

In [1]:

```
dir(str)
```

Out[1]:

```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattr__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mod__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmod__',
 '__rmul__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'capitalize',
 'casefold',
 'center',
 'count',
 'encode',
 'endswith',
 'expandtabs',
 'find',
 'format',
 'format_map',
 'index',
 'isalnum',
 'isalpha',
 'isascii',
 'isdecimal',
 'isdigit',
 'isidentifier',
 'islower',
 'isnumeric',
 'isprintable',
 'isspace',
 'istitle',
 'isupper',
 'join',
 'ljust',
 'lower',
```

```
'lstrip',  
'maketrans',  
'partition',  
'replace',  
'rfind',  
'rindex',  
'rjust',  
'rpartition',  
'rsplit',  
'rstrip',  
'split',  
'splitlines',  
'startswith',  
'strip',  
'swapcase',  
'title',  
'translate',  
'upper',  
'zfill']
```

In [13]:

```
class Vector:  
    def __init__(self,a,b):  
        self.a=a  
        self.b=b  
  
    def __repr__(self):  
        return "vector({},{})".format(self.a,self.b)  
  
    def __add__(self,other):  
        x=self.a+other.a  
        y=self.b+other.b  
        return Vector(x,y)
```

In [14]:

```
v1=Vector(2,3)  
v2=Vector(3,5)
```

In [15]:

```
v1
```

Out[15]:

```
vector(2,3)
```

In [16]:

```
v1+v2
```

Out[16]:

```
vector(5,8)
```

In []:

```
class Test:
    def __int__(self):
        self.x=0
class Derived(Test):
    def __init__(self):
        self.y=1
def main():
    b=Deriverd
```