برنامه نویسی به زبان

روبی

مرجع کامل

تالیف
مهندس هادی کیامرثی

تمام مثال های موجود در این کتاب با کامپیوتر تست شده اند تا از هر گونه خطا مبرا باشند با این حال ممکن است باز هم خطاهایی در آن وجود داشته باشد از کلیه خوانندگان این کتاب ، اساتید و دانشجویان محترم خواهشمندم برای مطلع کردن مولف (هادی کیامرثی) از این خطا ها لطفا با ایمیل آدرس زیر تماس بگیرند

hadikiamarsi@gmail.com

فصل اول

در این فصل مطالب زیر را خواهید آموخت

روبی (ruby) چیست ؟
تاریخچه روبی (ruby)
کلمات کلیدی در روبی (ruby)
بدست آوردن آرگومان های مفسر روبی (ruby)
بدست آوردن نسخه (version) مفسر روبی (ruby)
برنامه نویسی در خط فرمان
برنامه نویسی در محیط تعاملی (Interactive)

برنامه نویسی در حالت متنی

پسوند فایل های روبی (ruby)

روبی (ruby) زبانی حساس به بزرگی و کوچکی حروف
انواع اعداد در روبی (ruby)
متغیرها (Variables)
تعریف متغیرها (variables)
مقدار دادن به متغیرها
متغیرهای خاص
تعیین نوع متغیر
ثابت ها
تعریف ثابت ها
داده بی ارزش
عملگرها (operators)

عملگرهای محاسباتی در روبی (ruby)

عملگرهای مقایسه ای در روبی (ruby)

عملگرهای انتساب در روبی (ruby)

عملگرهای بیتی

عملگرهای منطقی در روبی (ruby)

عملگرهای بازه (range) در روبی (ruby)

عملگرهای رشته ای در روبی (ruby)

عملگر defined? در روبی (Ruby)

اولویت عملگرها در روبی (ruby)

کاراکتر جای خالی (space) در روبی (ruby)
توضیحات (comment) در روبی (ruby)
دستورات ورودی
دستورات خروجی
انتهای کد در زبان برنامه نویسی روبی (ruby)
پاک کردن صفحه خروجی
ساختار یک برنامه روبی (ruby)
تفاوت علامت های نقل قول تکی و جفتی
علامت ;
متغیرهای عمومی (global)
تبدیل اعداد از یک مبنا به مبنای دیگر در روبی (ruby)
تبدیل انواع (Type Casting)

روبی (ruby) چیست ؟

روبی (ruby) یک زبان برنامه نویسی (Language Programming) اسکریپتی ، همه منظوره ، شی گرا (object-oriented) ، و سطح بالا (high-level programming language) می باشد . زبان های برنامه نویسی به دو دسته تقسیم می گردند یا کامپایلر دارند یا مفسر (interpreter) . در زبان هایی که کامپایلر دارند تمام کد یکجا به زبان ماشین تبدیل می شود ولی در زبان هایی که مفسر (interpreter) دارند کد برنامه خط به خط به کد ماشین تبدیل می شود . زبان برنامه نویسی روبی (ruby) یک زبان مفسری می باشد به عبارت دیگر مفسر (interpreter) دارد .

تاریخچه روبی (ruby)

روبی (ruby) یک زبان برنامه نویسی تمام شی گرا (object-oriented) می باشد که در سال 1993 توسط یوکیهیرو ماتسوموتو (Yukihiro Matsumoto) از ژاپن ساخته شد . روبی (ruby) به صورت اوپن سورس (open-source) منتشر شده است به این معنا که کد منبع آن در اختیار همه هست و هرکسی ی تواند کد منبع آن را مطالعه نماید . روبی (ruby) مشخصاتی دارد که از زبان های پرل (perl) و پایتون (python) وام گرفته است ولی در کل بیشتر ویژگی های زبان برنامه نویسی روبی (ruby) از زبان برنامه نویسی اسمال تالک (Smalltalk) برگرفته شده است . و از لحاظ دستوری نیاز بسیار شبیه زبان برنامه نویسی اسمال تالک (Smalltalk) می باشد ولی باید بدانید برنامه نویسی به زبان برنامه نویسی روبی (ruby) بسیار راحت تر از برنامه نویسی به زبان برنامه نویسی اسمال تالک (Smalltalk) می باشد . زبان برنامه نویسی روبی (ruby) بر روی تعداد بیشماری از سیستم عامل ها نظیر ویندوز (Windows) ، مک او اس (Mac OS) و سیستم عامل های یونیکس (UNIX) بیس مانند لینوس (Linux) ، فری بی اس دی (FreeBSD) اجرا می گردد . زبان برنامه نویسی روبی (ruby) کاربرد های فراوانی در زمینه های

