



## 2.2 Reading and writing IPTC tags

Reading and writing IPTC tags works pretty much the same way as with EXIF tags. Let's retrieve the list of all available IPTC tags in the image:

```
>>> metadata.iptc_keys
['Iptc.Application2.Caption',
 'Iptc.Application2.Writer',
 'Iptc.Application2.Byline',
 'Iptc.Application2.ObjectName',
 'Iptc.Application2.DateCreated',
 'Iptc.Application2.City',
 'Iptc.Application2.ProvinceState',
 'Iptc.Application2.CountryName',
 'Iptc.Application2.Category',
 'Iptc.Application2.Keywords',
 'Iptc.Application2.Copyright']
```

Each of those tags can be accessed with the `[]` operator on the metadata:

```
>>> tag = metadata['Iptc.Application2.DateCreated']
```

An IPTC tag always has a list of values rather than a single value. This is because some tags have a repeatable character. Tags that are not repeatable only hold one value in their list of values.

Check the `repeatable` attribute to know whether a tag can hold more than one value:

```
>>> tag.repeatable
False
```

As with EXIF tags, the values of an `IptcTag` object can be accessed in two different ways: with the `raw_value` and with the `value` attributes:

```
>>> tag.raw_value
['2004-07-13']

>>> tag.value
[datetime.date(2004, 7, 13)]
```

Note that querying the values of a tag may raise an `IptcValueError` if the format of the raw values is invalid according to the IPTC specification (may happen if it was written by other software that implements the specification in a broken manner), or if pyexiv2 doesn't know how to convert it to a convenient python object.

Now let's modify the values of the tag and write it back to the file:

```
>>> tag.value = [datetime.date.today()]

>>> metadata.write()
```

Similarly to reading the values of a tag, one can set either the `raw_value` or the `value` (which will be automatically converted to correctly formatted byte strings by pyexiv2).

You can also add new tags to the metadata by providing a valid key and values pair (see exiv2's documentation for a list of valid [IPTC tags](#)):

```
>>> key = 'Iptc.Application2.Contact'
>>> values = ['John', 'Paul', 'Ringo', 'George']
>>> metadata[key] = pyexiv2.IptcTag(key, values)
```