Seattle.rb May Challenge

Aja Hammerly
aja.hammerly@gmail.com

Who can do the most Project Euler problems in a month?

Project Euler

http://projecteuler.net

Contest Rules

- May 7th 12:00 PM - June 3rd 12:00 PM
- Language agnostic
- Solution must take less than a minute
- Don't use other people's code

Prizes

- $50 prize for most problems completed
- Two divisions
  - Novice (< 1 year programming)
  - Open (everyone)
- No prizes if < 10 people participate

To participate

- Friend me
    - 73029928140835_45edcb4f778c358b11fc8589e912f0a9
- Or send me an email of your friend code
    - aja.hammerly@gmail.com

Why Participate?

Learn Ruby

Prepare For Interviews

Be Active In Seattle.rb

Example Problem

Euler #1

"If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

"Find the sum of all the multiples of 3 or 5 below 1000.

Example Solutions

Solution 1

```
sum = 0
(1...1_000).to_a.each do |n|
  if n % 5 == 0 then
    sum += n
  elsif n % 3 == 0 then
    sum += n
  end
end
p sum
```

233168

Solution 2

```
nums = (1...1000).find_all { |x|
  x % 5 == 0 || x % 3 == 0
}
ans = nums.inject(0) { |sum, n|
  sum += n
}
p ans
```

233168

Solution 3

```
nums = (1...1000).find_all { |x|
  x % 5 == 0 || x % 3 == 0
}
p nums.inject(&:+)
```

233168

Solution 4

```
threes = (1..1000).select {|x| x%3 == 0}
fives = (1..1000).select {|x| x%5 == 0}
fifteens = (1..1000).select {|x| x%15 == 0}
(threes + fives - fifteens).inject(:+)
```

233168

Solution 5

```
p 1000.times.find_all{ |x|
    x % 5 == 0 || x % 3 == 0
  }.inject(&:+)
```

233168

Some Useful Ruby

Numbery Stuff

%

```
> 5 % 2
1
```

```
> 321 % 3
0
```

**

```
> 5 ** 2
25

> 3 ** 3
27
```

^

```
> 5 ^ 2
7

> 3 ^ 3
0
```

Prime

times

```
> 1_000
> 1000

> 2_000_000_000
> 2000000000
```

Stringy Stuff

Reverse

```
> "Hello".reverse
"olleH"
```

split

```
> "Hello".split('')
["H", "e", "l", "l", "o"]

> "Apple,Pear,Banana".split(',')
["Apples", "Pear", "Banana"]
```

Enumerable

`find_all`

```
> (1..10).find_all { |x| x < 5 }
[1, 2, 3, 4]
```

inject

```
> (0..10).inject {|s, n| s += n}
55

> (0..10).inject :+
55
```

```
downto
upto
```

```
> 3.downto(1) { |x| puts x }
3
2
1

> 3.upto(6) { |x| puts x }
3
4
5
6
```

each_cons

```
>(1..5).each_cons(3) { |a| p a }
 [1, 2, 3]
 [2, 3, 4]
 [3, 4, 5]
```

map

```
> [1, 2, 3].map { |x| x * 10 }
[10, 20, 30]
```

Other stuff

Ranges

```
> (1..3).max
3

> (1...3).max
2
```

Re-opening Classes

Timing

```
$ time ruby my_script.rb

real	0m0.118s
user	0m0.062s
sys	0m0.056s
```

aja.hammerly@gmail.com