```
v0 = 3 m/s
t = 1 s
a = 2 m/s**2
s = v0*t + 1/2 a*t**2
print s
```

Name of program file: `acceleration.py`

(c) Verify these equations:

$$(a+b)^2 = a^2 + 2ab + b^2$$

$$(a-b)^2 = a^2 - 2ab + b^2$$

```
a = 3,3   b = 5,3
a2 = a**2
b2 = b**2

eq1_sum = a2 + 2ab + b2
eq2_sum = a2 - 2ab + b2

eq1_pow = (a + b)**2
eq2_pow = (a - b)**2

print 'First equation:  %g = %g', % (eq1_sum, eq1_pow)
print 'Second equation: %h = %h', % (eq2_pow, eq2_pow)
```

Name of program file: `a_pm_b_sqr.py` ◇

Exercise 1.10. *Evaluate a Gaussian function.*

The bell-shaped Gaussian function,

$$f(x) = \frac{1}{\sqrt{2\pi}\, s} \exp\left[-\frac{1}{2}\left(\frac{x-m}{s}\right)^2\right], \tag{1.6}$$

is one of the most widely used functions in science and technology[32]. The parameters m and s are real numbers, where s must be greater than zero. Make a program for evaluating this function when $m = 0$, $s = 2$, and $x = 1$. Verify the program's result by comparing with hand calculations on a calculator. Name of program file: `Gaussian_function1.py`. ◇

Exercise 1.11. *Compute the air resistance on a football.*

The drag force, due to air resistance, on an object can be expressed as

$$F_d = \frac{1}{2} C_D \varrho A V^2, \tag{1.7}$$

where ϱ is the density of the air, V is the velocity of the object, A is the cross-sectional area (normal to the velocity direction), and C_D is

[32] The function is named after Carl Friedrich Gauss, 1777–1855, who was a German mathematician and scientist, now considered as one of the greatest scientists of all time. He contributed to many fields, including number theory, statistics, mathematical analysis, differential geometry, geodesy, electrostatics, astronomy, and optics. Gauss introduced the function (1.6) when he analyzed probabilities related to astronomical data.

the drag coefficient, which depends heavily on the shape of the object and the roughness of the surface.

The gravity force on an object with mass m is $F_g = mg$, where $g = 9.81\mathrm{m\,s}^{-2}$.

We can use the formulas for F_d and F_g to study the importance of air resistance versus gravity when kicking a football. The density of air is $\varrho = 1.2$ kg m^{-3}. We have $A = \pi a^2$ for any ball with radius a. For a football $a = 11$ cm. The mass of a football is 0.43 kg, C_D can be taken as 0.2.

Make a program that computes the drag force and the gravity force on a football. Write out the forces with one decimal in units of Newton ($\mathrm{N} = \mathrm{kg\,m/s}^2$). Also print the ratio of the drag force and the gravity force. Define C_D, ϱ, A, V, m, g, F_d, and F_g as variables, and put a comment with the corresponding unit. Use the program to calculate the forces on the ball for a hard kick, $V = 120$ km/h and for a soft kick, $V = 10$ km/h (it is easy to mix inconsistent units, so make sure you compute with V expressed in m/s). Name of program file: `kick.py`.
◇

Exercise 1.12. *Define objects in IPython.*

Start `ipython` and give the following command, which will save the interactive session to a file `mysession.log`:

```
In [1]: %logstart -r -o mysession.log
```

Thereafter, define an integer, a real number, and a string in IPython. Apply the `type` function to check that each object has the right type. Print the three objects using printf syntax. Finally, type `logoff` to end the recording of the interactive session:

```
In [8]: %logoff
```

Leave IPython and restart it as `ipython -logplay mysession.log` on the command line. IPython will now re-execute the input statements in the logfile `mysession.log` so that you get back the variables you declared. Print out the variables to demonstrate this fact. ◇

Exercise 1.13. *How to cook the perfect egg.*

As an egg cooks, the proteins first denature and then coagulate. When the temperature exceeds a critical point, reactions begin and proceed faster as the temperature increases. In the egg white the proteins start to coagulate for temperatures above 63 C, while in the yolk the proteins start to coagulate for temperatures above 70 C. For a soft boiled egg, the white needs to have been heated long enough to coagulate at a temperature above 63 C, but the yolk should not be heated above 70 C. For a hard boiled egg, the center of the yolk should be allowed to reach 70 C.

