

Programlama Dilleri Laboratuvar Föyü

Ruby Programlama Dili 1

Ruby Tanımı ve Dil Yapıları

Şevket Umut ÇAKIR

1 Giriş

Ruby 'hızlı ve kolay', nesneye yönelik yazılım geliştirmeye yarayan yorumlanan bir betik dilidir. Doğrudan işletim sistemi çağrılarını yapabilme yeteneği, güçlü metin işlemleri, düzenli ifadeler ve geliştirme sırasında anında geri besleme özelliklerine sahiptir[*Ruby Kullanıcı Kılavuzu*]. Ruby dilinde değişken bildirimi yoktur ve her şey bir nesnedir. Bellek yönetimi otomatik olarak gerçekleştirilir.

1.1 Kurulum ve Entegre Geliştirme Ortamları

Ruby yorumlayıcısı Linux ve Mac OS işletim sistemlerinde kurulu olarak gelmektedir ama sürüm yükseltme işlemleri için farklı araçlar kullanılabilir. Windows işletim sisteminde kolay kurulum yapmak için RubyInstaller programı kullanılabilir.

Sadece Ruby programlama dilini kurmak için bu adresteki bağlantılardan biri kullanılabilir. Ruby ortamını bilgisayara kurmak ve sürüm yönetimi yapmak için **rvm** ve **rbenv** adında iki araç bulunmaktadır. Belki de Ruby'nin en çok kullanıldığı rails kütüphanesi ile birlikte Ruby ortamı Go Rails web sitesindeki kılavuz yardımıyla kurulabilir.

Ruby ortamı bilgisayarınıza kurulmadan çevrim içi olarak da kullanılabilir. Bunun için aşağıda bağlantılar verilmiştir.

- RubyMine : **pau.edu.tr** uzantılı e-posta adresi ile ücretsiz olarak kullanılabilir. Gelişmiş kod tamamlama ve hata ayıklama özelliklerine sahiptir.
- Aptana Studio : Eclipse geliştirme ortamı baz alınarak Ruby dili için özelleştirilmiş bir entegre geliştirme ortamı. Entegre hata ayıklayıcı, terminal, sürüm kontrol sistemi ve konuşlandırıcı(deploy) gibi özellikleri bulunmaktadır.
- Atom : Genel amaçlı bir metin(kod) düzenleyicisi.
- repl.it : Çevrim içi Ruby yorumlayıcı ve kabuğu
- Coding Ground kabuğu ve yorumlayıcısı

2 Konu Anlatımı ve Deney Hazırlığı

Bu bölümde etkileşimli kabuk, değişkenler, işlemler ve karar-kontrol yapıları anlatılmaktadır.

2.1 Ruby Etkileşimli Kabuğu

Ruby etkileşimli kabuğu basit işlemlerin yapılabileceği gibi karmaşık yapıların da oluşturulabileceği bir sanal makinedir. Başarılı bir Ruby kurulumundan sonra terminal penceresinden **irb** komutu verilerek kabuk çalıştırılabilir. Kurulum türüne bağlı olarak kabuk çalıştırıldığında Şekil 1'dekine benzer bir görünüme sahip olacaktır. Kabuk içinde Ruby kodları etkileşimli olarak çalıştırılabilir ve oluşturulan nesnelerin çıktıları ekranda görülebilir. Aşağıdaki şekilde ekrana metin yazdıran **puts** komutunun geri döndürdüğü değer **nil** olarak gözükmektedir.

```
irb(main):001:0> puts "Merhaba, dünya!"
Merhaba, dünya!
=> nil
irb(main):002:0> █
```

Şekil 1: Ruby etkileşimli kabuğu görüntüsü

2.2 Açıklama Satırları

Tek satırdan oluşan açıklamalar **#** sembolü ile, çok satırlı açıklamalar ise **=begin** **=end** sembolleri arasına yazılarak tanımlanır.

```
# Açıklama satırı # karakteri ile başlar
x = 5 # satır sonu açıklama
=begin
Bu da çok satırlı
bir açıklama
=end
```

2.3 Değişkenler ve Veri Türleri

Ruby değişkenleri için bildirime gerek yoktur. Atama işlemi ile değişkenin türü belirlenir. Ruby dinamik tip kontrol sistemine sahiptir. Ruby dilinde her şey bir nesnedir ve veri türüne özgü metotları bulunmaktadır. Değişkenin türünü öğrenmek için o değişkenin **class** metodu kullanılabilir. Ruby dilinde Integer, Float, String, Array ve Hash en çok kullanılan veri türleridir. Aşağıdaki örnekte değişken tanımlama ve değişkenlerin türleri arasındaki dönüşüm vardır.

