

Forritunarmálið Python

Day 3

Modules and Functional Programming

Hjalti Magnússon

29. nóvember 2017

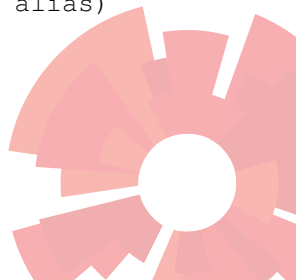


Modules



Python modules

- Python Standard Library is a large library of very useful things
- Syntax
 - `import` module (as alias)
 - from module `import` fun/cls (as alias)
- All python files are modules



Example

```
>>> import string
>>> dir(string)
['Formatter', 'Template', '_ChainMap',
 '_TemplateMetaclass', '__all__', '__builtins__',
 '__cached__', '__doc__', '__file__',
 '__loader__', '__name__', '__package__',
 '__spec__', '_re', '_string', 'ascii_letters',
 'ascii_lowercase', 'ascii_uppercase', 'capwords',
 'digits', 'hexdigits', 'octdigits', 'printable',
 'punctuation', 'whitespace']

>>> from string import punctuation
>>> punctuation
'!"#$%&\'() *+, - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~ '
```

Extending the basics



decimal - arbitrary precision floating point numbers

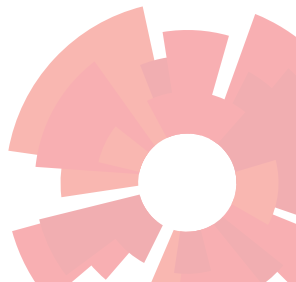
```
>>> from decimal import Decimal as D
>>> D('1.2') + D('2.2')
Decimal('3.4')
>>> 1.2 + 2.2
3.40000000000000004

>>> D('0.1') + D('0.1') + D('0.1') - D('0.3')
Decimal('0.0')
>>> 0.1 + 0.1 + 0.1 - 0.3
5.551115123125783e-17
```

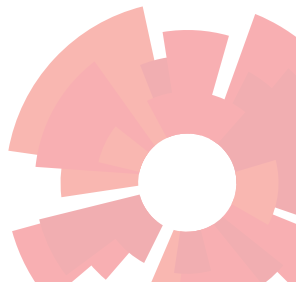
fractions - rational numbers

```
>>> from fractions import Fraction as Fr
>>> Fr(2, 3) + Fr(1, 3)
Fraction(1, 1)
>>> Fr(2, 3) + Fr(1, 3) == 1
True
>>> Fr(1.2)
Fraction(5404319552844595, 4503599627370496)
>>> Fr('1.2')
Fraction(6, 5)
>>> Fr('9/5')
Fraction(9, 5)
>>> f = Fr('66.6')
>>> f.numerator
333
>>> f.denominator
5
```

- `defaultdict`
 - A nicer version of dictionary
- `Counter`
 - One of Python's coolest data structures
- `deque`
 - If you need fast prepend, append



- A module that allows us to work with
 - Dates
 - Time
 - Dates with time
 - Time intervals



```
>>> from datetime import datetime, date
>>> datetime.now()
datetime.datetime(2017, 11, 29, 2, 5, 17, 518209)
>>> date.today()
datetime.date(2017, 11, 29)

>>> d1 = datetime(2017, 10, 28)
>>> d2 = datetime.now()
>>> d2 - d1
datetime.timedelta(32, 7517, 518447)
```

Printing and parsing

```
>>> now = datetime.now()
>>> now.strftime('%d-%m-%Y %H:%M:%S')
'29-11-2017 02:05:17'
>>> now.strftime('%d. %B %Y at %H:%M:%S')
'29. November 2017 at 02:05:17'

>>> now.strptime('29-09-2018', '%d-%m-%Y')
datetime.datetime(2018, 9, 29, 0, 0)

>>> import time
>>> time.time()
1511921117.518958
>>> datetime.fromtimestamp(time.time())
datetime.datetime(2017, 11, 29, 2, 5, 17, 519030)
```

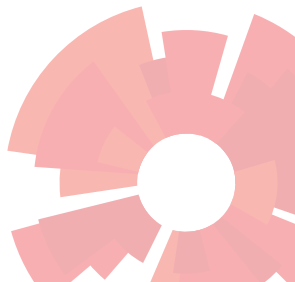
Debugging



pdb - the Python debugger

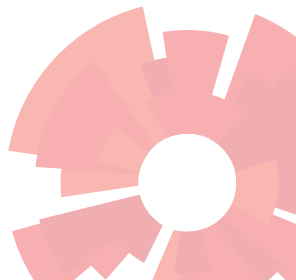
Python comes with a built in debugger, pdb

```
# Set a breakpoint  
import pdb; pdb.set_trace()
```