مدیریت سیستم
پردازش متن
برنامه نویسی سمت سرور (server-side)
برنامه نویسی شبکه و سوکت (socket)
برنامه نویسی اینترنت (Internet and intranet applications)
ارتباط با پایگاه داده
برنامه نویسی سی جی آی (CGI) که اختصار (Common Gateway Interface) هست

امروزه زبان برنامه نویسی روبی (ruby) جای خود را بسیار بین برنامه نویسان حوزه وب (web) باز کرده است . روز به روز بر حوزه های کاربردی زبان برنامه نویسی روبی (ruby) افزوده می گردد .

کلمات کلیدی در روبی (ruby)

BEGIN	Do	next	Then
END	Else	nil	True
alias	Elsif	not	Undef
and	End	or	Unless
begin	Ensure	redo	Until
break	False	rescue	When
case	For	retry	While
class	If	return	While
def	In	self	__FILE__
defined?	Module	super	__LINE__

بدست آوردن آرگومان های مفسر روبی (ruby)

برای بدست آوردن تمام آرگومان های مفسر (interpreter) زبان برنامه نویسی روبی (ruby) باید دستور زیر را در محیط متنی سیستم عامل خود وارد نمایید

```
ruby -h
```

بدست آوردن نسخه (version) مفسر روبی (ruby)

برای دریافت نسخه زبان برنامه نویسی روبی (ruby) باید دستور زیر را در محیط متنی سیستم عامل خود وارد نمایید

```
ruby -v
```

برنامه نویسی در خط فرمان

مفسر (interpreter) زبان برنامه نویسی روبی (ruby) دارای آرگومان های خط فرمانی می باشد که بوسیله آن ها می توان به صورت مستقیم در خط فرمان کدنویسی کرد برای کد نویسی مستقیم در خط فرمان (محیط متنی سیستم عامل) باید از آرگومان -e استفاده نمایید

برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
ruby –e ‘print(“Hadi Kiamarsi\n”)’
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Hadi Kiamarsi
```

برنامه نویسی در محیط تعاملی (Interactive)

محیط تعاملی محیطی می باشد که در آن می توان کد نوشت و همان لحظه نتیجه اجرای کدخود را ببینید برای وارد شدن به محیط تعاملی در زبان برنامه نویسی روبی (ruby) باید دستور irb را در محیط متنی سیستم عامل خود وارد نمایید در زیر مثالی از برنامه نویسی در محیط تعاملی (interactive) را می بینید

```
irb(main):001:0> 3 ** 3
=> 27
irb(main):001:0>
```

برای خروج از محیط تعاملی (interactive) در زبان برنامه نویسی روبی (ruby) باید دستور exit یا quit را وارد نمایید

برنامه نویسی در حالت متنی

شما می توانید دستورات زبان برنامه نویسی روبی (ruby) را در یک فایل نوشته و آن را اجرا نمایید به این فایل که حاوی کد های زبان برنامه نویسی روبی (ruby) می باشد اسکریپت (script) گفته می شود . برای مثال دستورات زیر را با استفاده از یک ویرایشگر فایل (در سیستم عامل های یونیکس بیس مانند لینوکس از ویرایش گرهای vi یا vim یا nano می توانید استفاده نمایید) در فایل hello.rb نوشته و ذخیره نمایید .

