SomeCode

○ **Class definition**

Classes are used to create user defined datatypes.
- By convention, they are capitalized.
- A class is a python object, and is a template used to create class instances. A class instance is created by instantiation (inst = class()).
- Classes can have docstrings.
- Use the pass statement to define a null class.

```
class Dog:
    """ This is a docstring for the Dog class. """
    pass

>>> d = Dog()
>>> Dog.__doc__
>>> ' this is a docstring for the Dog class. '
```

○ **Class variables**

Class variables in python are shared among all of the class instances.
- If you change the class variable with the class object (class.attr = value), the value is changed in all existing and future class instances (Dog.sound = 'yip').
- If you change the variable through a class instance (big_dog.sound = 'growl') a local variable is created for that instance and added to the instance's dictionary.

```
class Dog:
    sound = 'bark'

>>> print Dog.sound bark
>>> big_dog = Dog()
>>> small_dog = Dog()
>>> print big_dog.sound, small_dog.sound bark bark
>>> Dog.sound = 'yip'
>>> print big_dog.sound, small_dog.sound yip yip
>>> big_dog.sound = 'growl'
>>> print big_dog.sound, small_dog.sound growl yip
```

SomeCode

Q Class 243,121 Sign In ⚙

≡ "Class" search + ▥ ☰ C D ☐

○ **Class definition**

Classes are used to create user defined datatypes.
- By convention, they are capitalized.
- A class is a python object, and is a template use
  A class instance is created by instantiation (inst
- Classes can have docstrings.
- Use the pass statement to define a null class.

○ **Class variables**

Class variables in python are shared among all of the
- If you change the class variable with the class ob
  the value is changed in all existing and future cla
  'yip').
- If you change the variable through a class instan
  'growl') a local variable is created for that instanc
  instance's dictionary.

class Dog:
    """ This is a docstring for the Dog class. """

...cstring for the Dog class. '

Sign In

[f] Sign In with Facebook

[🐦] Sign In with Twitter

[g] Sign In with Google

...und bark
...og()
...Dog()
g.sound, small_dog.sound bark bark
...= 'yip'
g.sound, small_dog.sound yip yip
...nd = 'growl'
g.sound, small_dog.sound growl yip

JavaScript OO ▼

➕ ▥ ▤ C D ☐

○ **Snippet title - this is a one-liner that will be searched during user querries**

This is a snippet explanation. It can contain basic formatting to help with redability such as bold, italic, bullets, indention.
- Bold and italic text
- Bullets
- Indention

The text font size and color is fixed as part of the apps standard look and feel.

```
def snippet()
    """ This is the snippet code. The code will
    have syntax highlighting according
    to the language of the snippet. All snippets
    have three parts: 1) the snippet title,
    2) the code, and 3) an optional explanation. """
    pass
```

○ **Class definition**

Classes are used to create user defined datatypes.
- By convention, they are capitalized.
- A class is a python object, and is a template used to create class instances. A class instance is created by instantiation (inst = class()).
- Classes can have docstrings.
- Use the pass statement to define a null class.

```
class Dog:
    """ This is a docstring for the Dog class. """
    pass

>>> d = Dog()
>>> Dog.__doc__
>>> ' this is a docstring for the Dog class. '
```

○ **Class variables**

Class variables in python are shared among all of the class instances.
- If you change the class variable with the class object (class.attr = value), the value is changed in all existing and future class instances (Dog.sound = 'yip').
- If you change the variable through a class instance (big_dog.sound = 'growl') a local variable is created for that instance and added to the instance's dictionary.

```
class Dog:
    sound = 'bark'

>>> print Dog.sound bark
>>> big_dog = Dog()
>>> small_dog = Dog()
>>> print big_dog.sound, small_dog.sound bark bark
>>> Dog.sound = 'yip'
>>> print big_dog.sound, small_dog.sound yip yip
>>> big_dog.sound = 'growl'
>>> print big_dog.sound, small_dog.sound growl yip
```

JavaScript OO ▾   ➕ ▥ ▤ C D □

👁   Topic ▾   Launguage ▾   Cancel   Save

○ Add snippet title here

Add snippet description here   Add snippet code here

⬍   ⬍

○ **Snippet title - this is a one-liner that will be searched during user querries**

This is a snippet explanation. It can contain basic formatting to help with redability such as bold, italic, bullets, indention.
- Bold and italic text
- Bullets
- Indention

The text font size and color is fixed as part of the apps standard look and feel.

