

Ruby Essential

Управляющие структуры. Условия

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

Тема

Управляющие структуры. Условия

Ruby Essential

Управляющие структуры. Условия

1. Что такое условные конструкции?
2. Условия if, if-else, if-elsif-else.
3. Работа if в качестве модификатора.
4. Условие unless.
5. Условие case.
6. Тернарный оператор ?: .

Ruby Essential

Простое понятие алгоритма

Алгоритм – это набор инструкций, описывающих порядок действий исполнителя для достижения необходимого результата.

Алгоритмы разделяют на :

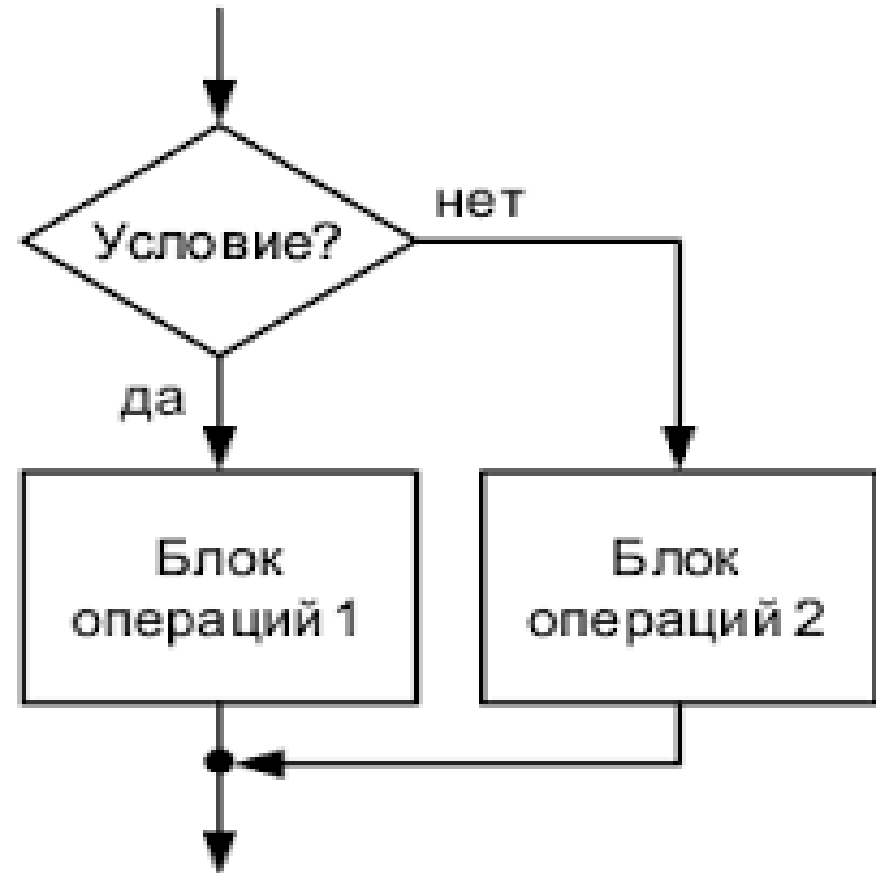
- Линейные алгоритмы – это такие алгоритмы, при которых набор инструкций выполняется строго по порядку и один раз.
- Алгоритмы ветвления- это алгоритмы, которые имеют хоть один оператор ветвления, после выполнения которого и определяется последующая последовательность инструкций для выполнения.
- Циклические алгоритмы - это алгоритмы, в которых набор инструкций повторяется заданное количество раз или пока не выполнится заданное условие.

Ruby Essential

Что такое условные конструкции?

Условная конструкция- это базовая конструкция, которая содержат условия, и от хода выполнения данного условия и зависит последующая работа всей конструкции (исполнение тех или иных инструкций).

К таким конструкциям относят конструкции `if`, `if-else`, `if-else-if`, `if-elsif`, `case`, `unless` и тернарный оператор.



Ruby Essential

Условие if

Наиболее распространенным условием является условие **if** (если).

Код между ключевыми словами if-end выполняется лишь в том случае, если выражение условия вычисляется true.

Ограничителем выражения является символ новой строки или ключевое слово then.

```
if 2+2==4 then puts "Все верно!"  
end
```

