Google Search API

Google Search API is a python based library for searching various functionalities of google. It uses screen scraping to retrieve the results, and thus is unreliable if the way google's web pages are returned change in the future.

Disclaimer: This software uses screen scraping to retreive search results from google.com, and therefore this software may stop working at any given time. Use this software at your own risk. I assume no responsibility for how this software API is used by others.

Google Web Search

You can search google web in the following way:

```
search_results = Google.search("This is
my query")
```

`search_results` will contain a list of `GoogleResult` objects

```
GoogleResult:
    self.name # The title of the link
```

```
    self.link # The link url
    self.description # The description of
the link
    self.thumb # The link to a thumbnail
of the website (not implemented yet)
    self.cached # A link to the cached
version of the page
    self.page # What page this result was
on (When searching more than one page)
    self.index # What index on this page
it was on
```

Google Calculator

Attempts to search google calculator for the result of an expression. Returns a `CalculatorResult` if successful or `None` if it fails.

```
Google.calculate("157.3kg in grams")
```

```
{'expr': u'157.3 kilograms',
 'fullstring': u'157.3 kilograms =
157\xa0300 grams',
 'result': u'157 300 grams',
 'unit': u'grams',
 'value': u'157300'}
```

```
Google.calculate("cos(25 pi) / 17.4")
```

```
{'expr': u'cos(25 * pi) / 17.4',
 'fullstring': u'cos(25 * pi) / 17.4 =
-0.0574712644',
 'result': u'-0.0574712644',
 'unit': None,
 'value': u'-0.0574712644'}
```

Google Image Search

Searches google images for a list of images. Image searches can be filtered to produce better results.

Perform a google image search on “banana” and filter it:

```
options = ImageOptions()
options.image_type = ImageType.CLIPART
options.larger_than = LargerThan.MP_4
options.color = "green"
results = Google.search_images("banana",
options)
```

Sample Result:

```
{'domain': u'exitrealworld.com',
 'filesize': u'4054k',
 'format': u'jpg',
 'height': u'3103',
 'index': 0,
 'link':
u'http://www.exitrealworld.com/tools_v2/resc

 'name': u'Lib Tech Skate Banana BTX',
 'page': 0,
 'thumb': u'http://t3.gstatic.com/images?
q=tbn:ANd9GcRzvAUW0en9eZTag3giWelcQ_xbrnBMXV

 'width': u'3104'}
```

Filter options:

```
ImageOptions:
    image_type # face, body, clipart,
line drawing
    size_category # large, small, icon
    larger_than # the well known name of
the smallest image size you want
    exact_width # the exact width of the
image you want
    exact_height # the exact height of
the image you want
    color_type # color, b&w, specific
    color # blue, green, red
```

Enums of values that can be used to filter image searches:

```
class ImageType:
    NONE = None
    FACE = "face"
    PHOTO = "photo"
    CLIPART = "clipart"
    LINE_DRAWING = "lineart"

class SizeCategory:
    NONE = None
    ICON = "i"
    LARGE = "l"
    MEDIUM = "m"
    SMALL = "s"
    LARGER_THAN = "lt"
    EXACTLY = "ex"

class LargerThan:
    NONE = None
    QSVGA = "qsvga" # 400 x 300
    VGA = "vga"     # 640 x 480
    SVGA = "svga"   # 800 x 600
    XGA = "xga"     # 1024 x 768
    MP_2 = "2mp"    # 2 MP (1600 x 1200)
    MP_4 = "4mp"    # 4 MP (2272 x 1704)
    MP_6 = "6mp"    # 6 MP (2816 x 2112)
    MP_8 = "8mp"    # 8 MP (3264 x 2448)
    MP_10 = "10mp"  # 10 MP (3648 x 2736)
```

```
    MP_12 = "12mp"  # 12 MP (4096 x 3072)
    MP_15 = "15mp"  # 15 MP (4480 x 3360)
    MP_20 = "20mp"  # 20 MP (5120 x 3840)
    MP_40 = "40mp"  # 40 MP (7216 x 5412)
    MP_70 = "70mp"  # 70 MP (9600 x 7200)

class ColorType:
    NONE = None
    COLOR = "color"
    BLACK_WHITE = "gray"
    SPECIFIC = "specific"
```

Google Currency Converter (Exchange Rates)

Convert between one currency and another using google calculator. Results are real time and can change at any time based on the current exchange rate according to google.

Convert 5 US Dollars to Euros using the official 3 letter currency acronym:

```
euros = Google.convert_currency(5.0,
"USD", "EUR")
print "5.0 USD = {0} EUR".format(euros)
```

```
5.0 USD = 3.82350692 EUR
```

Convert 1000 Japanese Yen to US Dollars:

```
yen = Google.convert_currency(1000,
"yen", "us dollars")
print "1000 yen = {0} us
dollars".format(yen)
```

```
1000 yen = 12.379 us dollars
```

Instead you can get the exchange rate which returns what 1 `from_currency` equals in `to_currency` and do your own math:

```
rate = Google.exchange_rate("dollars",
"pesos")
print "dollars -> pesos exchange rate =
{0}".format(rate)
```

```
dollars -> pesos exchange rate =
13.1580679
```

Perform your own math. The following 2 statements are equal:

```
5.0 * Google.exchange_rate("USD", "EUR")
```

```
Google.convert_currency(5.0, "USD",
"EUR")
```

As a side note, `convert_currency` is always more accurate than performing your own math on `exchange_rate` because of possible rounding errors. However if you have more than one value to convert it is best to call `exchange_rate` and cache the result to use for multiple calculations instead of querying the google server for each one.