⊕ ⊖ ▥ ▤ 👁 ✕

```
def snippet()
    """ This is the snippet code. The code will
    have syntax highlighting according
    to the language of the snippet. All snippets
    have three parts: 1) the snippet title,
    2) the code, and 3) an optional explanation. """
    pass
```

○ **Class definition**

Classes are used to create user defined datatypes.
- By convention, they are capitalized.
- A class is a python object, and is a template used to create class instances. A class instance is created by instantiation (inst = class()).
- Classes can have docstrings.
- Use the pass statement to define a null class.

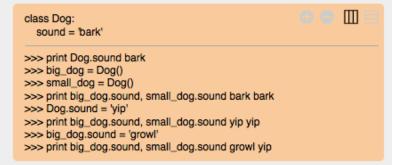⊕ ⊖ ▥ ▤ 👁 ✕

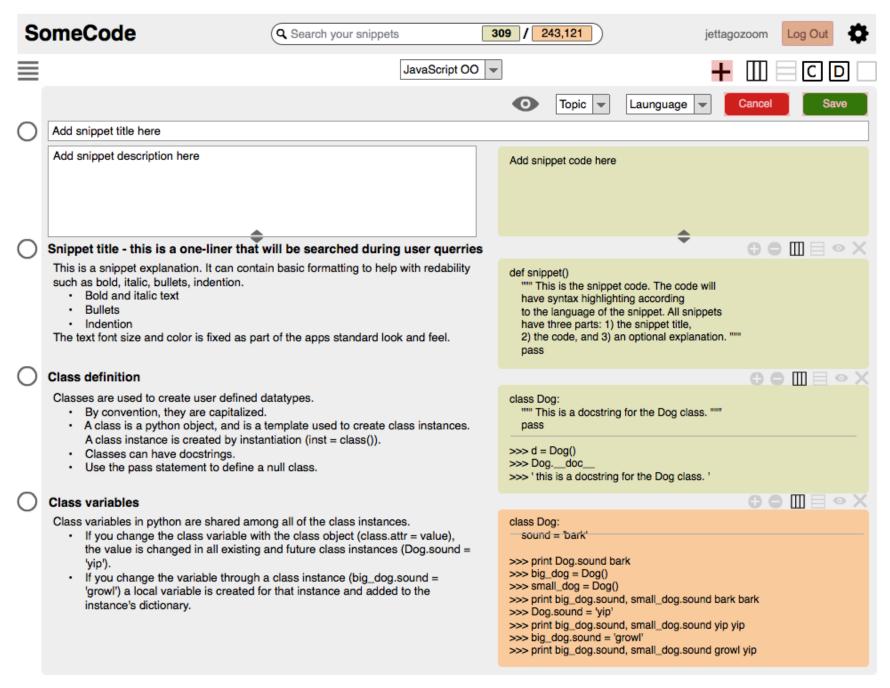```
class Dog:
    """ This is a docstring for the Dog class. """
    pass

>>> d = Dog()
>>> Dog.__doc__
>>> ' this is a docstring for the Dog class. '
```

○ **Class variables**

Class variables in python are shared among all of the class instances.
- If you change the class variable with the class object (class.attr = value), the value is changed in all existing and future class instances (Dog.sound = 'yip').
- If you change the variable through a class instance (big_dog.sound = 'growl') a local variable is created for that instance and added to the instance's dictionary.

⊕ ⊖ ▥ ▤ 👁 ✕

```
class Dog:
    sound = 'bark'

>>> print Dog.sound bark
>>> big_dog = Dog()
>>> small_dog = Dog()
>>> print big_dog.sound, small_dog.sound bark bark
>>> Dog.sound = 'yip'
>>> print big_dog.sound, small_dog.sound yip yip
>>> big_dog.sound = 'growl'
>>> print big_dog.sound, small_dog.sound growl yip
```

Topic ▾   Launguage ▾   Cancel   Save

Add snippet title here

Add snippet description here

Add snippet code here

>>> print big_dog.sound, small_dog.sound growl yip

SomeCode

JavaScript OO

Topics   +

General	87
Common C++ Syntax	14
Common Javascript Syntax	22
Common jQuery	34
JavaScript OO	**12**
Python OO	9
HTML	34

Language

Javascript	34
Python	9
C++	14
jQuery	34
HTML	34
CSS	97
Less	42
jinja2	17

Snippet title - this is a one-liner that will be searched during user querries

This is a snippet explanation. It can contain basic formatting to help with redability such as bold, italic, bullets, indention.
- Bold and italic text
- Bullets
- Indention

The text font size and color is fixed as part of the apps standard look and feel.

