

آموزش زبان برنامه نویسی Ruby

محمدرضا حقیری

۱۵ بهمن ۱۳۹۳

فهرست مطالب

۳	۱	مقدمه
۳	۱.۱	این کتاب برای چه کسانی است؟
۳	۲.۱	آیا میتوانیم این کتاب را به اشتراک بگذاریم؟
۳	۳.۱	چگونه این کتاب را بخوانیم؟
۴	۲	پایه ها و مقدمات یادگیری روبی
۴	۱.۲	نصب روبی
۴	۱.۱.۲	وبسایت روبی
۴	۲.۱.۲	نصب از مخازن
۵	۲.۲	استفاده از محیط تعاملی
۵	۳.۲	چند دستور در محیط تعاملی
۶	۴.۲	نوشتن اسکریپت در یک فایل
۶	۱.۴.۲	انتخاب ویرایشگر
۶	۲.۴.۲	پسوند فایل های روبی
۶	۳.۴.۲	نوشتن و اجرای اسکریپت از طریق مفسر
۷	۴.۴.۲	اجرای اسکریپت با استفاده از اجازه ها
۷	۵.۲	جمع بندی
۸	۶.۲	تمرینات
۹	۳	انواع داده و متغیرها
۹	۱.۳	در این فصل چه فرا خواهید گرفت؟
۹	۲.۳	داده های عددی
۹	۱.۲.۳	اعداد صحیح
۱۰	۲.۲.۳	اعداد ممیز شناور
۱۰	۳.۲.۳	اعداد اوکتال
۱۱	۴.۲.۳	اعداد هگزادسیمال
۱۲	۳.۳	متغیرهای بولین
۱۲	۴.۳	متغیرهای رشته ای
۱۲	۵.۳	آرایه ها و هش ها
۱۲	۶.۳	تعریف متغیرها
۱۳	۱.۶.۳	تبدیل نوع متغیرها
۱۳	۷.۳	جمع بندی
۱۳	۸.۳	تمرینات

۱ مقدمه

زبان روبی، یک زبان اسکریپتی، شی گرا، تابعی و مدرن است. این زبان توسط یوکیهیرو ماتسوموتو^۱ ایجاد شده و هم اکنون توسعه دهندگانی از سراسر جهان، آن را توسعه میدهند. این زبان، نحو^۲ ساده ای داشته، و می توان در عرض چند روز آن را به خوبی فرا گرفت. این زبان، برای توسعه برنامه های کاربردی دسکتاپ، وب و حتی نوشتن کتابخانه و سرویس های مختلف، کاربرد دارد. شما میتوانید به سادگی و با استفاده از این کتاب، این زبان را بیاموزید.

۱.۱ این کتاب برای چه کسانی است؟

به اختصار، باید گفت همه علاقمندان به یادگیری این زبان ساده و جذاب. جواب طولانی تر، اینست که این کتاب، برای کسانی است که واقعا میخواهند به سراغ روبی بروند، و چون روبی مرجع فارسی مطمئنی ندارد، پشیمان می شوند. پس اگر از هر دو دسته هستید، توصیه میکنم این کتاب را از دست ندهید.

۲.۱ آیا میتوانیم این کتاب را به اشتراک بگذاریم؟

این کتاب، رایگان عرضه شده است، همچنین شما نیز میتوانید «بدون قصد تجاری»، آن را با دوستان و خانواده خود به اشتراک بگذارید، یا برای دانلود در وبلاگ و وبسایت شخصی خود قرار دهید.

۳.۱ چگونه این کتاب را بخوانیم؟

برای خواندن این کتاب، دو راه دارید، یا صرفا مطالعه روزنامه وار انجام دهید تا به آشنایی اجمالی با نحو زبان روبی برسید، یا اینکه تک تک نرم افزارها و مراحل گفته شده در کتاب را مرحله به مرحله انجام دهید، تا کاملا به روبی مسلط شوید.