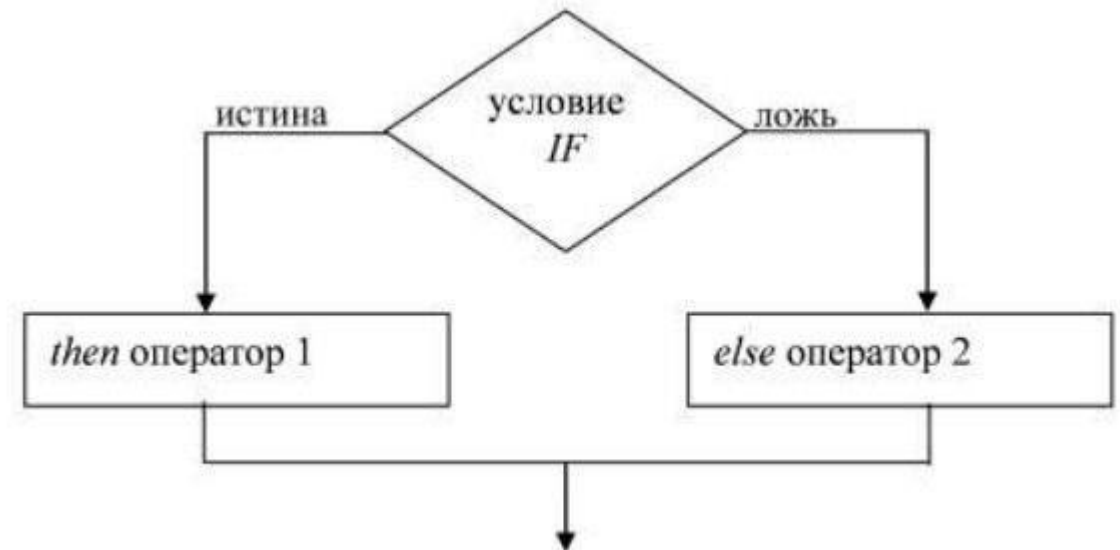


Ruby Essential

Условие IF-ELSE

- Если продолжить инструкцию `if` то можно написать выражение, которое будет выполняться, если условие не отработало верно. Для этого используется ключевое слово `else`(иначе).
- Код между словами `else` и `end` выполняется в том случае, если условие `if` вычислилось в `false` или `nil`.

```
if balance > 50.00
  puts "Ваша покупка оплачена"
else
  puts "Ваш лимит на карте меньше заданной суммы"
end
```



Ruby Essential

Конструкция со множественными условиями elsif

Если требуется проверить сразу несколько условий, то между if и else добавляется ключевое слово elsif. Elixir является сокращенной формой записи if else.

Блок elsif может содержать ключевое слово then в виде разделителя.

Принцип работы аналогичен: если первое условие if-true, то выполняется инструкция, следующая за ним. Если if вычисляется в false или nil, то проверяются условия elsif по порядку их объявления. Если последний блок elsif вычисляется в false или nil, то выполняется блок кода между else и end, если таковой имеется.

Если блока нет, то никакая инструкция не выполняется.

```
puts "Сколько будет 3х3 ?"  
i=gets.to_i  
if i==6  
  puts "Это 3+3"  
elsif i==9  
  puts "Это верно"  
elsif i==12  
  puts "Это 3х4"  
else  
  puts "Ваш ответ не верный"  
end
```

Ruby Essential

Работа if в качестве модификатора

Для написания малого однострочного условия удобнее всего воспользоваться так называемой формой if в виде модификатора.

Особенностью такой формы является то, что сначала объявляется исполняемый блок, далее идет ключевое слово if а затем выражение условия.

Не смотря на такое объявление, сначала проверяется условие, а потом уже в случае успешного вычисления исполняется код.

```
x = 0  
x += 1 if x.zero?  
puts x
```

Ruby Essential

Условие unless

Такое условие является полной противоположностью `if` то есть код выполняется лишь в том случае, если связанное с ним условие вычисляется как `false` или `nil`.

Unless, как и `if` может ветвиться в одном или нескольких направлениях (с использованием `else`).

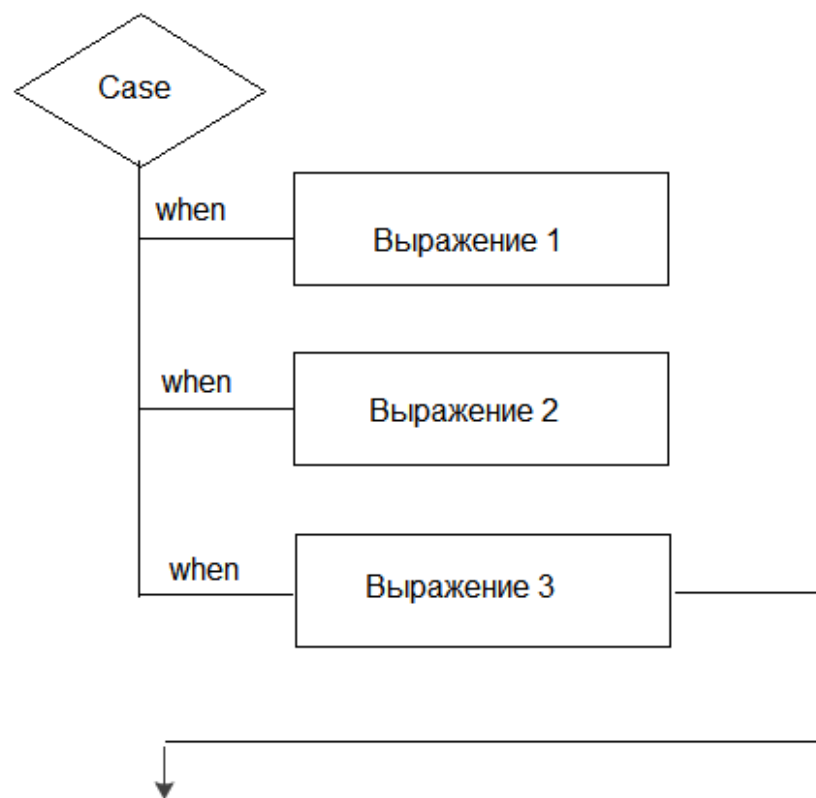
```
puts "Какая сумма у вас на счету?"
balance=gets.to_i
unless balance<50
  puts "Ваша покупка оплачена"
else
  puts "Ваш лимит на карте меньше заданной суммы"
end
```

```
x = 0
x += 1 unless x.zero?
puts x
```

