

Regular Expressions

(in 3 minutes... lolz)

jay.hannah@iinteractive.com
[@deafferret](https://twitter.com/deafferret)

/Hannah/

/Hannah/

hannah

/Hannah/i

hannah

/Hannah/i

Hannahnator

/\bHannah\b/i

Hannahnator

/\bHannah\b/i

O'Hannah

/\bO'Hannah\b/i

O'Hannah

/\bO' ?Hannah\b/i

O' Hannah

/\b(O')?Hannah\b/i

O'Hannah

/\b(0')?Hannah\b/i

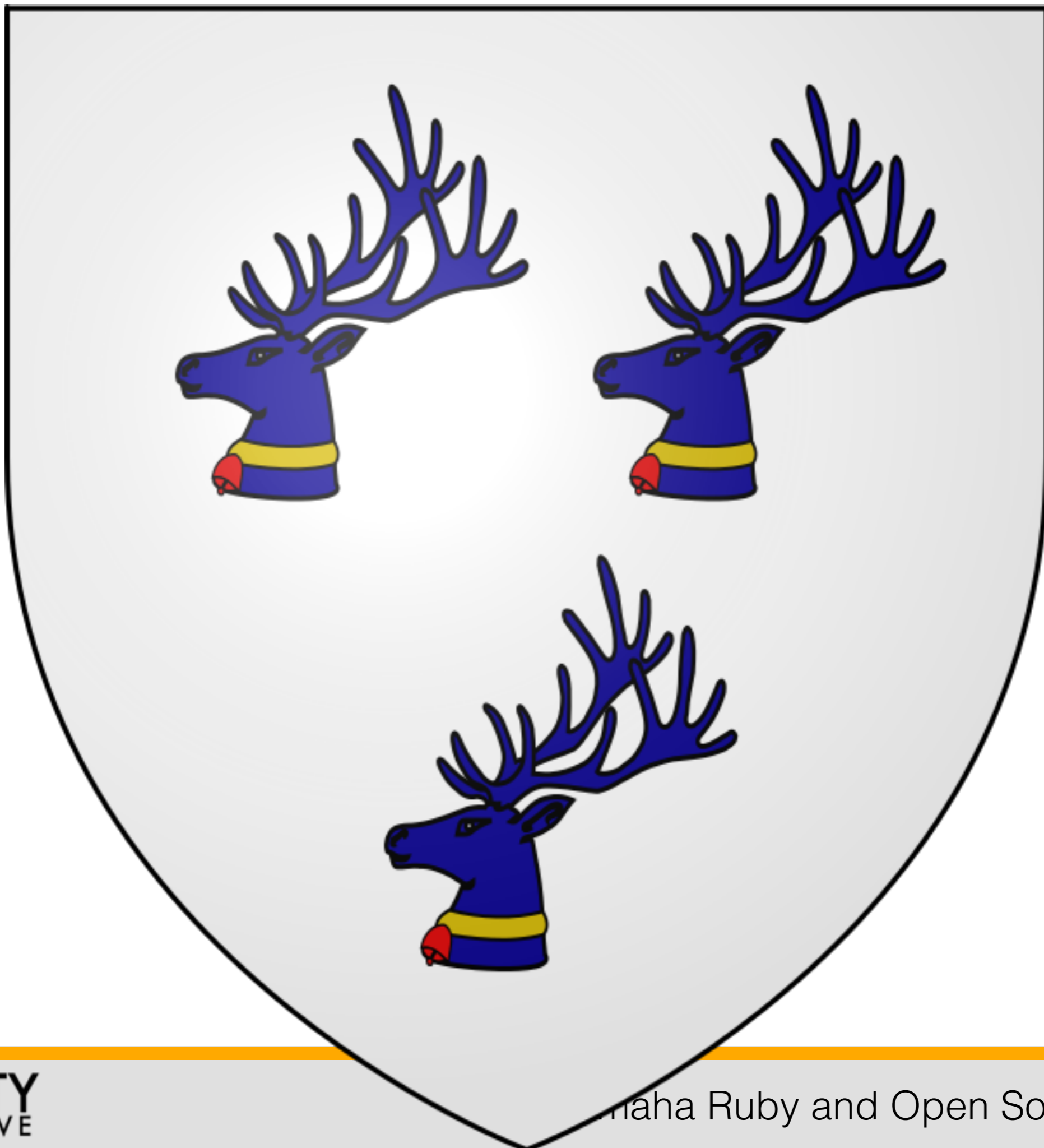
Hannay

Clan

Hannay!







/\b(O')?Hannah\b/i

Hannay

/\b(O')?Hanna[hy]\b/i

Hannay

/\b(O')?Hanna[h_y]\b/i

Brad Hannah

/\b (0') ?Hanna [hy] \b/ix

Brad Hannah

```
/  
  \b  
  (O')?  
Hanna [hy]  
  \b  
/ix
```

```
/
\b          # Word boundary
(O')?     # O' is optional
Hanna[hy] # Hannah or Hannay
\b          # Word boundary
/ix
```

```
/
\b          # Word boundary
(O')?     # O' is optional
Hanna[hy] # Hannah or Hannay
\b          # Word boundary
/iX        # i for case ins.
           # x for comments
```

```
my $re = qr/  
  \b  
  (O')?  
  Hanna[hy]  
  \b  
/ix;  
# qr// to compile  
# Word boundary  
# O' is optional  
# Hannah or Hannay  
# Word boundary  
# i for case ins.  
# x for comments
```

```
my $re = qr/      # qr// to compile
  (?<!Brad\s)    # Brad is a jerk
  \b              # Word boundary
  (O')?          # O' is optional
  Hanna[hy]      # Hannah or Hannay
  \b              # Word boundary
/iX;             # i for case ins.
                  # x for comments
```

Brad Hannah


```
use Test::More;
ok("J Hannah" =~ $re);
ok("J O'Hannay" =~ $re);
ok("Brad Hannah" !~ $re);
done_testing();
```

[https://github.com/jhannah/sandbox/tree/
master/regex](https://github.com/jhannah/sandbox/tree/master/regex)

jay.hannah@iinteractive.com
[@deafferret](#)