

```
In [2]: import numpy as np
```

A simple object `Circle`, which has one parameter `radius`. Methods include `diameter`, `circumference`, and `area`.

```
In [58]: class Circle():  
  
    def __init__(self, radius):  
        self.radius = radius  
  
    def diameter(self):  
        return 2*self.radius  
  
    def circumference(self):  
        return 2*np.pi*self.radius  
  
    def area(self):  
        return np.pi*(self.radius*self.radius)
```

Create an instance of a Circle with radius=10.

```
In [59]: mycircle = Circle(radius=10)
```

```
In [60]: mycircle.area()
```

```
Out[60]: 314.1592653589793
```

```
In [61]: mycircle.circumference()
```

```
Out[61]: 62.83185307179586
```

```
In [62]: mycircle.radius = 2
```

```
In [23]: mycircle.area()
```

```
Out[23]: 12.566370614359172
```

Another instance with radius=100.

```
In [63]: bigcircle = Circle(radius=100)
```

```
In [64]: bigcircle.area()
```

```
Out[64]: 31415.926535897932
```

```
In [ ]:
```

Class parameters can be any object - numbers, arrays, and even functions and other classes. Here is an example where `self.calclength` is an external function.

```
In [53]: class Vector2D():  
  
    def __init__(self, x, y):  
        self.x = x  
        self.y = y  
        self.calclength = calclength  
  
    def length(self):  
        return calclength(self.x, self.y)  
#         return np.sqrt(self.x*self.x+self.y*self.y)  
  
    def angleFromx(self):  
        return np.arctan2(self.y, self.x)
```

```
In [54]: def calclength(x, y):  
    return np.sqrt(x*x+y*y)
```

```
In [55]: vec1 = Vector2D(x=1, y=2)
```

```
In [56]: vec1.length()
```

```
Out[56]: 2.23606797749979
```

```
In [57]: vec1.angleFromx()
```

```
Out[57]: 1.1071487177940904
```

```
In [ ]:
```