^۱Yukihiro Matsumoto

^۲Syntax

۲ پایه ها و مقدمات یادگیری روبی

در این فصل، چه فراخواهید گرفت؟

- نصب روبی در سیستم عامل
- آشنایی با محیط اینتراکتیو روبی
- ایجاد چند مثال ساده با روبی
- اجرای اسکریپت نوشته شده در روبی

این موارد، اولین قدم های شما در یادگیری زبان روبی هستند. در فصول بعدی، با ساختارها و کدهای بیشتری آشنا خواهید شد، و طبیعتا آن زمان است که میتوانید از روبی، استفاده کاربردی کنید.

۱.۲ نصب روبی

برای نصب روبی، شما ابتدا باید نام و نسخه سیستم عامل خود را بدانید. در این کتاب، فرض بر آنست که شما از گنو/لینوکس و توزیع اوبونتو استفاده میکنید.^۳

۱.۱.۲ وبسایت روبی

شما میتوانید با مراجعه به وبسایت روبی^۴ نسخه مورد نظر روبی را دانلود نموده، سپس آن را کامپایل کنید.^۵

۲.۱.۲ نصب از مخازن

برای استفاده از مخازن اوبونتو، کافیست تا یک پنجره ترمینال باز کنید، و سپس دستورات زیر را اجرا کنید :

```
sudo apt-get install ruby
```

همچنین اگر مفسر^۶ دیگری در نظر دارید، میتوانید در صورت موجود بودن در مخازن، آن را نصب کنید.

^۳ تفاوتی میانی استفاده از روبی در ویندوز و لینوکس نیست، طبیعتا میتوانید همین دستوراتی که در این کتابچه موجودند را به کار ببرید

^۴<http://ruby-lang.org>

^۵ وقتی روبی را از سورس کامپایل میکنید، باید پیشاپیش هر چه پیش نیاز آن است را دستی نصب کنید. چنانچه دانش کافی در زمینه کامپایل ندارید، توصیه میکنم از مخازن استفاده کنید.

^۶Interpreter

۲.۲ استفاده از محیط تعاملی

محیط تعاملی ^۷ روبی، به شما این امکان را میدهد تا بدون نیاز به نوشتن کد در یک ویرایشگر متن، آن را مستقیماً در ترمینال تایپ نموده، سپس نتیجه برنامه را در ترمینال ببینید. برای دسترسی به محیط تعاملی، یک پنجره ترمینال باز کرده و مانند دستور زیر عمل کنید :

```
prp-e@prp-e ~ $ irb
```

پس از اجرای دستور فوق، باید یک پرامپت ^۸ به این شکل مشاهده کنید :

```
irb(main):001:0>
```

درون این محیط، میتوانید دستورات مورد نظر را اجرا و تست کنید.

۳.۲ چند دستور در محیط تعاملی

مثال معروفی به نام «سلام دنیا» در تمامی زبان های برنامه نویسی وجود دارد. بیایید این مثال، اولین مثال ما نیز در زبان روبی باشد. اکنون، پنجره ترمینال را باز کرده، و همانند شکل زیر دستور را درون محیط تعاملی روبی، وارد کنید :

```
print "Hello, World\n"
```

در صورتی که کد را درست وارد کرده باشید، باید نتیجه به این شکل گرفته باشید :

```
irb(main):001:0> print "Hello, World\n"
Hello, World
=> nil
irb(main):002:0>
```

اما نکته اینجاست، در زبان روبی، اصولاً نیازی به دستوری برای ایجاد خط جدید نداریم. دستور مناسبتری وجود دارد که جایگزین `print` است و می توان به کمک آن، به خوبی داده ها را آنگونه که میخواهیم، چاپ کنیم. برای انجام کار عملی، دستورات زیر را در محیط تعاملی اجرا کنید :

```
puts "Hello, World"
```

اکنون باید نتیجه ای مشابه این دریافت کنید :

```
irb(main):002:0> puts "Hello, World"
Hello, World
=> nil
irb(main):003:0>
```

دستور `puts` جایگزین خوبی برای `print` است. علی الخصوص وقتی نتیجه مورد نظر، تنها یک خط باشد. اگرچه در آینده، از هر دو دستور استفاده خواهیم کرد.