```
#!/usr/bin/ruby

# This will print "Hello, World"
```

```
print "Hello, world\n";
```

این قسمت از کد **/usr/bin/ruby** مسیر فایل اجرایی روبی را نشان می دهد . حال برای اجرای کد موجود در فایل hello.rb ابتدا باید سطح دسترسی به فایل را به 0755 تغییر دهید سپس دستورات را اجرا نمایید برای اجرا دستورات زیر را در خط فرمان لینوکس اجرا نمایید .

```
$chmod 0755 hello.rb
$./hello.rb
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Hello, world
```

برای اجرای فایل hello.rb در سیستم عامل ویندوز از دستور زیر استفاده می نماییم

```
$./hello.rb
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Hello, world
```

در زبان برنامه نویسی روبی (ruby) برای استفاده از توابع هم می توانید از پرانتز استفاده نمایید و هم می توانید بدون پرانتز آن ها را بکار ببرید برای روشن شدن موضوع به مثال زیر توجه نمایید هر دو خط زیر نتیجه یکسانی را ایجاد می نمایند

```
print("Hello, world\n");
print "Hello, world\n";
```

پسوند فایل های روبی (ruby)

یک فایل که حاوی متن کدهای روبی (ruby) می باشد یک اسکریپت (script) نامیده می شود و می تواند توسط هر نرم افزار ویرایش متنی که در سیستم عامل شما موجود است ایجاد گردد فقط نکته حائذ اهمیت در این قضیه این می باشد که پسوند این فایل باید حتما rb باشد . لازم به ذکر است که در نام فایل حاوی کدهای روبی (ruby) نباید فاصله (space) بکار رفته باشد .

روبی (ruby) زبانی حساس به بزرگی و کوچکی حروف

زبان برنامه نویسی روبی (ruby) به بزرگی و کوچکی حروف حساس می باشد به عنوان نمونه true با True تفاوت دارد

انواع اعداد در روبی (ruby)

```
123                     # اعداد دسیمال
1_234                   # اعداد دسیمال
-500                    # اعداد منفی
0377                    # اعداد اکتال
0xff                    # اعداد هگزادسیمال
0b1011                  # اعداد باینری
123456789012345678901   # اعداد خیلی بزرگ


123.4                   # اعداد اعشاری
1.0e6                   # اعداد علمی اعشاری
4E20                    # اعداد علمی
4e+20                   # اعداد علمی
```

متغیرها (Variables)

متغیر (variables) نامی است که به یک خانه از حافظه اشاره دارد که در آن مقادیری ذخیره می گردد .

تعریف متغیرها (variables)

برای تعریف متغیرها در زبان برنامه نویسی روبی (ruby) باید نامی را برای آن برگزینیم . برای انتخاب نام برای تعریف متغیر ها در زبان برنامه نویسی روبی (ruby) باید قوانین زیر را رعایت نمایید

1) نام یک متغیر نمی تواند با اعداد شروع گردد
2) نام یک متغیر نمی تواند با کاراکتر فضای خالی شروع گردد
3) نام متغیرها در روبی (ruby) باید حتما با حروف کوچک شروع گردد

برای آشنایی با نحوه تعریف متغیرها به مثال زیر توجه نمایید

```
var1 = 12
var2 = ‘hello’
var3 = 2.5
```

مقدار دادن به متغیرها

در زبان برنامه نویسی روبی (ruby) برای تعیین مقدار برای متغیرها از کاراکتر (=) استفاده می گردد
برای آشنایی با نحوه تعیین مقدار برای متغیرها به مثال زیر توجه نمایید

```
var1 = 12
var2 = ‘hello’
var3 = 2.5
```

در زبان برنامه نویسی روبی (ruby) امکان تعیین مقدار چندگانه نیز برای متغیرها وجود دارد برای نحوه انجام این کار به مثال زیر توجه نمایید

