

# Parameters

Parameters are the objects you can pass to [Functions](#), [Closures](#), or [Callable Types](#) which handle [Events](#).

A parameter is declared with a name, followed by "as", followed by the type: paramname as type. If there is no "as type", then the type is assumed to be type object.

## Examples

### Method/Function example

```
def mymethod(x as int, y as long):  
    ...
```

### Closure

```
c = def(z):  
    print z  
  
obj = "a string"  
c(obj)
```

### Callable Type + Event

```
import System

class Sandwich:
    event Eating as EatingEvent
    callable EatingEvent(sammich as object,
type as string)

    def Eat():
        Eating(self, "Turkey sammich.")

turkeyAndSwiss = Sandwich()
turkeyAndSwiss.Eating += def(obj, sammich):
    print "You're eating a $sammich! It must be
good."
turkeyAndSwiss.Eat()
```

## Variable number of parameters

Boo allows you to call or declare methods that accept a variable (unknown) number of parameters.

You add an asterisk (\*) before the parameter name to signify that it holds multiple parameter values. If there is no 'as type', the type is assumed to be an array of objects: (object). You can declare the type as any array type. For example (int) if your method only accepts int parameters.

Here is an example:

```
def mymethod(x as int, *rest):
    print "first arg:", x
    for item in rest:
        print "extra param:", item
    print

mymethod(1, "a", "b", "c")
mymethod(2, 3, 4, 5, 6, 7)
```

Some boo builtins accept a variable number of parameters, like `matrix()` and `ICallable.Call`.

## ByRef parameters

Add a "ref" keyword before the parameter name to make a parameter be passed by reference instead of by value. This allows you to change a variable's value outside of the context where it is being used. Some examples:

### Basic byref example:

```
def dobyref(ref x as int):
    x = 4

x = 1
print x //-->1
dobyref(x)
print x //-->4
```

### DllImport example:

Wrapping a native method that takes a parameter by reference.

```

import System.Windows.Forms from
System.Windows.Forms
import System.Drawing from System.Drawing
import System.Runtime.InteropServices

class ExtTextBox(TextBox):
    //must be static
    [DllImport("user32")]
    static def GetCaretPos(ref p as Point):
        pass

f = Form(Text: "byref test")
t = ExtTextBox()
f.Controls.Add(t)

b = Button(Text: "GetCaretPos")
b.Click += do:
    p = Point(0,0)
    t.GetCaretPos(p)
    MessageBox.Show(p.X.ToString())
b.Location = Point(0,100)
f.Controls.Add(b)
Application.Run(f)

```

See also tests/testcases/integration/byref\*.boo in the boo source distribution.