^۷Interactive

^۸Prompt

۴.۲ نوشتن اسکریپت در یک فایل

روبی یک زبان اسکریپتی است^۹ پس میتوانیم یک اسکریپت بنویسیم و با استفاده از مفسر، آن را اجرا کنیم. یا حتی می توانیم به آن یک دسترسی اجرایی بدهیم، سپس مستقیماً اجراش کنیم.

۱.۴.۲ انتخاب ویرایشگر

شما برای آن که بتوانید کد خود را درون یک فایل روبی بنویسید، طبیعتاً به یک ویرایشگر متن^{۱۰} نیاز دارید. ویرایشگرهای متن که همراه سیستم عامل ها نصب می شوند، گزینه های خوبی هستند. اگر میخواهید بدانید من از چه استفاده میکنم، از nano به عنوان ویرایشگر تحت ترمینال، و از gedit به عنوان ویرایشگر گرافیکی استفاده میکنم.^{۱۱}

۲.۴.۲ پسوند فایل های روبی

سورس کدهای روبی، عموماً با پسوند های rb و یا ruby ساخته می شوند. توجه کنید که کامپیوتر به پسوند ها کاری ندارد، و این پسوند ها صرفاً برای این ساخته شده اند که انسان بتواند با استفاده از آنها، نوع فایل را تشخیص دهد.

۳.۴.۲ نوشتن و اجرای اسکریپت از طریق مفسر

مفسر روبی، از طریق خط فرمان با نام ruby در دسترس است، پس میتوانیم به سادگی یک قطعه کد بنویسیم، آن را روی دسکتاپ قرار داده و سپس دستورات مورد نظر را درونش وارد کنیم. اکنون، یک فایل خالی روی دسکتاپ اجرا کنید و این دستور را درونش بنویسید :

```
puts "Hello, World"
```

با فرض آن که این برنامه، روی دسکتاپ شما با نام HelloWorld.rb ذخیره شده است، آن را اینگونه اجرا میکنیم :

```
prp-e@prp-e ~ $ ruby ~/Desktop/HelloWorld.rb
```

سپس باید روی ترمینال چنین نتیجه ای را مشاهده کنیم :

```
prp-e@prp-e ~ $ ruby ~/Desktop/HelloWorld.rb
Hello, World
prp-e@prp-e ~ $
```

این اسکریپت، از طریق مفسر روبی، اجرا شد. در قسمت بعدی، از طریق اجازه ها^{۱۲} اسکریپت را اجرا خواهیم نمود.

^۹زبان های اسکریپتی، خط به خط اجرا می شوند. برخلاف زبانهای کامپایل شده، که ابتدا کل برنامه به یک زبان قابل فهم برای سیستم عامل یا ماشین ترجمه می شود، سپس اجرا میگردد.

^{۱۰}Text Editor

^{۱۱}در نظر داشته باشید چنانچه میزکار MATE را نصب دارید، ادیتور gedit با نام pluma در دسترس است.

^{۱۲}Permissions

۴.۴.۲ اجرای اسکریپت با استفاده از اجازه ها

این بار، از خاصیت اجازه دسترسی در یونیکس، استفاده خواهیم کرد و سپس اسکریپت قسمت قبل را مستقلاً اجرا خواهیم کرد. اکنون، نام HelloWorld.rb را به HelloWorld تغییر دهید و سپس با مفسر اجرای کنید :

```
prp-e@prp-e ~ $ ruby ~/Desktop/HelloWorld
Hello, World
prp-e@prp-e ~ $
```

نتیجه تفاوتی نکرد، نباید هم بکند. مفسر دستورات خودش را تشخیص میدهد. حالا میخواهیم مفسر را درون خود برنامه جای دهیم!
کد برنامه را به این شکل ویرایش کنید :

```
#!/usr/bin/ruby
```

```
puts "Hello, World"
```

اکنون، مفسر را به اسکریپت خود شناسانده ایم. اکنون وقت آن است که اجازه اجرا به سورس کد خود بدهیم. اکنون فقط کافیه تا این دستور را در ترمینال اجرا کنید :

```
prp-e@prp-e ~ $ chmod +x Desktop/HelloWorld
```