Tanımlanan her bir değişkeni ayırt eden kimlik değeri vardır. Kimlik değerleri nesneleri birbirinden ayırt etmek için kullanılır. Integer ve Float tipleri için aynı

değere sahip değişkenlerin kimliği aynıdır, fakat String ve bir çok tür için farklı kimliklere sahip olacaktır.

```
x = 7 # değişken tanımlama
puts x.class # Integer
puts x.is_a?(Integer) # true
puts x.to_s.class # String
y = gets.chomp # Kullanıcıdan değer alma
# .chomp ile sondaki "\n" karakteri silinir
# "merhaba" girelim
y = y.to_f # Tamsayıya dönüştürme
puts y.class # Float
puts y # 0.0
a, b = "merhaba", "merhaba"
puts a.object_id == b.object_id # false
```

Yukardaki örnekte kullanılan `puts`, `is_a?`, `class`, `object_id` birer metottur. Ruby dilinde metot çağırısı yapmak için parantez kullanımına gerek yoktur.

2.4 İşleçler

Ruby dilinde işleçler infix notasyonunu(işlem ortada) kullanır. Tablo 1’de çok kullanılan işleç türleri ve örnekleri verilmiştir.

Tablo 1: Ruby dilinde kullanılan işleçler

Tür	İşleçler	Örnek
Aritmetik	+, -, *, /, %, **, //	x+y
Karşılaştırma	==, !=, <=>, >, <, >=, <=, ==~, .eql?, equal?	a>b
Atama	=, +=, -=, *=, /=, %=, **=, //=	x += 2
Bit tabanlı	&, , ^, ~, <<, >>	a&b
Mantıksal	and, or, not, &&, , !	true and false
Üyelik	.include?	liste.include? x
Aralık	.. ve ...	(1..10)

2.5 Karar-Kontrol Yapıları

Bu bölümde if, unless, case, times, while, until, for ve each yapıları incelenecektir. Ruby dilinde bloklar `end` sözcüğü ile sonlandırılır.

2.5.1 if ve unless Yapısı

if yapısı bir kontrol mekanizmasıdır. Bir koşulun değerine göre işlemler yapılır veya yapılmaz. unless ise anlamsal olarak if’in tersi işleve sahiptir. Aşağıda örnek kullanımları vardır:

```

print 'Bir sayı girin: '
sayi = gets.chomp.to_i
if sayi<10
  puts "sayı 10'dan küçük"
elsif sayi>=10 and sayi<=20
  puts "sayı 10 ile 20 arasında"
else
  puts "sayı 20'den büyük"
end
unless sayi>=2
  puts "sayı 2'den küçük"
else
  puts "sayı 2'den büyük veya eşit"
end

puts "alternatif if kullanımı" if sayi<10
puts "unless alternatif" unless sayi>100

```

2.5.2 case Yapısı

`case` yapısı `when` ile verilen seçeneklerden hangisi doğru ise onu gerçekleştirir. `when` içinde aralık dışında düzenli ifadeler(regular expression) de verilebilir. Aşağıda örnek kullanımı vardır:

```

print "Ne kadar benzin kaldı(%)? "
benzin = gets.chomp.to_i
durum=case benzin
  when 0
    "yolda kaldınız"
  when (1..15)
    "benzin istasyonuna gidin"
  when (16..100)
    "yola devam edebilirsiniz"
  else
    "hatalı girdi!!"
  end

puts durum

```

2.5.3 times Metodu

Integer sınıfından nesneler ile kullanılan ve verilen sayı kadar bir işin yapılmasını sağlayan döngü türüdür.

```

100.times do
  puts "I will not talk in class!"
end

```

```
10.times do |i| # döngü değişkeni olarak i
  puts i*2
end
20.times{ |i| puts 2**i } # 2'nin kuvvetlerini yazdır
```

2.5.4 while ve until Döngüleri

while döngüsü koşul doğru olduğu sürece çalışan bir döngü türüdür. **until** döngüsü ise koşul doğru olana kadar çalışan bir döngüdür. Metin içindeki `"#{toplam}"` ifadesi değişkenin değerini yazdırmak için kullanılır. İçeri yazılan ifade herhangi bir Ruby ifadesi olabilir.

```
toplam = 0
while (girdi=gets.chomp) != ""
  toplam += girdi.to_i
end
puts "Girilen sayıların toplamı: #{toplam}"
```

2.5.5 for ve each Döngüleri

for ve **each** döngüleri bir veri seti veya aralık üzerinde dolaşmayı sağlar. Yazım biçimi farklı olsa da aynı işleve sahiptir.

```
for i in 1..5
  puts i
end

(1..10).each do |i|
  puts i
end

dizi = [2,3,5,7,11]
dizi.each { |i| puts i }
```

3 Deneyin Uygulanması

Bu deneyde taş-kağıt-makas oyunu, mükemmel sayı ve Collatz sanısı uygulamaları bulunmaktadır. Bu uygulamaların Ruby dilinde <http://bilmoodle.pau.edu.tr/> bulunan aktiviteler ile çözülmesi amaçlanmaktadır.

