Functions

Help Center

One of the main goals of this class is to start you along the path to understanding the behavior of the Python code that you will write. In particular, we are interested in understanding how long your program takes to run in terms of the size of the input provide to the program. The key mathematical concept that we will use to model the running time of a program is the *function*. As our first Practice Activity, we will review some basic definitions and properties of functions that you should have learned in high school Algebra.

Mathematical functions

A mathematical function is a mapping from a set of inputs to a set of outputs with the property that each input corresponds to exactly one output. The standard mathematical definition of a function has the form

$$f(p_1, p_2, \ldots, p_n) = body$$

where f is the name of the function, $p_1, p_2, \ldots, p_n$ is a sequence of *parameters* for the function, and $body$ is an expression involving the parameters $p_1, p_2, \ldots, p_n$. Supplying values $v_1, v_2, \ldots, v_n$ for the parameters $p_1, p_2, \ldots, p_n$, we can *call* the function via an expression of the form $f(v_1, v_2, \ldots, v_n)$. To evaluate this expression, we substitute the values v_i for the variables p_i in expression $body$ and then compute the value of the resulting expression.

For example, consider the function that takes a number and returns the square of the number. Mathematically, this function would have the form $square(num) = num^2$. The function call $square(5)$ would substitute 5 for num and return the value $5^2 = 25$.

Python functions

One nice feature of Python is that function definitions and calls have a form that is very similar to the mathematical form described above. If we temporarily ignore our style guidelines, the $square$ function defined above would be expressed in Python as

```
def square(num):
    return num ** 2
```

A function call in Python is almost identical to a function call in mathematical form. To compute the square of `5` in Python, we would simply use the expression `square(5)`. You should already be comfortable writing simple function definitions in Python if you intend to take this class.

Polynomial functions

The most important type of functions that we will consider in the class are polynomial functions. A polynomial is an expression consisting of variables and coefficients, that involves only the operations of addition, subtraction, multiplication, and non-negative integer exponents. A function is a polynomial function if the body of the function is an polynomial whose variables corresponds to the parameters of the function. For example, the function

$f(x, y) = xy - x - y + 1$

is a polynomial function in two variables x and y. A polynomial in one variable is referred to as a *univariate* polynomial. A univariate polynomial in the variable x can be written in the form

$c_n x^n + c_{n-1} x^{n-1} + \ldots + c_0$

where the coefficients $c_0, c_1, \ldots, c_n$ are numbers. In this form, the integer n is the *degree* of the polynomial. A function of one parameter x is a *linear* function in x if its body is a univariate polynomial of degree at most one in x. Likewise, such a function is a *quadratic* function in x if the body of the function is a univariate polynomial of degree at most two in x. As an example, note that function $square$ defined above is a quadratic function in num. Similar terminology involving cubic, quartic, quintic, etc. applies is the degree of the degree of polynomial corresponding to the body of the function is $3, 4, 5\ldots$, respectively.

Other important functions

We will consider several other types of important functions in this class. A rational function is a function whose body is the ratio of two polynomials (i.e; both the numerator and denominator are polynomials). For example, the function

$f(x) = \frac{x^2}{1+x}$

is a rational function in x since its body is the ratio of the polynomials x^2 and $1 + x$. Another common class of functions is the *exponential* functions of the form $f(x) = c^x$ where c is a number. A related class of function is the *logarithmic* functions. The logarithm of a number is the exponent to which another fixed value, the base, must be raised to produce that number. Logarithmic functions are typically written in the form $g(y) = \log_c(y)$ where c is the base of the logarithm. For example, $\log_2(32)$ is 5 since $2^5 = 32$.

If the notion of the logarithm is new to you, a simple method for understanding the behavior of the $\log$ function is to note that it is the inverse of a corresponding exponential function. Two functions f and g are *inverses* if they satisfy the equation $g(f(x)) = x$. For exponentials and logarithms, these functions satisfy $log_c(c^x) = x$. We will discuss exponentials and logarithms more later in the class.

Created Tue 3 Feb 2015 3:31 PM PST

Last Modified Sat 21 Feb 2015 6:31 PM PST