` для обозначения, при этом сначала выполняется тело цикла , а далее уже проверяется условие цикла. В таком случае модификатор начинает работать как цикл с постусловием.

Тело цикла ***while*** условие
Тело цикла ***until*** условие

begin
код
end until/while условие

Ruby Essential

Цикл for ... in

Цикл for является циклической конструкцией, он осуществляет последовательный перебор элементов перечисляемого объекта(такого как массив).

При каждой итерации значение элемента присваивается заданной переменной цикла, а после выполняется тело цикла.

Переменной в данном случае может выступать как одиночная переменная, так и список переменных.

Цикл for можно наглядно увидеть при проходе по значениям диапазона

```
ar = [1,2,3,4,5,6]
for item in ar
  puts item
end
```

Ruby Essential

Метод loop

Метод loop является итератором, который создает бесконечный цикл.

Ключевые слова do и end можно опустить, если вместо этого поставить скобки {}.

При помощи ключевого слова break можно остановить цикл на определенной итерации.

При помощи ключевого слова next можно пропускать определённые значения.

```
i = 0
loop do
  i += 1
  puts "#{i}"
end
```

Ruby Essential

Итераторы-самодельные циклические конструкции

Итератор - это метод, который принимает блок и выполняет код в блоке для элементов коллекций (массивов или хэшей).

Рассмотрим такие итераторы, как `times`, `each`, `map` и `upto`. За итераторами стоит конструкция `yield`.

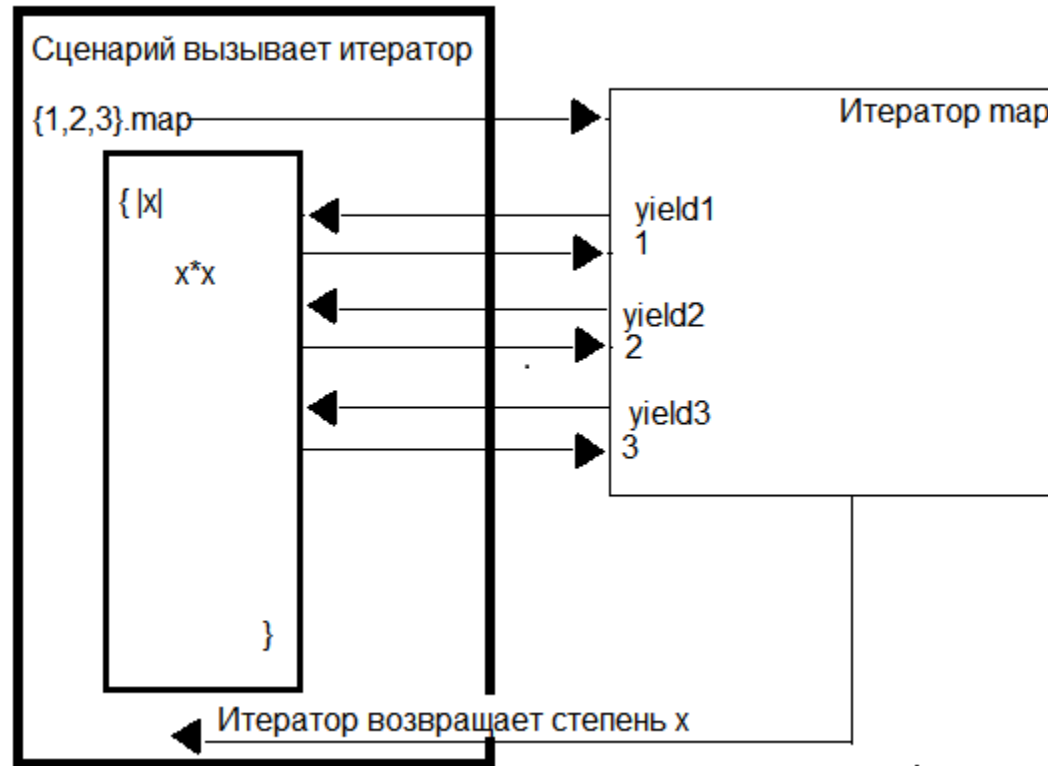
`Yield` – это инструкция, которая возвращает управление от метода-итератора к методу, вызывавшему итератор.

Чтобы реализовать циклическую конструкцию, итератор вызывает `yield` по несколько раз.

Ruby Essential

Итераторы-самодельные циклические конструкции

Итератор `map` - это метод, возвращающий результат в виде нового массива после перебора всех объектов в блоке, к которому он был вызван.



Ruby Essential

Each, times и upto.

