Function Demos

Announcements

Currying

Function Currying

```
def make_adder(n):
    return lambda k: n + k
```

```
>>> make_adder(2)(3)
5
>>> add(2, 3)
5
```

There's a general relationship between these functions

(Demo)

Curry: Transform a multi-argument function into a single-argument, higher-order function

Example: Reverse

The square function can be defined in terms of the built-in pow function:

```
def square(x):                def cube(x):
    """Square x.                  """Cube x.

    >>> square(3)                 >>> cube(3)
    9                             27
    """                           """
    return pow(x, 2)              return pow(x, 3)
```

Define square and cube in one line without using lambda or ** (using curry and reverse).

```
def reverse(f):                             def curry(f):
    return lambda x, y: f(y, x)                 def g(x):
                                                    def h(y):
                                                        return f(x, y)
                                                    return h
                                                return g
```

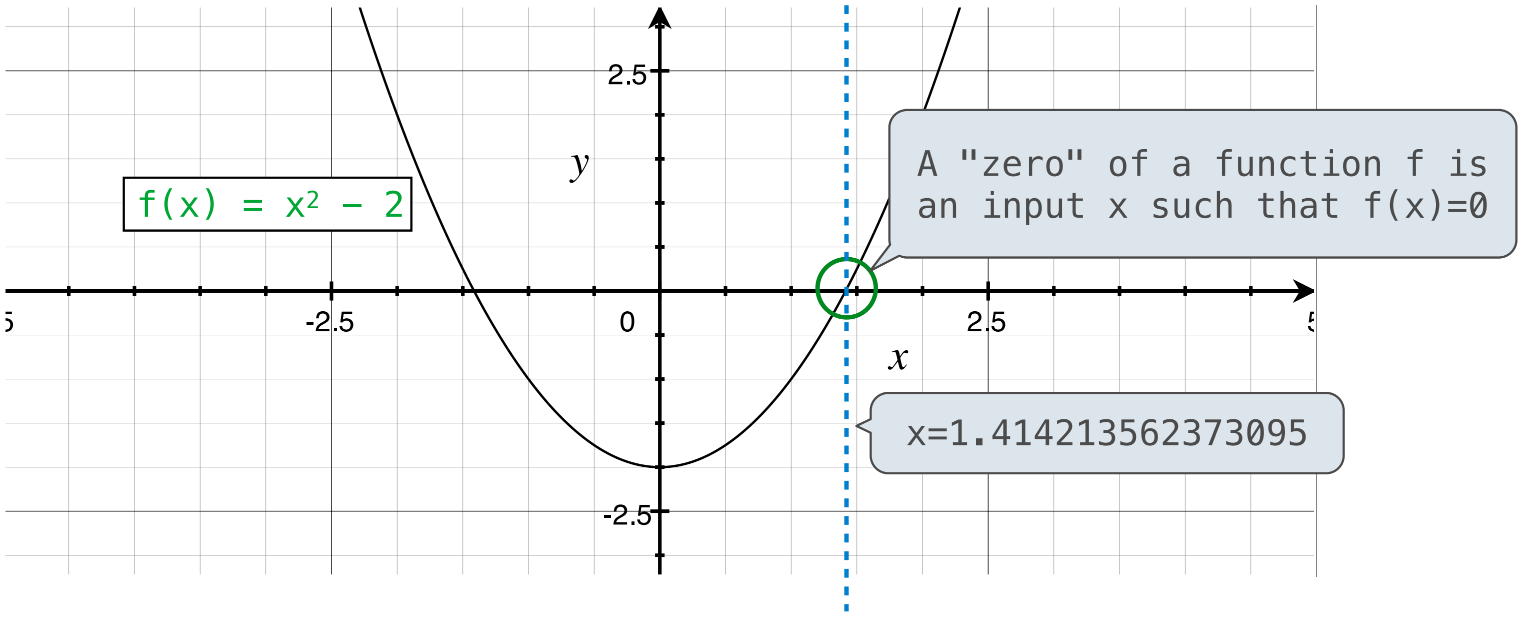
```
square = curry(reverse(pow))(2)
cube   = curry(reverse(pow))(3)
```

Example: Newton's Method (OPTIONAL)

`(Once upon a time, this example was tested on midterms, but now it's not.)`

Newton's Method Background

Quickly finds accurate approximations to zeroes of differentiable (smooth) functions



Application: a method for computing square roots, cube roots, etc.