```
a=b=c=12
```

مثالی دیگر

```
a,b,c=24,’hello’,”world”
```

متغیرهای خاص

در زبان برنامه نویسی روبی (ruby) بعضی متغیرهایی وجود دارند که از قبل توسط مفسر (interpreter) روبی (ruby) مقداردهی شده اند از انواع این متغیرها می توان به __FILE__ و __LINE__ اشاره کرد که اولی نام اسکریپت جاری در حال اجرا را نشان می دهد و دومی خطی که در آن نوشته شده است را بر می گرداند برای آشنایی بیشتر با نحوه کاربرد و طریقه استفاده این متغیرها به مثال زیر توجه نمایید

```
print __FILE__
print __LINE__
```

اجرای دستورات بالا نتایج زیر را در صفحه خروجی نشان خواهد داد

```
test.rb
5
```

تعیین نوع متغیر

در برنامه نویسی بخصوص در برنامه نویسی برنامه های بزرگ موقعیت هایی پیش می آید که برنامه نویس نیاز دارد بداند یک متغیر از چه نوعی می باشد برای این کار در زبان برنامه نویسی روبی (ruby) ازکلمه کلیدی class استفاده می گردد برای آشنایی با نحوه کاربرد کلمه کلیدی class به مثال زیر توجه نمایید

```
a = ‘salam’
b = 12
print a.class
print “\n”
print b.class
```

اجرای دستورات بالا نتایج زیر را در صفحه خروجی نشان خواهد داد

```
String
Integer
```

ثابت ها

ثابت ها همانطور که از نامشان پیداست مقدارشان تا انتهای برنامه تغییر نمی کند

تعریف ثابت ها

تعریف ثابت ها از همان قوانین تغریف متغیرها پیروی می کند با این تفاوت که برای تعریف ثابت ها باید در ابتدای آن ها از حروف بزرگ استفاده نمایید. برای آشنایی با ثابت ها به مثال زیر توجه نمایید

```
Var1 = 200
Const = 300
```

داده بی ارزش

موقعیت هایی پیش می آید در برنامه نویسی که نیاز می باشد یک متغیر را به گونه ای تعریف نماییم که مقدار آن یک مقدار بی ارزش و بلا استفاده باشد برای این کار در زبان برنامه نویسی روبی (ruby) از کلمه کلیدی nil استفاده می نماییم نحوه تعریف متغیر با داده بی ارزش در مثال زیر نشان داده شده است

