

# A Sample Class



class is a reserved word that defines a template for making objects

Each PartyAnimal object has a bit of code

Tell the an object to run the party() code within it

```
class PartyAnimal:

    def __init__(self):
        self.x = 0

    def party(self) :
        self.x = self.x + 1
        print("So far",self.x)

an = PartyAnimal()

an.party()
an.party()
an.party()
```

When the object is constructed, a specially named method is called to allocate and initialize attributes.

Construct a PartyAnimal object and store in an

PartyAnimal.party(an)

```
class PartyAnimal:

    def __init__(self):
        self.x = 0

    def party(self) :
        self.x = self.x + 1
        print("So far",self.x)

an = PartyAnimal()

an.party()
an.party()
an.party()
```

\$ python party2.py

```
class PartyAnimal:

    def __init__(self):
        self.x = 0

    def party(self) :
        self.x = self.x + 1
        print("So far",self.x)

an = PartyAnimal()

an.party()
an.party()
an.party()
```

\$ python party2.py

an



The diagram shows a green rectangular box representing an object instance. Inside the box, on the left, is the label 'x'. To its right is a yellow rectangular box containing the value '0'. Below these, on the left, is the label 'party()'. To its right is a green rectangular box containing the text 'party()'.

x	0
party()	party()

```
class PartyAnimal:

    def __init__(self):
        self.x = 0

    def party(self) :
        self.x = self.x + 1
        print("So far",self.x)

an = PartyAnimal()

an.party()
an.party()
an.party()
```

\$ python party2.py

So far 1

So far 2

So far 3

an  
self

x

2

party()

# Playing with `dir()` and `type()`

# A Nerdy Way to Find Capabilities

The `dir()` command lists capabilities

Ignore the ones with underscores -  
these are used by Python itself

The rest are real operations that the  
object can perform

It is like `type()` - it tells us something  
\*about\* a variable

```
>>> y = list()
>>> type(y)
<class 'list'>
>>> dir(y)
['__add__', '__class__',
 '__class_getitem__',
 '__contains__', '__delattr__',
 '__delitem__', '__dir__',
 '__doc__', '__eq__', ...
 'append', 'clear', 'copy',
 'count', 'extend', 'index',
 'insert', 'pop', 'remove',
 'reverse', 'sort']
>>>
```

```
class PartyAnimal:

    def __init__(self):
        self.x = 0

    def party(self) :
        self.x = self.x + 1
        print("So far",self.x)

an = PartyAnimal()

print("Type", type(an))
print("Dir ", dir(an))
print ("Type", type(an.x))
print ("Type", type(an.party))
```

We can use `dir()` to find the “capabilities” of our newly created class.

```
$ python party3.py
Type <class '__main__.PartyAnimal'>
Dir  ['__class__', ... 'party', 'x']
Type <class 'int'>
Type <class 'method'>
```

party3.py



# Try dir() with a String

```
>>> x = 'Hello there'
>>> dir(x)
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
 '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__',
 '__getnewargs__', '__gt__', '__hash__', '__init__', '__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', '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']
```

# Object Lifecycle

[http://en.wikipedia.org/wiki/Constructor\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Constructor_(computer_science))



## Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance ([www.dr-chuck.com](http://www.dr-chuck.com)) of the University of Michigan School of Information and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information

... Insert new Contributors here

# Additional Source Information

- "Snowman Cookie Cutter" by Didriks is licensed under CC BY

<https://www.flickr.com/photos/dinnerseries/23570475099>

- Photo from the television program *Lassie*. Lassie watches as Jeff (Tommy Rettig) works on his bike is Public Domain

[https://en.wikipedia.org/wiki/Lassie#/media/File:Lassie\\_and\\_Tommy\\_Rettig\\_1956.JPG](https://en.wikipedia.org/wiki/Lassie#/media/File:Lassie_and_Tommy_Rettig_1956.JPG)