سپس میتوان این کد را مستقیماً اجرا کرد. برای اجرای مستقیم کد کافیه دستور زیر را اجرا کنید :

```
prp-e@prp-e ~ $ Desktop/HelloWorld
```

نتیجه دریافتی روی ترمینال باید به این شکل باشد :

```
prp-e@prp-e ~ $ Desktop/HelloWorld
Hello, World
prp-e@prp-e ~ $
```

۵.۲ جمع بندی

در اینجا، باید یک جمع بندی نهایی از مباحثی که در فصل مطرح شدند، داشته باشیم. اول اینکه، فراگیری روبی بسیار آسان است، ولی باید بدانید کجا و چگونه، چه دستوری را به کار ببرید. دوم، باید بدانید که چه موقع محیط تعاملی، و چه موقع محیط ویرایشگر را برای نوشتن برنامه انتخاب کنید. در نهایت هم همیشه حواستان به چگونگی نوشتن کد در ویرایشگر باشد، چرا که کوچک ترین اشتباهی، میتواند مانع از اجرای درست برنامه شما باشد.

۶.۲ تمرینات

همانگونه که در فروم اوپونتو قول داده بودم^{۱۳}، در پایان هر فصل، تعدادی تمرین خواهم گنجانم. با انجام این تمرینات، به سادگی میتوانید دانش و توانایی خود در زمینه روبی را بسنجید.

۱. برنامه ای بنویسید که نام خودتان را چاپ کند

۲. برنامه ای بنویسید که نام دانش آموزان یک کلاس ۱۵ نفره را چاپ کند، و آن را در قالب یک اسکریپت با اجازه اجرا، اجرا کنید

^{۱۳}انجمن های فارسی اوپونتو، در آدرس forum.ubuntu.ir در دسترس هستند. قول این کتاب را در تاپیکی با مضمون آموزش روبی، در آن انجمن داده بودم.

۳ انواع داده و متغیرها

در هر زبان برنامه نویسی، چندین نوع داده مختلف وجود دارد. استفاده از انواع مختلف داده، کار شخص برنامه نویس و حتی خود برنامه را آسان میکند. چرا که اینگونه، ماشین میفهمد که باید چگونه عملیاتی را روی داده انجام دهد. در این فصل، قصد داریم چندین نوع داده و همچنین تعریف آنها به عنوان متغیر را در روبی بررسی کنیم.

۱.۳ در این فصل چه فرا خواهید گرفت؟

- شناسایی انواع مختلف داده ها
- تعریف انواع مختلف داده در متغیر
- تبدیل انواع داده

۲.۳ داده های عددی

به طور کل، این نوع داده ها، اولین داده هایی هستند که در آموزش هر زبان برنامه نویسی، با آنها کار خواهیم داشت. چرا که گاهی با مقایسه و یا کم و زیاد کردن اعداد، کارهای خاصی در روند یک برنامه انجام میشود که میتواند به سرعت اجرای برنامه، کمک کند.

۱.۲.۳ اعداد صحیح

اعداد صحیح^{۱۴}، نوعی از اعداد هستند که جزء اعشاری ندارند. دامنه این اعداد در طبیعت از منفی بی نهایت تا مثبت بی نهایت ادامه دارد، با این تفاوت از اعداد حقیقی، که این دسته از اعداد، گسسته هستند. برای مثال در فاصله ۲ تا ۳ در مجموعه اعداد صحیح، هیچ عدد دیگری قرار نمیگیرد، در حالی که در سایر گونه های اعداد، اینگونه نیست. اگر چه اینجا بحث، بحث ریاضیات نیست، ولی بهتر بود توضیح ساده ریاضی اعداد صحیح را در اینجا ذکر میکردم. حالا نوبت آن است تا در محیط تعاملی روبی، چند عدد صحیح را تایپ کنید. برای تایپ کردن کافیسست با دستور `irb` وارد محیط تعاملی شده و اعداد زیر را تایپ کنید :

```
2
3
7
```

باید چنین خروجی هم دریافت کنید :

```
irb(main):001:0> 2
=> 2
irb(main):002:0> 3
=> 3
```

^{۱۴}Integer

```
irb(main):003:0> 7
=> 7
irb(main):004:0>
```

البته میتوانید هر کدام از اعداد را به دلخواه، منفی کنید. مهم اینست که در این قسمت، شما با نوع داده صحیح آشنا شدید.