```
a = nil
```

عملگرها (operators)

برای تعریف یک عملگر باید از یک مثال استفاده نماییم عبارت 3 + 4 را در نظر بگیرید اعداد 3 و 4 را عملوند و علامت + را عملگر می نامیم . زبان برنامه نویسی روبی (ruby) از تعداد زیادی عملگر (operator) پشتیبانی می نماید که در زیر با آن ها آشنا می شوید

عملگرهای محاسباتی در روبی (ruby)

عملگر	نام	مثال
+	جمع	10 + 20 = 30
-	تفریق	10 - 20 = -10
*	ضرب	10 * 20 = 200
/	تقسیم	20 / 10 = 2
%	باقیمانده	20 % 10 = 0
**	توان	3 ** 2 =9

عملگرهای مقایسه ای در روبی (ruby)

عملگر	نام	مثال
==	مساوی	برابر نیست با (10 == 20) true.
!=	نامساوی	برابراست با (10 != 20) true.
>	بزرگتر	برابر نیست با (10 > 20) true.
<	کوچکتر	برابراست با (10 < 20) true.
>=	بزرگتر مساوی	برابر نیست با (10 >= 20) true.
<=	کوچک تر مساوی	برابراست با (10 <= 20) true.
<=>	اگر عملوند اول با عملوند دوم برابر باشد عدد 0 را بر می گرداند اکر عملوند اول از عملوند دوم بزرگتر باشد عدد 1 اگر عملوند اول از عملوند دوم کوچکتر باشد عدد -1 را بر می گرداند	برمی گرداند (10 <=> 20) -1.
===	بررسی تساوی بر اساس شی	بر می گرداند (1...10) === 5 true. 1 === 1 = true

		Fixnum === Fixnum = false Fixnum === 1 = true
.eql?	بررسی تساوی	1 == 1.0 بر می گرداند true, ولی 1.eql?(1.0) برابر است با false.
Equal?	بررسی تساوی بر اساس شماره (id) شی	a = 'xyz' b = 'xyz' a.Equal? b = False b=a a.Equal? b = True

عملگرهای انتساب در روبی (ruby)

عملگر	مثال
=	c = a + b a + b قرار بگیرد در c
+=	c += a برابر است با c = c + a
-=	c -= a برابر است با c = c - a
*=	c *= a برابر است با c = c * a
/=	c /= a برابر است با c = c / a
%=	c %= a برابر است با c = c % a
**=	c **= a برابر است با c = c ** a

عملگرهای بیتی

عملگر	نام	مثال
&	و	(60 & 13) = 12
|	یا	(60 | 13) = 61
^	یای انحصاری	(60 ^ 13) = 49
~	نقیض	(~60) = -61
<<	شیفت چپ	60 << 2 = 240
>>	شیفت راست	60 >> 2 = 15

عملگرهای منطقی در روبی (ruby)

عملگر	نام	مثال
And	و	(true and true) = true.
Or	یا	(true or false) = true.
&&	و	(true && true) = true.
||	یا	(true || false) = true.
!	نقیض	!(true && true) = false.
Not	نقیض	not(true && true) = false.

عملگرهای بازه (range) در روبی (ruby)

عملگر	توضیح	مثال
..	یک بازه (range) از اعداد از اولین عدد تا آخرین عدد ایجاد می نماید	1..10 = 1 تا 10
...	یک بازه (range) از اعداد از اولین عدد تا یکی مانده به آخرین عدد ایجاد می نماید	1...10 = 1 تا 9.

عملگرهای رشته ای در روبی (ruby)

عملگر	نام	مثال
+	الحاق یا اتصال	'Hadi'+'Kiamarsi'='HadiKiamarsi'
*	تکرار	'a'*3='aaa'

عملگر defined? در روبی (Ruby)

از این عملگر جهت شناسایی نوع یک شی (object) یا اینکه از چه نوعی (متغیر ، متد ، تابع ، کلاس و ...) توسط برنامه نویس تعریف شده است در کد برنامه استفاده می گردد برای آشنایی با کاربرد های این عملگر به مثال های زیر توجه نمایید

کاربرد 1

```
defined? variable # True if variable is initialized
```

مثال

```
foo = 42
defined? foo
defined? $_
defined? bar
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
local-variable

global-variable

nil (undefined)
```

در مثال بالا نوع شی ها مشخص شده است

کاربرد 2

```
defined? method_call # True if a method is defined
```

مثال

```
defined? puts        # => "method"
defined? puts(bar)   # => nil (bar is not defined here)
defined? unpack      # => nil (not defined here)
```

کاربرد 3

```
# True if a method exists that can be called with super user
defined? super
```

مثال

```
defined? super     # => "super" (if it can be called)
defined? super     # => nil (if it cannot be)
```

کاربرد 4

```
defined? yield   # True if a code block has been passed
```

مثال

```
defined? yield    # => "yield" (if there is a block passed)
defined? yield    # => nil (if there is no block)
```

اولویت عملگرها در روبی (ruby)

در زبان های برنامه نویسی ترتیب اجرای عملگر ها بر اساس اولویت آنها تعیین می گردد
در زیر اولویت عملگرها در زبان برنامه نویسی روبی (ruby) آورده شده است

عملگر
::
[ ] [ ]=
**
! ~ + -
* / %
+ -
>> <<
&
^ |
<= < > >=
<=> == === != =~ !~
&&
||
.. ...
? :
= %= { /= -= += |= &= >>= <<= *= &&= ||= **=
defined?
Not
or and

کاراکتر جای خالی (space) در روبی (ruby)

در زبان برنامه نویسی روبی (ruby) کاراکتر جای خالی (space) و تب (tab) به جز در رشته ها در جاهای دیگر کد نادیده گرفته می شوند برای روشن شدن بیشتر این درس به مثال زیر توجه نمایید

