

Overloading Numerical Objects/Classes

As we said, in some cases,
it may be meaningful to discuss
what $+$, $-$, \div , $*$, \dots mean for
a class. Just like for logical
operators, there is a way that we
can define/redefine what these
operations mean. $\underbrace{\hspace{1cm}} = \text{"overload"}$

Suppose that we wanted to define
a "time difference":

$$\text{time 4} = \text{time 3} - \text{time 1}$$

At the moment, this is not clearly defined,
we will get a run-time error if

And we can
we try to execute the time.py code.

```
def __sub__(self, other):  
    time_diff = Time(0,0)  
    if (self.minutes >= other.minutes):  
        time_diff.minutes  
            = self.minutes - other.  
                minutes  
        time_diff.hours = self.hours  
            - other.hours  
    else:  
        time_diff.minutes =  
            self.minutes - other.  
                minutes  
            + 60  
        time_diff.hours =  
            self.hours - other.hours  
            - 1  
    return time_diff
```

Other methods for numerical types

--add--	(self, other)	+
--sub--	(self, other)	-
--mul--	(self, other)	*
--truediv--	(self, other)	/
--floordiv--	(self, other)	// (floor)
--mod--	(self, other)	%
--pow--	(self, other)	**
--and--	(self, other)	Logical and
--or--	(self, other)	Logical or
--abs--	(self)	abs ()
--int--	(self)	int ()
--float--	(self)	float ()