Самым простым методом-итераторов является each. Все итерируемые объекты в Ruby имеют метод each чтобы можно было последовательно пробегать по значениям в итерируемом объекте(например, массиве) и делать что-то с каждым его элементом.

```
arr1 = [1,2,3,4,5]
arr1.each do |i|
  puts i
end
```

Итератор times - позволяет выполнять цикл и совершать определенные операции которые передаются ему в блоке указанное количество раз.

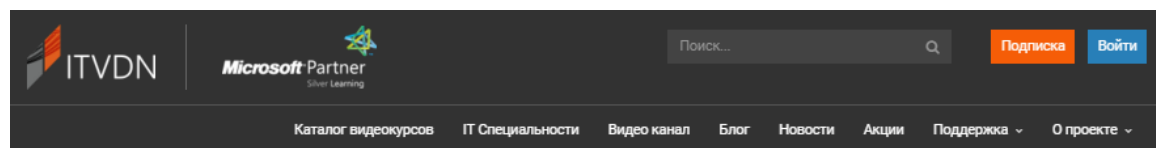
```
2.upto(4) {|i| puts i}
```

Аналогичен итератору times итератор upto, при этом в данном итераторе включена как начальная, так и конечная точка итерации .

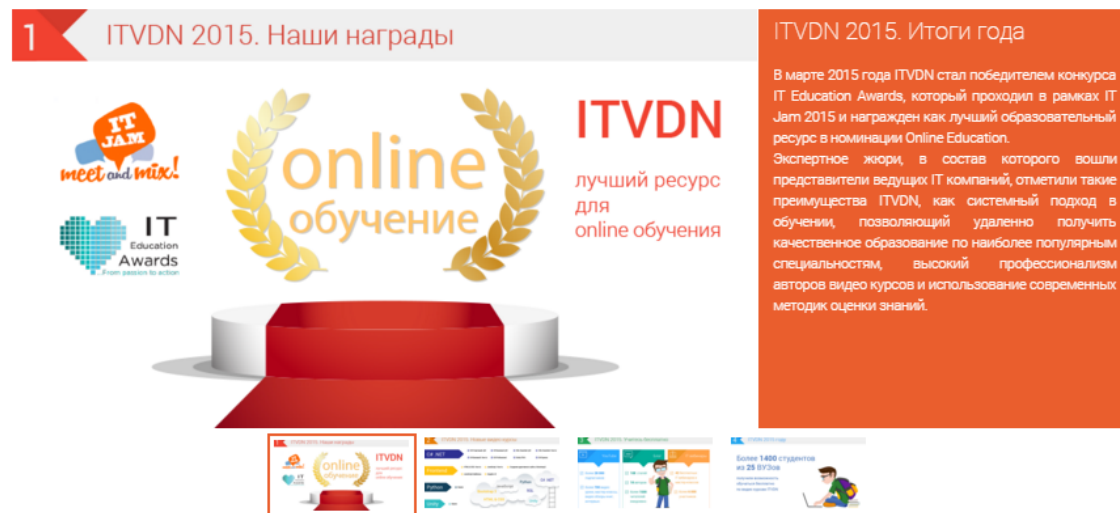
```
2.upto(4) {|i| puts i}
```

Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.



Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics

Новые видео

Исключения	0
Итераторы и генераторы	0

Популярные видео курсы

Видео курс C# Стартовый (для начинающих)	9 уроков (16 ч. 3 мин.)
Видео курс по шаблонам проектирования	29 уроков (16 ч. 7 мин.)

Теги

.NET Developer
Frontend Developer



Проверка знаний

TestProvider.com

TestProvider

Мы поможем людям оценить себя

Регистрация Войти

Главная Каталог Сертификация Microsoft Поддержка О нас

Тестирование

Языки программирования и информационные технологии

Microsoft

C# ASP.NET MVC JavaScript Patterns Of Design SQL Architecture Guide WCF HTML&CSS XML SEO WPF HTML5&CSS3 JQuery XNA SharePoint GUI for Android Windows Azure Platform Microsoft Patterns&Practices TFS SCRUM ReSharper TDD WWF LINQ Entity Framework Windows Forms Refactoring Microsoft Expression Blend 4 Windows Phone 8 Windows 8 AppStore Visual Studio Tips&Tricks MSF MEF SilverLight AJAX MEF Service Oriented Architecture

Пройти тест

Наши партнеры

Microsoft Partner CyberBionic ITVDN PROMETRIC TEST CENTER PEARSON VUE Authorized Test Center Windows Azure Cloud Partner EBA

Дополнительные ресурсы:

Очное обучение On-line обучение Видео обучение

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на TestProvider.com

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



TRANSACT-SQL

Q&A

Информационный видеосервис для разработчиков программного обеспечения