```
a + b
a+b
```

در مثال بالا هر دو خط یکی هستند

توضیحات (comment) در روبی (ruby)

توضیحات در هر زبان برنامه نویسی موجب خوانا تر شدن بیشتر کد برنامه می شود و بخصوص برای برنامه نویسان بسیار مفید می باشد . در زبان برنامه نویسی روبی (ruby) دو نمونه توضیحات (comment) وجود دارد

1) توضیحات تک خطی
2) توضیحات چند خطی

برای ایجاد توضیحات (comment) تک خطی در زبان برنامه نویسی روبی (ruby) از علامت # استفاده می گردد برای بهتر متوجه شدن این درس به مثال زیر توجه نمایید

```
# I am a comment. Just ignore me.
```

مثالی دیگر

```
name = "Madisetti" # This is again comment
```

مثالی دیگر چند خط توضیح با استفاده از #

```
# This is a comment.
# This is a comment, too.
# This is a comment, too.
# I said that already.
```

برای ایجاد توضیحات (comment) چند خطی در زبان برنامه نویسی روبی (ruby) از علامت =begin و =end استفاده می گردد برای بهتر متوجه شدن این درس به مثال زیر توجه نمایید در مثال زیر نمونه ای از توضیحات (comment) چند خطی نشان داده شده است

```
=begin
This is a comment.
This is a comment, too.
This is a comment, too.
I said that already.
=end
```

دستورات ورودی

برای دریافت اطلاعات از ورودی صفحه کلید در زبان برنامه نویسی روبی (ruby) از دستور gets.chomp استفاده می گردد برای آشنایی با نحوه استفاده از این دستور به مثال زیر توجه نمایید

```
print “Please Enter String : ”
str1 = gets.chomp
print str1
```

اجرای دستورات بالا نتایج زیر را در صفحه خروجی نشان خواهد داد

```
Please Enter String : Hello World!
Hello World!
```

دستورات خروجی

در زبان برنامه نویسی روبی (ruby) برای چاپ اطلاعات بر روی صفحه نمایش از تابع print و puts استفاده می گردد . برای آشنایی با کاربرد این توابع به مثال های زیر توجه نمایید

```
print ‘Hello World!’
print "Hello","World!"
print 'Hello','World','!'
```

اجرای دستورات بالا نتایج زیر را در صفحه خروجی نشان خواهد داد

```
Hello World!
Hello World!
Hello World!
```

مثالی دیگر

```
#!/usr/bin/ruby -w

print <<EOF
   This is the first way of creating
   here document ie. multiple line string.
EOF
```

```
print <<"EOF";                # same as above
   This is the second way of creating
   here document ie. multiple line string.
EOF

print <<`EOC`                 # execute commands
       echo hi there
       echo lo there
EOC

print <<"foo", <<"bar"  # you can stack them
       I said foo.
foo
       I said bar.
bar
```

اجرای دستورات بالا نتایج زیر را در صفحه خروجی نشان خواهد داد

```
   This is the first way of creating
   her document ie. multiple line string.
   This is the second way of creating
   her document ie. multiple line string.
hi there
lo there
      I said foo.
      I said bar.
```

مثالی دیگر

```
puts ‘hadi’,’kiamarsi’
```

اجرای دستورات بالا نتایج زیر را در صفحه خروجی نشان خواهد داد

```
hadi
kiamarsi
```

مثالی دیگر

```
puts “Hello \n”
puts ‘World!’
```

اجرای دستورات بالا نتایج زیر را در صفحه خروجی نشان خواهد داد