The positive zero of $f(x) = x^2 - a$ is $\sqrt{a}$. (We're solving the equation $x^2 = a$.)

Newton's Method

Given a function f and initial guess x,

Repeatedly improve x:

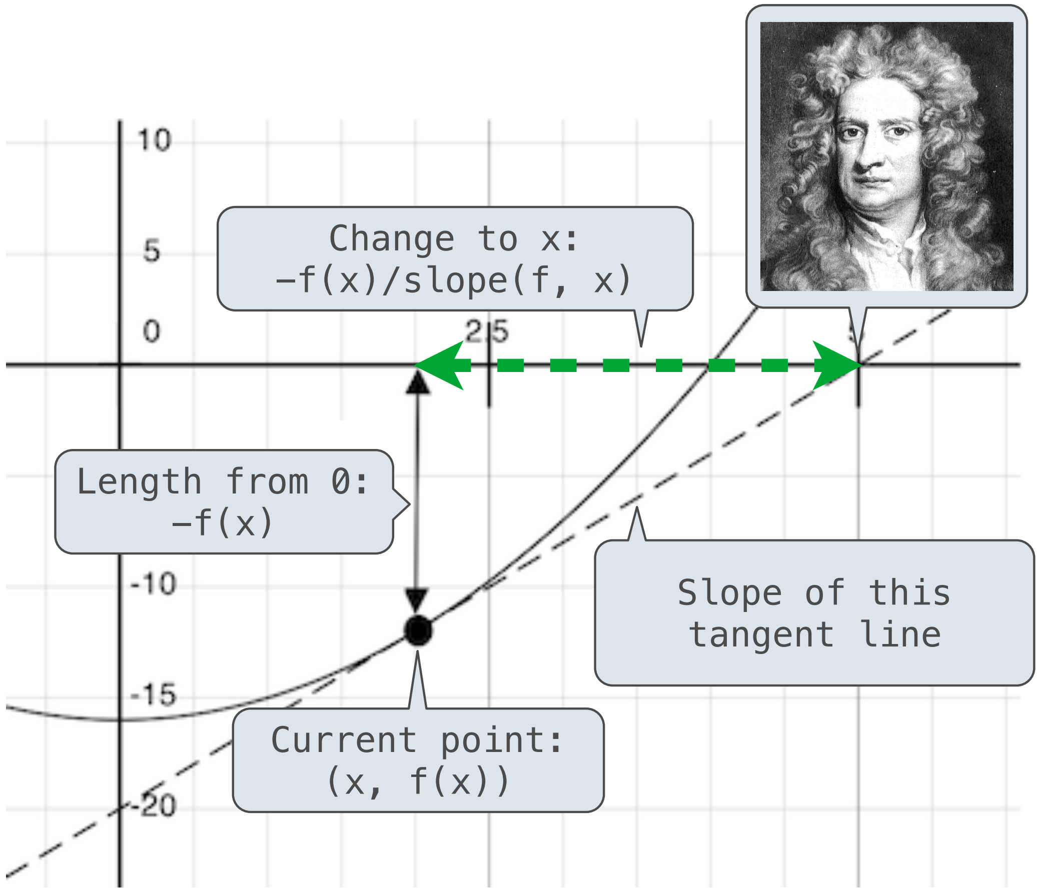
Compute the value of f at the guess: f(x)

Compute the slope of f at the guess: slope(f, x)

