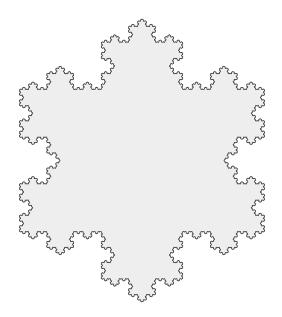
Fractal animation introduction

Grégory Châtel

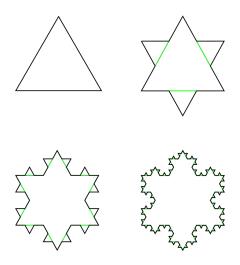
@rodgzilla github.com/rodgzilla

October 18th, 2018

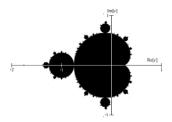
What are fractals?



Koch snowflake



Pythagora's tree demo



Complex numbers and complex plane

$$c = ai + b$$
$$i^2 = -1$$

The *norm* of a complex number is its euclidiean distance to 0:

$$c = ai + b$$
$$|c| = \sqrt{a^2 + b^2}$$

```
In [1]: c = complex(1, 2)
In [2]: c
Out[2]: (1+2j)
In [3]: c + complex(0,3)
Out[3]: (1+5j)
In [4]: c * c
Out[4]: (-3+4j)
```

Function to iterate

The formula that generates everything is the following one:

$$z_0 = c$$
$$z_{n+1} = z_n^2 + c$$

For each pixel (x, y) of the screen, we compute the corresponding complex number $c_{x,y}$.

Now we compute a fixed number of terms of the sequence above starting with $z_0 = c_{x,y}$.

$$z_0 = c_{x,y}$$

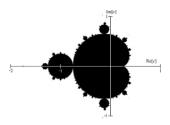
 $z_1 = c_{x,y}^2 + c_{x,y}$
 $z_2 = z_1^2 + c_{x,y} = (c_{x,y}^2 + c_{x,y})^2 + c_{x,y}$
...
 $z_{100} = z_{99}^2 + c_{x,y}$

In Python:

 $mandelbrot_function = lambda z, c: z ** 2 + c$

From the formula to the drawing

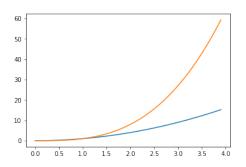
While we are iterating the formula for a pixel (x, y), if we find a step i for which $|z_i| > 2$ then we draw the (x, y) pixel black, otherwise we draw it white.



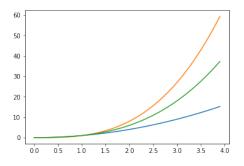
Animation

Function interpolation with Python

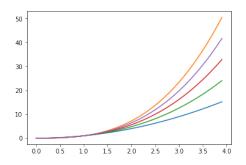
```
f_square = lambda x: x ** 2
f_cube = lambda x: x ** 3
```



Function interpolation with Python



Function interpolation with Python



Mandelbrot set animations