```
Hello World!
```

در زبان برنامه نویسی روبی (ruby) علامت هایی برای قالب بندی خروجی در صفحه نمایش وجود دارد که لیست آن ها را در جدول زیر می بینید

علامت	توضیح
\n	مکان نما را به خط بعد منتقل می نماید
\r	در سیستم عامل ویندوز برای انتقال مکان نما به خط جدید در فایل ها باید همراه ( ) بکار رود
\t	مکان نما را به هشت کاراکتر بعد منتقل می نماید
\’	نشان دادن نقل قول تکی در خروجی
\”	نشان دادن نقل قول جفتی در خروجی
\f	مکان نما را به خط بعد و هشت کاراکتر جلوتر منتقل می نماید
\b	یک کاراکتر را پاک می کند و مکان نما را به عقب منتقل می نماید
\a	یک صدای بیپ از بلندگوی کامپیوتر پخش می نماید
\e	کاراکتر اسکیپ را در خروجی نشان می دهد
\s	یک کاراکتر فضای خالی در صفحه خروجی نشان می دهد
\nnn	برای نشان دادن اعداد در مبنای اکتال در خروجی بکار می رود
\xnn	برای نشان دادن اعداد در مبنای هگزال در خروجی بکار می رود
\cx, \C-x	کاراکتر کنترل به همراه ایکس را در خروجی نشان می دهد
\x	کاراکتر ایکس را در خروجی نشان می دهد

در زبان برنامه نویسی روبی (ruby) دستوری دیگری نیز برای نمایش اطلاعات درخروجی وجود دارد و این دستور printf می باشد این دستور از زبان برنامه نویسی C و C++ وام گرفته شده است و در روبی (ruby) گنجانده شده است برای استفاده از این دستور باید مقادیر جدول زیر را یاد بگیرید

علامت	توضیح
%s	برای نمایش رشته ها بکار می رود
%d	برای نمایش اعداد دسیمال بکار می رود
%f	برای نمایش اعداد اعشاری بکار می رود

برای آشنایی با نحوه کاربرد این دستور به مثال زیر توجه نمایید

```
printf(“%s”,’Hello’)
a=12
printf(“Number is : %d\n”,a)
printf(“%d%s”,a,’Hello’)
```

اجرای دستورات بالا نتایج زیر را در صفحه خروجی نشان خواهد داد

```
Hello
Number is : 12
12 Hello
```

انتهای کد در زبان برنامه نویسی روبی (ruby)

در انتهای خطوط کد در زبان برنامه نویسی روبی (ruby) هم می تواند علامت سمی کولن (;) قرار بگیرد هم می تواند هیچ چیز قرار نگیرد به مثال زیر توجه نمایید

```
a = " Hello ";
b = "World!\n";
print a
print b
```

پاک کردن صفحه خروجی

برای پاک کردن صفحه خروجی در زبان برنامه نویسی روبی (ruby) از دستور زیر استفاده می گردد .

```
System("clear") || system("cls")
```

ساختار یک برنامه روبی (ruby)

هر برنامه روبی با یک خط کد راهنمای مفسر (interpreter) شروع می شود که در زیر نشان داده شده است

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

و در نهایت ساختار یک برنامه روبی به صورت زیر می باشد

```
#!/usr/bin/ruby

در این قسمت کد اصلی برنامه نوشته می شود
```

در برنامه نویسی موقعیت هایی پیش می آید که نیاز داریم قطعه کدی پیش از اجرای هر کد دیگری اجرا گردد در زبان برنامه نویسی روبی (ruby) این کار بوسیله بلاک کد زیر انجام می پذیرد

```
BEGIN {
   code
}
```

برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/ruby

puts "This is main Ruby Program"

BEGIN {
   puts "Initializing Ruby Program"
}
```

بعد از اجرای کد بالا نتیجه زیر ظاهر خواهد گردید

```
Initializing Ruby Program
This is main Ruby Program
```

همچنین در برنامه نویسی موقعیت هایی پیش می آید که نیاز داریم قطعه کدی در انتهای اجرای همه کدهای دیگر اجرا گردد در زبان برنامه نویسی روبی (ruby) این کار بوسیله بلاک کد زیر انجام می پذیرد

```
END {
   code
}
```

برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/ruby

puts "This is main Ruby Program"

END {
   puts "Terminating Ruby Program"
}
BEGIN {
   puts "Initializing Ruby Program"
}
```

بعد از اجرای کد بالا نتیجه زیر ظاهر خواهد گردید