3.1 Taş, Kağıt ve Makas

Bu uygulamada iki kişi ile oynanan klasik taş, kağıt, makas oyununu gerçekleştirmek istenmektedir. Kullanıcıdan içerikleri {"taş", "kağıt", "makas"} olan iki adet metin alınacaktır. İlk girilen metin birinci oyuncunun seçimini, ikinci girilen metin de diğer oyuncunun seçimini göstermektedir. Girilen metinlere göre

ekrana “birinci oyuncu kazandı”, “ikinci oyuncu kazandı” veya “berabere” mesajlarını yazdıran Ruby kodunu yazınız.

3.2 Mükemmel Sayı

Tam bölenlerinin toplamı kendisine eşit olan sayılara mükemmel sayı denir. 28 ve 96 mükemmel sayılara verilebilecek iki örnektir. Kullanıcıdan alına sayının mükemmel sayı olup olmadığını bulan, mükemmelse ekrana “mükemmel” aksi takdirde “mükemmel değil” yazan Ruby kodunu yazınız.

3.3 Collatz Sanısı

Henüz doğruluğu kanıtlanmamış olan Collatz sanısı yapılan iki işlem sonunda 1’e indirgenebileceğini söyler. Bu iki işlem aşağıdaki gibidir:

$$f(n) = \begin{cases} \frac{n}{2}, & n \text{ çift ise} \\ 3n + 1, & n \text{ tek ise} \end{cases}$$
$$a_i = \begin{cases} n, & i=0 \\ f(a_{i-1}), & i>0 \end{cases}$$

Collatz sanısında bir seri sayının kendisi ile başlar ve 1 ile sonlanır. Örneğin 13 sayısı için Collatz serisi 13, 40, 20, 10, 5, 16, 8, 4, 2, 1 sayılarından oluşan 10 sayılıklı bir seridir.

Kullanıcının girdiği herhangi bir sayının Collatz serisinde kaç adımda 1’e ulaştığını bulan Ruby programını yazınız.

Kaynaklar

20 Dakikada Ruby

key20dk

20 Dakikada Ruby. URL: <https://www.ruby-lang.org/tr/documentation/quickstart/> (son erişim: 11.3.2019).

Abstract: Ruby’nin temellerini kapsayan hoş bir kılavuz. Başlayıp bitirmesi yirmi dakikadan fazla zamanınızı almayacak.

Programlama Öğrenin

keyltp

Programlama Öğrenin. URL: <http://pine.fm/LearnToProgram/> (son erişim: 11.3.2019).

Abstract: Chris Pine tarafından, yeni başlayanlar için yazılmış harika bir kılavuz. Eğer programlama bilmiyorsanız, buradan başlayın.

Ruby & Rails Searchable API Docs

keyrubydocs

Ruby & Rails Searchable API Docs. URL: <http://rubydocs.org/> (son erişim: 11.3.2019).

Abstract: Akıllı arama özellikleri olan Rails ve Ruby dökümantasyonu.

Ruby & Ruby on Rails programlama kılavuzları **keyprogkil**

Ruby & Ruby on Rails programlama kılavuzları. URL: http://www.meshplex.org/wiki/Ruby/Ruby_on_Rails_programming_tutorials (son erişim: 11.3.2019).

Abstract: Hem Ruby hem Rails için zengin içerikli kılavuzlar sunan, iyi organize edilmiş bir site.

Ruby Çekirdek Referansı **keyrubdoc**

Ruby Çekirdek Referansı. URL: <http://www.ruby-doc.org/core> (son erişim: 11.3.2019).

Abstract: RDoc ile kaynak kodundan üretilmiş bir belge. Bu belge tüm çekirdek sınıfları ve modülleri (String, Array, Symbol gibi) içerir.

Ruby Koans **keykoans**

Ruby Koans. URL: <http://rubykoans.com/> (son erişim: 11.3.2019).

Abstract: Koans Ruby öğrenmek için bir aydınlanma yolu üzerinden yürümenizi sağlar. Amaç Ruby dilini, deyimlerini, yapısını ve birkaç çok kullanılan yapılarını öğrenmek ama bununla beraber size kültürünü de öğretiyor.

Ruby Kullanıcı Kılavuzu **keykullanicikil**

Ruby Kullanıcı Kılavuzu. URL: <http://www.belgeler.org/uygulamalar/ruby/ruby-ug.html> (son erişim: 11.3.2019).