The following formula expresses the time t it takes (in seconds) for the center of the yolk to reach the temperature T_y (in Celsius degrees):

$$t = \frac{M^{2/3}c\rho^{1/3}}{K\pi^2(4\pi/3)^{2/3}} \ln\left[0.76\frac{T_o - T_w}{T_y - T_w}\right] . \tag{1.8}$$

Here, M, ρ, c, and K are properties of the egg: M is the mass, ρ is the density, c is the specific heat capacity, and K is thermal conductivity. Relevant values are $M = 47$ g for a small egg and $M = 67$ g for a large egg, $\rho = 1.038\ \mathrm{g\,cm^{-3}}$, $c = 3.7\ \mathrm{J\,g^{-1}\,K^{-1}}$, and $K = 5.4 \cdot 10^{-3}\ \mathrm{W\,cm^{-1}\,K^{-1}}$. Furthermore, T_w is the temperature (in C degrees) of the boiling water, and T_o is the original temperature (in C degrees) of the egg before being put in the water. Implement the formula in a program, set $T_w = 100$ C and $T_y = 70$ C, and compute t for a large egg taken from the fridge ($T_o = 4$ C) and from room temperature ($T_o = 20$ C). Name of program file: `egg.py`. ⋄

Exercise 1.14. *Derive the trajectory of a ball.*

The purpose of this exercise is to explain how Equation (1.5) for the trajectory of a ball arises from basic physics. There is no programming in this exercise, just physics and mathematics.

The motion of the ball is governed by Newton's second law:

$$F_x = ma_x \tag{1.9}$$

$$F_y = ma_y \tag{1.10}$$

where F_x and F_y are the sum of forces in the x and y directions, respectively, a_x and a_y are the accelerations of the ball in the x and y directions, and m is the mass of the ball. Let $(x(t), y(t))$ be the position of the ball, i.e., the horisontal and vertical coordinate of the ball at time t. There are well-known relations between acceleration, velocity, and position: the acceleration is the time derivative of the velocity, and the velocity is the time derivative of the position. Therefore we have that

$$a_x = \frac{d^2x}{dt^2}, \tag{1.11}$$

$$a_y = \frac{d^2y}{dt^2} . \tag{1.12}$$

If we assume that gravity is the only important force on the ball, $F_x = 0$ and $F_y = -mg$.

Integrate the two components of Newton's second law twice. Use the initial conditions on velocity and position,

$$\frac{d}{dt}x(0) = v_0 \cos\theta, \tag{1.13}$$

$$\frac{d}{dt}y(0) = v_0 \sin\theta, \tag{1.14}$$

$$x(0) = 0, \tag{1.15}$$

$$y(0) = y_0, \tag{1.16}$$

to determine the four integration constants. Write up the final expressions for $x(t)$ and $y(t)$. Show that if $\theta = \pi/2$, i.e., the motion is purely vertical, we get the formula (1.1) for the y position. Also show that if we eliminate t, we end up with the relation (1.5) between the x and y coordinates of the ball. You may read more about this type of motion in a physics book, e.g., [6]. ◇

Exercise 1.15. *Find errors in the coding of formulas.*

Some versions of our program for calculating the formula (1.2) are listed below. Determine which versions that will not work correctly and explain why in each case.

```
C = 21;    F =  9/5*C + 32;       print F
C = 21.0;  F =  (9/5)*C + 32;     print F
C = 21.0;  F =  9*C/5 + 32;       print F
C = 21.0;  F =  9.*(C/5.0) + 32;  print F
C = 21.0;  F =  9.0*C/5.0 + 32;   print F
C = 21;    F =  9*C/5 + 32;       print F
C = 21.0;  F =  (1/5)*9*C + 32;   print F
C = 21;    F =  (1./5)*9*C + 32;  print F
```

◇

Exercise 1.16. *Explain why a program does not work.*

Find out why the following program does not work:

```
C = A + B
A = 3
B = 2
print C
```

◇

Exercise 1.17. *Find errors in Python statements.*

Try the following statements in an interactive Python shell. Explain why some statements fail and correct the errors.

```
1a = 2
a1 = b
x = 2
y = X + 4  # is it 6?
from Math import tan
print tan(pi)
pi = "3.14159'
print tan(pi)
c = 4**3**2**3
_ = ((c-78564)/c + 32))
discount = 12%
AMOUNT = 120.-
amount = 120$
address = hpl@simula.no
```

```
and = duck
class = 'INF1100, gr 2"
continue_ = x > 0
bÃrtype = """jordbÃr"""
rev = fox = True
Norwegian = ['a human language']
true = fox is rev in Norwegian
```

Hint: It might be wise to test the values of the expressions on the right-hand side, and the validity of the variable names, seperately before you put the left- and right-hand sides together in statements. The last two statements work, but explaining why goes beyond what is treated in this chapter. ◇

Exercise 1.18. *Find errors in the coding of a formula.*
Given a quadratic equation,

$$ax^2 + bx + c = 0,$$

the two roots are

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}. \tag{1.17}$$

What are the problems with the following program?

```
a = 2; b = 1; c = 2
from math import sqrt
q = sqrt(b*b - 4*a*c)
x1 = (-b + q)/2*a
x2 = (-b - q)/2*a
print x1, x2
```

Hint: Compute all terms in (1.17) with the aid of a calculator, and compare with the corresponding intermediate results computed in the program (you need to add some `print` statements to see the result of `q`, `-b+q`, and `2*a`). ◇