Funciones

In [14]:
```
def max(x, y):
    return (x + y + abs(x - y)) // 2
max(2, 3)
```

Out[14]: 3

In [9]:
```
def mean(a1, a2, a3, a4):
    s = 0.0
    s = s + a1
    s = s + a2
    s = s + a3
    s = s + a4
    return s / 4

mean(1, 2, 3, 4)
```

Out[9]: 2.5

In [8]:
```
import math
def circle(radius):
    return math.pi * radius ** 2
```

In [5]:
```
a = circle(2) # the typical use of a function
a
```

Out[5]: 12.566370614359172

Debemos indicar lo que hace la función, el tipo que se espera en los parámetros de entrada y el valor devuelto

In [14]:
```
def max(x, y):
    """Function that return the maximum of 2 values
    @type x: int
    @type y: int
    @rtype: int
    """
    return (x + y + abs(x - y)) // 2
max(2, 3)
```

Out[14]: 3

In [10]:
```
def mean(a1, a2, a3, a4):
    """Function that return the mean of 4 values
    @type a1: real
    @type a2: real
    @type a3: real
    @type a4: real
    @rtype: real
    """
    s = 0.0
    s = s + a1
    s = s + a2
    s = s + a3
    s = s + a4
    return s / 4

mean(1, 2, 3, 4)
```

Out[10]: 2.5

In [4]:
```
def circle(radius):
    """Function that computes the surface of a circle
    @type radius: real
    @rtype: real
    @precondition: radius>=0
    """
    sur = math.pi * radius ** 2
    return sur
```

Una función se puede dividir en pasos más sencillos.

Además, una función puede no funcionar con todos los posibles valores. Una ecuación de segundo grado no tiene soluciones reales si el discriminante es negativo o si \(a=0\), siendo \(a\) el coeficiente del monomio de grado 2.

Por último, la función debe devolver 2 soluciones.

In [5]:
```
def quadratic(a, b, c):
    """Function that computes the solution of a quadratic ecuation
        a * x**2 + b * x + c = 0.
    @type a: real
    @type b: real
    @type c: real
    @rtype: tuple of reals
    @precondition: a>0 and b*b-4*a*c >= 0
    """
    disc = b*b - 4*a*c
    sol1 = (-b + math.sqrt(disc)) / (2*a)
    sol2 = (-b - math.sqrt(disc)) / (2*a)
    return sol1, sol2
```

In [6]:
```
a, b = quadratic(1, 0, -4) # the function returns a tuple of reals. We need two variab
les to access them.
a
```

Out[6]: 2.0

In [7]:
```
b
```

Out[7]:
```
-2.0
```

In [8]:
```
b
```

Out[8]:
```
-2.0
```

¿qué ocurrre a la función si no se llama con los valores que se indican en la precondición?

In [9]:
```
a, b = quadratic(1,0,1)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-9-f60c5d5106da> in <module>()
----> 1 a, b = quadratic(1,0,1)

<ipython-input-5-c7d771477d7a> in quadratic(a, b, c)
      9     """
     10     disc = b*b - 4*a*c
---> 11     sol1 = (-b + math.sqrt(disc)) / (2*a)
     12     sol2 = (-b - math.sqrt(disc)) / (2*a)
     13     return sol1, sol2

ValueError: math domain error
```

In [10]:
```
a, b = quadratic(0,1,1)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-10-42d03cb91481> in <module>()
----> 1 a, b = quadratic(0,1,1)

<ipython-input-5-c7d771477d7a> in quadratic(a, b, c)
      9     """
     10     disc = b*b - 4*a*c
---> 11     sol1 = (-b + math.sqrt(disc)) / (2*a)
     12     sol2 = (-b - math.sqrt(disc)) / (2*a)
     13     return sol1, sol2

ZeroDivisionError: float division by zero
```

In [4]:
```
def fibonacci(n):
    """This function returns the n-th fibonacci number
    @type n: int
    @rtype: int
    @precondition: n>0
    """
    phi = (1 + math.sqrt(5)) / 2
    psi = (1 - math.sqrt(5)) / 2
    return int(round( (phi**n - psi**n) / math.sqrt(5) )) # do not forget the int(...)
 and round functions, otherwise it will be a real number.

fibonacci(1), fibonacci(2), fibonacci(3), fibonacci(4), fibonacci(5), fibonacci(6)
```

Out[4]:
```
(1, 1, 2, 3, 5, 8)
```

Errores comunes

Se nos olvida el return

In [11]:
```
def triangular_number(n):
    """This function computes the n-th triangular number
    @type n: int
    @rtype: int
    @precondition: n>0
    """
    ntriag = (n * (n +1)) // 2
```

In [13]:
```
n = triangular_number(5)
n * 2
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-13-bc0e8e289781> in <module>()
      1 n = triangular_number(5)
----> 2 n * 2

TypeError: unsupported operand type(s) for *: 'NoneType' and 'int'
```

La función devuelve el valor None y None no se puede multiplcar por un entero.

Errores en el paso de valores numéricos

In [20]:
```
def mean(a, b):
    """This function computes the mean of a and b
    @type a: real
    @type b: real
    @rtype: real"""
    return (a + b) / 2

mean(1, 10) # In this case (a+b)/2 is the integer división of 11 and 2, that is  5.
            # This call does not fullfil the requisites and it des not work properly.
```

Out[20]:
```
5
```

In [21]:
```
mean(1.0, 10.0) # this call is correct.
```

Out[21]:
```
5.5
```

In []:
```

```