Abstract: Ruby diliyle ilgili pek çok alanda hoş görüşler sunan bu belgenin orijinali Yukihiko Matsumoto (Ruby'nin yaratıcısı) tarafından yazılmış olup, Goto Kentaro ve Mark Slagell tarafından İngilizce'ye, Pınar Yanardağ tarafından da Türkçe'ye çevrilmiştir.

Ruby Öğrenmek **keyrubylearning**

Ruby Öğrenmek. URL: <http://rubylearning.com/> (son erişim: 11.3.2019).

Abstract: Dile yeni olanlar ve Ruby'nin kavramlarına ve yapısına sağlam bir giriş yapmak isteyenler için Ruby öğrenci notlarından derlenmiş mükemmel bir koleksiyon.

Ruby Programlama **keyprogruby**

Ruby Programlama. URL: <http://www.ruby-doc.org/docs/ProgrammingRuby/> (son erişim: 11.3.2019).

Abstract: Pratik Programcılar'ın Kitabı 'nın ilk baskısı. İngilizce Ruby kaynakları arasında hatırı sayılır bir yere sahip olan bu kitabı, çevrimiçi okuyabilirsiniz.

Ruby Programlama Wikikitabı**keywikibook**

Ruby Programlama Wikikitabı. URL: http://en.wikibooks.org/wiki/Ruby_programming_language (son erişim: 11.3.2019).

Abstract: Başlangıç ve ileri düzeyde içeriğe sahip olan çevrimiçi, kapsamlı bir kılavuz.

Ruby SSS**keyrubysss**

Ruby SSS. URL: <http://faq.rubygarden.org/> (son erişim: 11.3.2019).

Abstract: Ruby deyimlerini öğrenmek ya da tipik programlama tekniklerinin Ruby yoluyla nasıl yapıldığını göstermek için toplanmış soru ve cevaplar.

Ruby Standart Kütüphane Referansı**keystdlib**

Ruby Standart Kütüphane Referansı. URL: <http://www.ruby-doc.org/stdlib> (son erişim: 11.3.2019).

Abstract: Yine RDoc ile üretilmiş bu referans belgesi, standart kütüphaneyi kapsayacak niteliktedir.

Ruby Stil Klavuzu**keyrubystyle**

Ruby Stil Klavuzu. URL: <https://github.com/rubocop-hq/ruby-style-guide> (son erişim: 13.3.2019).

Ruby ve Diğer Diller**keyrubyvediger**

Ruby ve Diğer Diller. URL: <https://www.ruby-lang.org/tr/documentation/ruby-from-other-languages/> (son erişim: 11.3.2019).

Abstract: Başka bir dilden Ruby'ye mi geçiyorsunuz? C, C++, Java, Perl, PHP, ya da Python, farketmez! Bu makale sizi saracak!

RubyDoc.info**keyrubydoc.info**

RubyDoc.info. URL: <http://www.rubydoc.info/> (son erişim: 11.3.2019).

Abstract: Ruby gem'leri için dokümanlar ve GitHub Ruby projeleri için tek kaynak site.

Tayfun Öziş Erikan Sunumlar**tayfunoziserikan**

Tayfun Öziş Erikan Sunumlar. URL: <https://speakerdeck.com/tayfunoziserikan> (son erişim: 10.3.2019).

Try Ruby!**keytryruby**

Try Ruby! URL: <https://ruby.github.io/TryRuby/> (son erişim: 11.3.2019).

Abstract: Ruby’yi tarayıcınızda deneyebileceğiniz interaktif bir tutorial. Bu 15 dakikalık çalışma ile bu dile yenibaşlayanların ilk denemelerini yapmaları amaçlanmıştır.

Tüm Yeni Gelenlerin Bilmesi Gereken Şeyler **keyyenigelen**

Tüm Yeni Gelenlerin Bilmesi Gereken Şeyler. URL: <http://www.rubygarden.org/ruby?ThingsNewcomersShouldKnow> (son erişim: 11.3.2019).

Abstract: Ruby Garden Wiki ‘nin “yeni gelenlerin bilmesi gereken şeyler” sayfası.

Why’s (Poignant) Guide to Ruby **keypoignant**

Why’s (Poignant) Guide to Ruby. URL: <http://mislav.uniqpath.com/poignant-guide/> (son erişim: 11.3.2019).

Abstract: Ruby’nin hikayelerle, nüktelerle ve karikatürlerle anlatıldığı sıradışı ama ilginç bir kitap. Orjinali why the lucky stiff tarafından yazılmıştır, bu klavuz Ruby’ye yeni başlayanlar için bir klasiktir.

Kaynakların bir çoğu <https://www.ruby-lang.org/tr/documentation/> adresinden derlenmiştir.