Update guess x to be:

$$x - \frac{f(x)}{slope(f, x)}$$

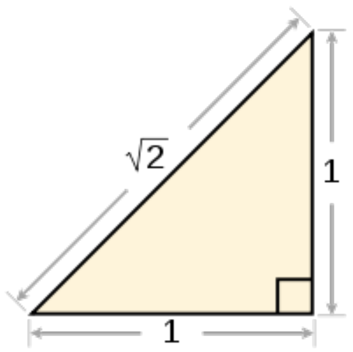
Finish when f(x) = 0 (or close enough)



http://en.wikipedia.org/wiki/File:NewtonIteration_Ani.gif

Using Newton's Method

How to find the square root of 2?



```
>>> f  = lambda x: x*x - 2
>>> find_zero(f)
1.4142135623730951
```

Applies Newton's method

(Demo)

Fall 2012 Midterm 1 Question 4(a): Inverse

Implement inverse, which takes a one-argument numerical function and returns its inverse.

The inverse of some function F is a function of argument X that returns you the Y, such that when you apply F to Y you recover the X.

```
def find_zero(f, x=1):
    """Return a zero of the function f."""

def sqrt(a):
    """Return the square root of a."""
    def f(x):
        return x*x - a
    return find_zero(f)

def inverse(f):
    """Return the inverse function of f.

    >>> sqrt = inverse(lambda x: x * x)
    >>> sqrt(16)
    4.0
    """
    return lambda y: find_zero(lambda x: f(x)-y)
```

Twenty-One Environment Diagram

Twenty-One Rules

Two players alternate turns, on which they can add 1, 2, or 3 to the current total

The total starts at 0

The game end whenever the total is 21 or more

The last player to add to the total loses

Some states are good; some are bad

$$\begin{tikzcd}
 & & 19 \arrow[dl] & & 15 \arrow[dl] & & \\
21+ & 20 \arrow[l] & 18 \arrow[l] & 16 \arrow[ul] \arrow[l] \arrow[dl] & 14 \arrow[l] & 12 \arrow[ul] \arrow[l] \arrow[dl] & \ldots \\
 & & 17 \arrow[ul] & & 13 \arrow[ul] & &
\end{tikzcd}$$

(Demo)

https://pythontutor.com/cp/composingprograms.html#code=def%20play%28strategy0,%20strategy1,%20goal%3D21%29%3A%0A%20%20%20%20n,
%20who%20%3D%200,%200%0A%20%20%20%20while%20n%20%3C%20goal%3A%0A%20%20%20%20%20%20%20%20if%20who%20%3D%3D%200%3A%0A%20%20%20%20%20%20%20%20%20%20%20%20n%20%3D%20n%20%2B%20strategy0%28n%29%0A%20%20%20%20%20%20%20%20%20%20%20%20who%20%3D%201%0A%20%20%20%20%20%20%20%20elif%20who%20%3D%3D%201%3A%0A%20%20%20%20%20%20%20%20%20%20%20%20n%20%3D%20n%20%2B%20strategy1%28n%29%0A%20%20%20%20%20%20%20%20%2
0%20%20%20who%20%3D%200%0A%20%20%20%20return%20who%20%0A%0Adef%20two_strat%28n%29%3A%0A%20%20%20%20return%202%0A%0Adef%20noisy_strat%28who,%20s%29%3A%0A%20%20%20%20def%20strat%28n%29%3A%0A%20%20%20%20%20%20%20%20choice%20%3D%20s%28n%29%0A%20%20%20%20%20%20%20%20print%28'Player',%20who,%20'added',%20choice,%20'to',%20n,%20'to%20reach',
%20choice%20%2B%20n%29%0A%20%20%20%20%20%20%20%20return%20choice%0A%20%20%20%20return%20strat%0A%0Aplay%28noisy_strat%280,%20two_strat%29,%20noisy_strat%281,%20two_strat%29%29&cumulative=true&curInstr=0&mode=display&origin=composingprograms.js&py=3&rawInputLstJSON=%5B%5D