```
Initializing Ruby Program
This is main Ruby Program
```

```
Terminating Ruby Program
```

تفاوت علامت های نقل قول تکی و جفتی

در زبان برنامه نویسی روبی (ruby) شما می توانید برای تعریف یک رشته هم از علامت نقل قول تکی (single quotes) و هم نقل قول جفتی (double quotes) استفاده نمایید

```
#!/usr/bin/ruby

print "Hello, world\n";
print 'Hello, world\n';
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Hello, world
Hello, world\n
```

در اینجا یک تفاوت خیلی مهم در استفاده از علامت نقل قول تکی و نقل قول جفتی وجود دارد اگر نام متغیری در بین علامت نقل قول جفتی به همراه علامت (#{}) قرار گیرد مقدار متغیر مد نظر قرار می گیرد در صورتی که اگر متغیری در بین علامت نقل قول تکی قرار گیرد نام خود متغیر مد نظر قرار می گیرد . برای روشن شدن بیشتر این درس لطفا به مثال زیر توجه نمایید

```
#!/usr/bin/ruby

a = 10;
print "Value of a = #{a}\n";
print "Value of a = a\n";
print 'Value of a = a\n';
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
Value of a = 10
Value of a = a
Value of a = a\n
```

علامت ;

اگر نیاز داشتید در زبان برنامه نویسی روبی (ruby) چند دستور را در یک خط بنویسید از علامت ; در بین دستورات استفاده می گردد برای آشنایی با کاربرد این علامت به مثال زیر توجه نمایید

```
$var1='hadi kiamarsi';print $var1
```

متغیرهای عمومی (global)

متغیرها به طور کلی به دو دسته محلی (خصوصی) و عمومی تقسیم می گردد متغیرهای محلی فقط در همان بلاک کدی که تعریف می شوند قابل دسترسی می باشند ولی متغیر های عمومی در تمام بلاک های برنامه قابل دسترسی هستند . برای تعریف متغیرهای عمومی در زبان برنامه نویسی روبی (ruby) از کاراکتر $ استفاده می گردد

```
#!/usr/bin/ruby

$a="Hello World!\n"

def test
   print $a
end

print $a
```

تبدیل انواع (Type Casting)

موقعیت هایی در برنامه نویسی پیش روی شما قرار می گیرد که نیاز می باشد نوعی از یک متغیر به نوعی دیگر تبدیل نمایید به عنوان نمونه یک متغیر از نوع عددی را به نوع رشته ای تبدیل نمایید هر چند زبان برنامه نویسی روبی (ruby) به صرت هوشمند خود توانایی تبدیل نوع یک متغیر به نوعی دیگر را دارد اما بهتر آن است برای اطمینان بیشتر خود به صورت کد این کار را انجام بدهیم

برای تبدیل عدد به رشته به صورت زیر عمل می نماییم

```
a = 12
print a.to_s
```

برای تبدیل رشته به عدد به صورت زیر عمل می نماییم

```
a = ‘12’
print a.to_i
```

تبدیل اعداد از یک مبنا به مبنای دیگر در روبی (ruby)

برای تبدیل اعداد در مبناهای مختلف در زبان برنامه نویسی روبی (ruby) از دستور sprintf استفاده می گردد که برای استفاده از آن به اطلاعات جدول زیر نیاز دارید

برای آشنایی بیشتر با این دستور به مثال زیر توجه نمایید

```
a = 5
b = sprintf(“%b”,a)
print b
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
101
```

در مثال بالا یک عدد دسیمال را به یک عدد باینری (عدد در مبنای 2) تبدیل کرده ایم
حال به صورت بر عکس نیز می توان این کار را انجام داد به مثال های زیر توجه نمایید

```
print sprintf(“%d”,0b101)
```

اجرای کد بالا نتیجه زیر را ظاهر خواهد نمود

```
5
```

حال به روش دیگری این کار را انجام خواهیم داد . به مثال زیر توجه نمایید

```
print “5”.to_s(2)
print “\n”
print 0b101.to_i
print “\n”
print “0a”.to_i(16)
print “\n”
print “101”.to_i(2)
```

حال به روش دیگری این کار را انجام خواهیم داد . به مثال زیر توجه نمایید

```
101
5
10
5
```