Ruby Essential

Условие Case

Данная условная конструкция представляет собой так называемый “множественный выбор”.



Ruby Essential

```
puts "Дни недели"
puts "Введите номер от 1 до 7 "
ch=gets.to_i
case ch
  when 1
    puts "Понедельник"
  when 2
    puts "Вторник"
  when 3
    puts "Среда"
  when 4
    puts "Четверг"
  when 5
    puts "Пятница"
  when 6
    puts "Суббота"
  when 7
    puts "Воскресенье"
  else
    puts "Вы ввели неправильный номер"
  end
end
```

Ruby Essential

Тернарный оператор

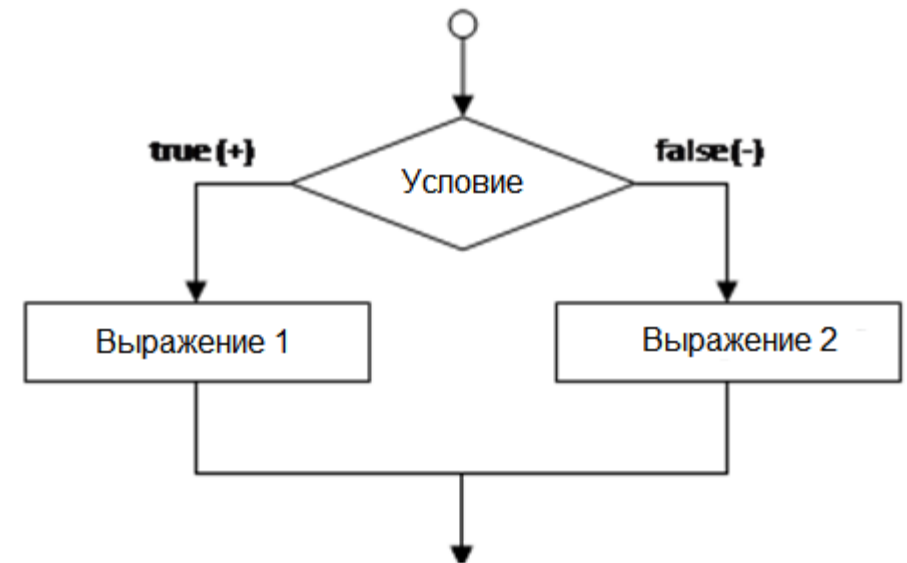
Тернарный оператор является единственной конструкцией с 3мя операндами.

Первый операнд идет перед знаком ?, второй – между ? и : , а третий-за знаком : .

Операторы вычисляются по порядку.

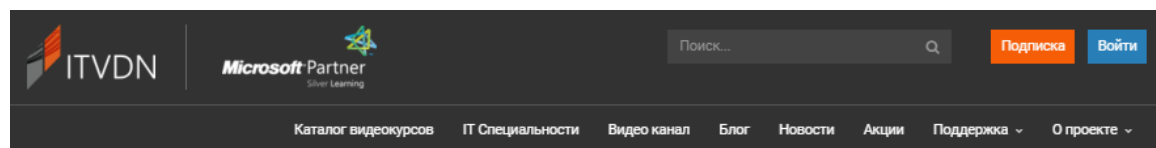
Первый операнд является проверяемым условием, второй – выполняется, если условие верно (как блок then), а третий – если условие вычислилось в false или nil.

В тернарном операторе всегда один из операндов не вычисляется.



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.

1 ITVDN 2015. Наши награды

ITVDN 2015. Итоги года

В марте 2015 года ITVDN стал победителем конкурса IT Education Awards, который проходил в рамках IT Jam 2015 и награжден как лучший образовательный ресурс в номинации Online Education. Экспертное жюри, в состав которого вошли представители ведущих IT компаний, отметили такие преимущества ITVDN, как системный подход в обучении, позволяющий удаленно получить качественное образование по наиболее популярным специальностям, высокий профессионализм авторов видео курсов и использование современных методик оценки знаний.

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 – 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.



TRANSACTION-SQL

Q&A

Информационный видеосервис для разработчиков программного обеспечения

