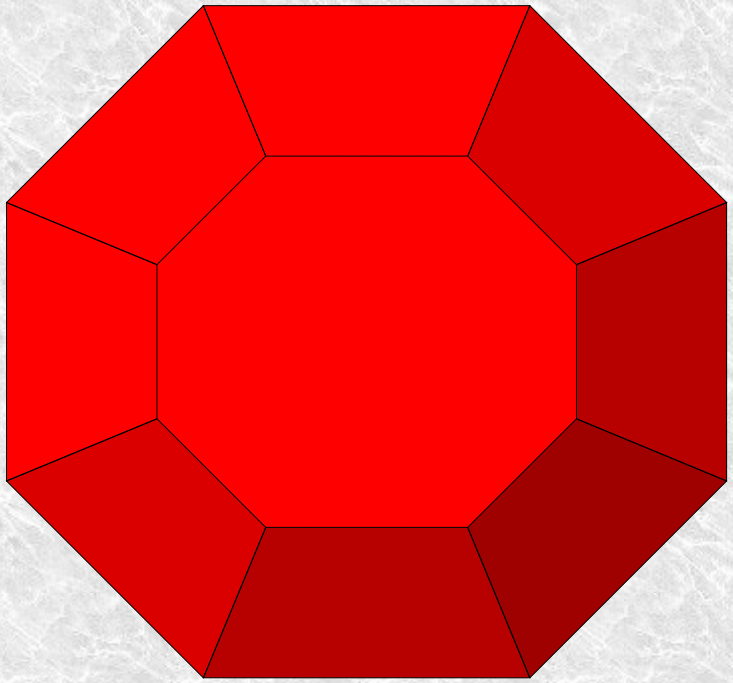
Ruby

Crash Course



Why learn ruby

- **It is the language Rails uses**
- **It may suit you better than Perl (nah!!!)**

Hello World!

```
#!/usr/bin/ruby

puts('hello world')

# note the lack of ;
```

Variables: objection

- **it's all objects**
  - **5 . minutes . ago**
- **local variables' names are plain**
  - **name="value"**
- **global variables begin with a $**
  - **$name = "value"**
- **instance variables begin with @**
  - **@name = "value"**

Variables: more

- **class (static) variables begin with @@**
  - **@@name = ”value”**
- **constants begin with a capital letter**
  - **Name = “value”**
- **arrays and hashes are just classes**
  - **arr = [’zero’,’one’,’two’]**
  - **hash = { ’one’ => 1, ’two’ => 2, ’three’ => 3}**
- **demented ruby hashes!**
  - **hash2 = { :one => 1, :two => 2 }**

strings/regexps

- **regular Perl-like regexp calls**
  - **string =~ /regexp/**
- **the *$1,$2*... exist, but discouraged**
- **there is substitute**
  - **string.sub(/regexp/,string2)**
- **concatenation it with +**
  - **string=str1+str2**

functions

Fibonacci series!
note the trinary operator ?:

```
def fib (p1)
   p1<=2?1: fib(p1-1) + fib(p1-2)
end
```

flow control

if statement:

```
if a>b
   a += 1
   b -= 1
elsif a<b
   a -= 1
   b+= 1
else
   a=0
end
```

flow control

while loop

```
while b > 0
  a *= b
  b -= 1
end
```

flow control

ruby's demented idea of a *for* loop

```
(1..10).each {|i| puts(i)}
10.times {puts('ten')}
5.upto(10) {|i| $a+=i}
# try downto
```

explanation of |i|

the *yield* function allows you to send a value inline to a function

```
def yielder
  yield('foo','bar')
end

yielder {|a,b| puts(a+b)}
```

classes: OOP(s)

```
class useless
   def initialize(parameter)
      # all parameters for new go here
      @instance_var = parameter
      @@static+=1
   end

   def method1
      #method
         puts(“why bother ”+ @instance_var +” sucks”)
   end
end

var=new useless(“itanium”)
var.method1
```

classes: injections!

```
class useless
  def method2
    puts(“try method 1!!!!!”)
  end
end

# we now have modified this class!
```

inheritance

```
class Frenchman < useless
   def initialize (parameters, so_on)
      super(parameters)
      #ignore so_on... I am lazy
   end

   def method1
      puts(“we are the French!”)
      super
   end
end
```

the list class

```
a = []                  #new
a = [1,2,3,4]
b = [2,3,4,5,6]
c = a + b               # [1,2,3,4,2,3,4,5]
c = a – b               # [1]
c = a . reject { |i| i > 3 } # [1,2,3]
a << 6                  # a == [1,2,3,4,6]
c = a * “,”             # “1,2,3,4”
a == b                  # false. exact array comparation
c = (a + b) . uniq          # [1,2,3,4,5]
#and much more
```