۲.۲.۳ اعداد ممیز شناور

اعداد ممیز شناور^{۱۵}، نوعی از اعداد هستند که بخش اعشاری دارند. در واقع، این اعداد در برنامه نویسی، همان اعداد حقیقی هستند. اگرچه، اعداد گویا، اصم و گنگ را نیز میتوان انواع دیگری از داده حساب نمود، اما در برنامه نویسی، همه اعداد غیر صحیح در مبنای ده را به نوعی میتوان ممیز شناور به حساب آورد.

برای تعریف این اعداد، کافیسیت به سادگی یک ممیز و یک صفر به اعداد معرفی شده در قسمت بالاتر، اضافه نمایید :

```
2.0
3.0
7.0
```

و شاهد چنین خروجی باشید :

```
2irb(main):005:0> 2.0
=> 2.0
irb(main):006:0> 3.0
=> 3.0
irb(main):007:0> 7.0
=> 7.0
irb(main):008:0>
```

همانگونه که ملاحظه میفرمایید، مفسر ممیز را لحاظ کرده است. پس طبیعتاً بین ۲ و ۲.۰ تفاوت قائل می شود. اکنون با دو نوع اصلی و عمده داده های عددی آشنا شده اید، بیاییم با دو نوع داده عددی دیگر نیز آشنا شویم.

۳.۲.۳ اعداد اوکتال

اعداد اوکتال^{۱۶} اعدادی هستند که از مبنای ده، به مبنای هشت برده شده اند. روبی، زبانی است که قابلیت تشخیص مبنای هشت و شانزده را دارد. برای این که این اعداد را به محیط تعاملی روبی بدهیم، کافیسیت با دستور ورود آنها، و رنجی از اعداد که پشتیبانی میکنند، آشنا باشیم.^{۱۷} اعداد اوکتال در روبی، با صفر آغاز می شوند^{۱۸} پس اکنون کافیسیت محیط تعاملی روبی را باز کرده و اعداد

^{۱۵}Floating Point

^{۱۶}Octal

^{۱۷}اعداد اوکتال یا مبنای هشت، اعدادی هستند که از ارقام صفر تا هفت تشکیل شده اند
^{۱۸}این یک قرار داد است که این اعداد، با صفر آغاز شوند.

زیر را وارد کنید :

012
02
013457

خب اکنون باید نتیجه زیر را شاهد باشید :

```
irb(main):001:0> 012  
=> 10  
irb(main):002:0> 02  
=> 2  
irb(main):003:0> 013457  
=> 5935  
irb(main):004:0>
```

احتمالا بعد از دیدن این نتایج تعجب کرده اید، و پیش خودتان حدس هایی زده اید. باید بگویم درست حدس زده اید! روبی این اعداد را به مبنای ده برده و برای شما، نشان داده است. روبی نوع دیگری از داده های در مبنای غیر ده نیز دارد که در قسمت بعدی با آن آشنا خواهید شد.

۴.۲.۳ اعداد هگزادسیمال

اعداد هگزادسیمال^{۱۹} در مبنای ۱۶ هستند. این اعداد، در علوم کامپیوتر بسیار پر استفاده اند و در دروسی مانند مبانی برنامه نویسی و ساختمان داده، بسیار به آنها پرداخته می شود.^{۲۰} برای تعریف این اعداد در روبی، عدد مورد نظر را پس از صفر و x در محیط تعاملی وارد میکنیم. برای امتحان، اعداد زیر را در محیط تعاملی وارد کنید :

0x12
0x2
0x13457

سپس باید شاهد چنین نتیجه ای باشید :

```
irb(main):004:0> 0x12  
=> 18  
irb(main):005:0> 0x2  
=> 2  
irb(main):006:0> 0x13457  
=> 78935  
irb(main):007:0>
```

^{۱۹}Hexadecimal

^{۲۰}در اینجا، ارقام ۰ تا ۹ را به علاوه ارقام A ، B ، C ، D و E داریم که نمایانگر ۱۰ تا ۱۵ هستند.

