Methods

ما تا الان چند مثال ساده از متدها دیدیم. اگر بخاطر نمیاورید، نگاهی به بحثمان در انواع متغیر در پایتون بیندازید. متدها توابعی هستند که درون یک شیء جاساز شده‌اند. در بخش‌های پیشرفته‌تر این درس یاد خواهیم گرفت که چگونه یک شی و کلاس را بنویسیم و برای آن متد تعریف کنیم. متدها عملیات مشخصی را روی یک شیء انجام می‌دهند، و می‌توانند ورودی بگیرند و شی را تغییر دهند و یا فقط یک خروجی برگردانند. در این بخش فقط یک نگاه کلی به متدها می‌اندازیم. و توضیحات بیشتر را به بخش برنامه نویسی شیء گرا اختصاص خواهیم داد. متدها به صورت زیر در دسترس ما قرار می‌گیرند:

```
object.method(arg1,arg2,etc...)
```

البته بعدا خواهید دید که همه متدها یک ورودی self را به معنی خود آن شیء می‌گیرند که هنگام فراخوانی ما آن را نمی‌بینیم. حالا بیایید یک نگاه سریع به چند مثال در پایتون بیندازیم. فرض کنید شیءای که درباره‌اش صحبت می‌کنیم یک list باشد:

In [1]:
```
# Create a simple list
lst = [1,2,3,4,5]
```

In [2]:
```
# See what methods built into the list:
help(lst)
```

```
Help on list object:

class list(object)
 |  list(iterable=(), /)
 |
 |  Built-in mutable sequence.
 |
 |  If no argument is given, the constructor creates a new empty list.
 |  The argument must be an iterable if specified.
 |
 |  Methods defined here:
 |
 |  __add__(self, value, /)
 |      Return self+value.
 |
 |  __contains__(self, key, /)
 |      Return key in self.
 |
 |  __delitem__(self, key, /)
 |      Delete self[key].
 |
 |  __eq__(self, value, /)
 |      Return self==value.
 |
 |  __ge__(self, value, /)
 |      Return self>=value.
 |
 |  __getattribute__(self, name, /)
 |      Return getattr(self, name).
 |
 |  __getitem__(...)
 |      x.__getitem__(y) <==> x[y]
 |
 |  __gt__(self, value, /)
 |      Return self>value.
 |
 |  __iadd__(self, value, /)
 |      Implement self+=value.
 |
 |  __imul__(self, value, /)
 |      Implement self*=value.
 |
 |  __init__(self, /, *args, **kwargs)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |
 |  __iter__(self, /)
 |      Implement iter(self).
 |
 |  __le__(self, value, /)
 |      Return self<=value.
 |
 |  __len__(self, /)
 |      Return len(self).
 |
 |  __lt__(self, value, /)
 |      Return self<value.
 |
 |  __mul__(self, value, /)
 |      Return self*value.
 |
 |  __ne__(self, value, /)
 |      Return self!=value.
 |
 |  __repr__(self, /)
 |      Return repr(self).
 |
 |  __reversed__(self, /)
 |      Return a reverse iterator over the list.
```

Fortunately, with iPython and the Jupyter Notebook we can quickly see all the possible methods using the tab key. The methods for a list are:

- append
- count
- extend
- insert
- pop
- remove
- reverse
- sort

Let's try out a few of them:

append() allows us to add elements to the end of a list:

```
In [3]: lst.append(0)

In [4]: lst
Out[4]: [1, 2, 3, 4, 5, 0]
```

Great! Now how about count()? The count() method will count the number of occurrences of an element in a list.

```
In [5]: # Check how many times 2 shows up in the list
        lst.count(2)
Out[5]: 1
```

You can always use Shift+Tab in the Jupyter Notebook to get more help about the method. In general Python you can use the help() function:

```
In [6]: help(lst.count)
        Help on built-in function count:

        count(value, /) method of builtins.list instance
            Return number of occurrences of value.
```

حالا با خیال راحت بقیه متدهای لیست را خودتان بررسی کنید. در ادامه این بخش تمرینی که از شما می‌خواهم قطعا شامل متدهای سایر جنس‌های متغیر است که انتظار دارم در اینترنت آن‌ها را پیدا کنید :)