Commands

- **s(tep)** “Jump into”
- **n(next)** “Jump over”
- **c(ontinue)** Run until next breakpoint
- **r(eturn)** Run until current function returns
- **l(ist)** (**ll**) Print where you are
- **p(rint)** (**pp**) Print
- **q(uit)** Quit the debugger
- **!** Execute statement
- **interact** Start Python interpreter

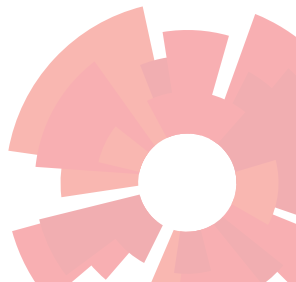


Functions



More about functions

- In Python functions are first class citizens
 - They can be passed as variables
 - They can be returned by functions



Lambda functions

Python provides anonymous functions

```
>>> lambda x: x ** 2
<function <lambda> at 0x7f06fbb391e0>

>>> f = lambda x: x ** 2
>>> f(4)
16

# They can take more than one parameter
>>> g = lambda x, y: x * y
>>> g(4, 5)
20

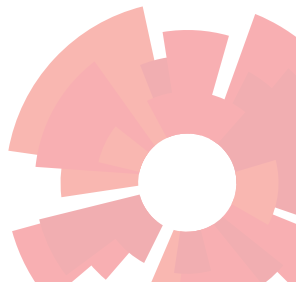
# Or none
>>> h = lambda: 1377
>>> h()
1377
```

Functional programming



Many builtin functions in Python take functions as parameters

- `min, max, sum`
- `map, filter, all, any`
- `sorted`

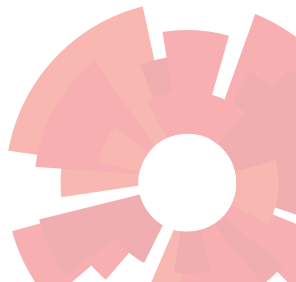


Problems



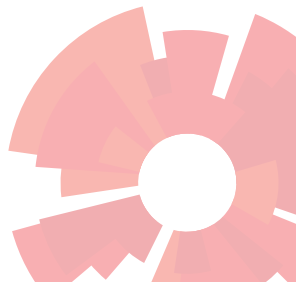
Problem 0

Find the word count of The Raven



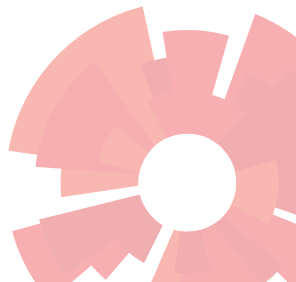
Problem 1

Find the country with the most unique letters



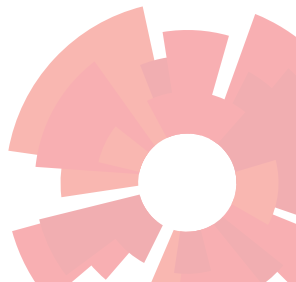
Problem 2

Find the country in Europe with the highest population



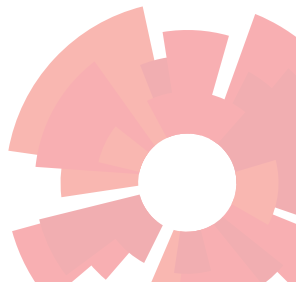
Problem 3

Sort all countries by population



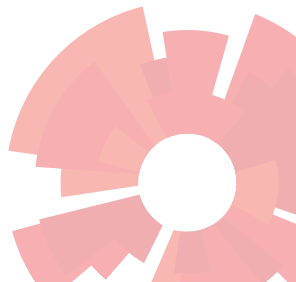
Problem 4

Sort all countries by population



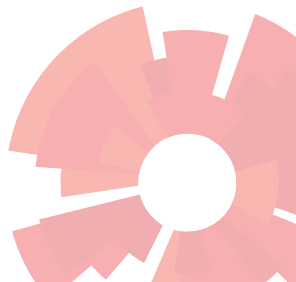
Problem 5

Find the user with the most accepted submissions



Problem 6

For each user that has submitted problems, find the number of solved problems he has solved and his full name



Assert



Assert is a friend

- Assert that the state of your program is consistent
 - Check that “this should never happen” doesn’t happen