این بار هم شاهد این بودید که مفسر روبی، اعداد را از مبنای شانزده به ده برده و به شما نمایش داده است. در ادامه، با انواع دیگری از متغیر ها آشنا خواهید شد، که دیگر عددی نیستند.

۳.۳ متغیرهای بولین

متغیر های بولین ^{۲۱}، متغیر هایی هستند که تنها دو مقدار درست (true) و یا غلط (false) دریافت میکنند.

۴.۳ متغیر های رشته ای

یک بار دیگر، برنامه زیر را چک کنید :

```
puts "Hello, World"
```

اگر دقت کنید، خواهید دید که عبارت Hello, World را درون علامت نقل قول قرار داده ایم. این نوع داده را، رشته ^{۲۲} میگویند. هر رشته، از بهم پیوستن چندین کاراکتر ^{۲۳} تشکیل می شود. یک رشته، میتواند شامل چند خط باشد، یا شامل یک کلمه باشد. در کل، قانون و قاعده خاصی برای تعریف مقادیر درون یک متغیر رشته ای، وجود ندارد.

۵.۳ آرایه ها و هش ها

آرایه ها ^{۲۴} و هش ها ^{۲۵} دو نوع متغیر مجموعه ای هستند. آرایه، مجموعه ای از متغیر های یک نوع خاص است، و هش مانند یک دیکشنری عمل می کند. با عملکرد این دو نوع متغیر، در آینده آشنا خواهید شد.

۶.۳ تعریف متغیرها

بر خلاف زبان هایی مانند C و C++ که ابتدا نوع، سپس نام و سپس مقدار متغیر تعریف میگردد، در روبی تنها نام متغیر کافیهست تا به آن مقدار داده شود. دستورات زیر، چند متغیر عددی و رشته ای را در روبی تعریف میکنند :

```
name = "Muhammadreza"  
a = "2"  
b = 3
```

متغیر های name و a از نوع رشته ای هستند (به علامت نقل قول توجه کنید) ، و متغیر b از نوع عدد صحیح. نکته اینجاست که متغیرها، امکان تبدیل به یک دیگر را نیز دارند.

^{۲۱}Boolean

^{۲۲}String

^{۲۳}Character

^{۲۴}Array

^{۲۵}Hash

۱.۶.۳ تبدیل نوع متغیر ها

متغیر a از نوع رشته ای، و متغیر b از نوع عدد صحیح است. برای تبدیل a به عدد صحیح، و برای تبدیل b به ممیز شناور، از دستورات زیر استفاده میکنیم :

`a.to_i`

`b.to_f`

و برای ذخیره نتایج، کافیسست متغیر جدیدی تعریف کرده، و دستورات را درونش بریزید. اگرچه، تغییر نوع های دیگری نیز وجود دارد که در ادامه به آنها خواهیم پرداخت.

۷.۳ جمع بندی

در این فصل، آموختید که هر نوع داده ای کجا و چگونه به کار میرود. مهم است که بدانید کجا و چطور، از چه نوع داده ای استفاده کنید. مثلاً موقعی که خرده داشتن نتیجه، برایتان مهم نیست، منطقی تر است حتی اگر ورودی ممیز شناور باشد، شما خروجی را به عدد صحیح تبدیل کنید. اما اگر جایی قرار است مثلاً تا n رقم اعشار محاسبه شود، منطقی تر آن است که حتی ورودی های صحیح، در خروجی به صورت ممیز شناور نمایش داده شوند. برخی از انواع داده نیز معرفی شدند، اما مثالی از آنها زده نشد. این نوع داده ها، هر کدام خودشان یک فصل جداگانه میطلبیدند، پس نگران نباشید، این ها را در ادامه فراخواهید گرفت.

۸.۳ تمرینات

این فصل، آموزش خاصی به همراه نداشت، در نتیجه تمریناتی در این فصل گنجانده نشده است. اما در فصل آینده و پس از آشنایی با عملگرها، تمرینات فراوانی را فرا خواهید گرفت.