The Descriptor Protocol

The protocol to create a descriptor is really quite easy. You only need to define one or more of the following:

> - __get__(self, obj, type=None), returns value
> - __set__(self, obj, value), returns None
> - __delete__(self, obj), returns None

Once you've defined at least one, you have created a descriptor. If you can you define both __get__ and __set__, you will have created a data descriptor. A descriptor with only __get__() defined are known as non-data descriptors and are usually used for methods. The reason for this distinction in descriptor types is that if an instance's dictionary happens to have a data descriptor, the descriptor will take precedence during the lookup. If the instance's dictionary has an entry that matches up with a non-data descriptor, then the dictionary's own entry will take precedence over the descriptor.

You can also create a read-only descriptor if you define both __get__ and __set__, but raise an **AttributeError** when the __set__ method is called.