```
def snippet()
    """ This is the snippet code. The code will
    have syntax highlighting according
    to the language of the snippet. All snippets
    have three parts: 1) the snippet title,
    2) the code, and 3) an optional explanation. """
    pass
```

Class definition

Classes are used to create user defined datatypes.
- By convention, they are capitalized.
- A class is a python object, and is a template used to create class instances. A class instance is created by instantiation (inst = class()).
- Classes can have docstrings.
- Use the pass statement to define a null class.

```
class Dog:
    """ This is a docstring for the Dog class. """
    pass

>>> d = Dog()
>>> Dog.__doc__
>>> ' this is a docstring for the Dog class. '
```

Class variables

Class variables in python are shared among all of the class instances.
- If you change the class variable with the class object (class.attr = value), the value is changed in all existing and future class instances (Dog.sound = 'yip').
- If you change the variable through a class instance (big_dog.sound = 'growl') a local variable is created for that instance and added to the instance's dictionary.

```
class Dog:
    sound = 'bark'

>>> print Dog.sound
bark
>>> big_dog = Dog()
>>> small_dog = Dog()
>>> print big_dog.sound, small_dog.sound
bark bark
>>> Dog.sound = 'yip'
>>> print big_dog.sound, small_dog.sound
yip yip
>>> big_dog.sound = 'growl'
>>> print big_dog.sound, small_dog.sound
```

SomeCode

JavaScript OO

Topics

General	87
Common C++ Syntax	14
Common Javascript Syntax	22
Common jQuery	34
JavaScript OO	12
Python OO	9
HTML	34

Language

Javascript	34
Python	9
C++	14
jQuery	34
HTML	34
CSS	97
Less	42
jinja2	17

Snippet title - this is a one-liner that will be searched during user querries

This is a snippet explanation. It can contain basic formatting to help with redability such as bold, italic, bullets, indention.
- Bold and italic text
- Bullets
- Indention

The text font size and color is fixed as part of the apps standard look and feel.

```
def snippet()
    """ This is the snippet code. The code will
        have syntax highlighting according
        to the language of the snippet. All snippets
        have three parts: 1) the snippet title,
        2) the code, and 3) an optional explanation.
    """
    pass
```

Class definition

Classes are used to create user defined datatypes.
- By convention, they are capitalized.
- A class is a python object, and is a template used to create class instances. A class instance is created by instantiation (inst = class()).
- Classes can have docstrings.
- Use the pass statement to define a null class.

```
class Dog:
    """ This is a docstring for the Dog class. """
    pass

>>> d = Dog()
>>> Dog.__doc__
>>> ' this is a docstring for the Dog class. '
```

Class variables

Class variables in python are shared among all of the class instances.
- If you change the class variable with the class object (class.attr = value), the value is changed in all existing and future class instances (Dog.sound = 'yip').
- If you change the variable through a class instance (big_dog.sound = 'growl') a local variable is created for that instance and added to the instance's dictionary.

```
class Dog:
    sound = 'bark'

>>> print Dog.sound bark
>>> big_dog = Dog()
>>> small_dog = Dog()
>>> print big_dog.sound, small_dog.sound bark bark
>>> Dog.sound = 'yip'
```

≡ JavaScript OO ▾ + ▨ ▤ C D ▢

○ **Snippet title - this is a one-liner that will be searched during user querries**

This is a snippet explanation. It can contain basic formatting to help with redability such as bold, italic, bullets, indention.
- Bold and italic text
- Bullets
- Indention

The text font size and color is fixed as part of the apps standard look and feel.

```
def snippet()
    """ This is the snippet code. The code will
    have syntax highlighting according
    to the language of the snippet. All snippets
    have three parts: 1) the snippet title,
    2) the code, and 3) an optional explanation. """
    pass
```

○ **Class variables**

Class variables in python are shared among all of the class instances.
- If you change the class variable with the class object (class.attr = value), the value is changed in all existing and future class instances (Dog.sound = 'yip').
- If you change the variable through a class instance (big_dog.sound = 'growl') a local variable is created for that instance and added to the instance's dictionary.

```
class Dog:
    sound = 'bark'

>>> print Dog.sound bark
>>> big_dog = Dog()
>>> small_dog = Dog()
>>> print big_dog.sound, small_dog.sound bark bark
>>> Dog.sound = 'yip'
>>> print big_dog.sound, small_dog.sound yip yip
>>> big_dog.sound = 'growl'
>>> print big_dog.sound, small_dog.sound growl yip
```
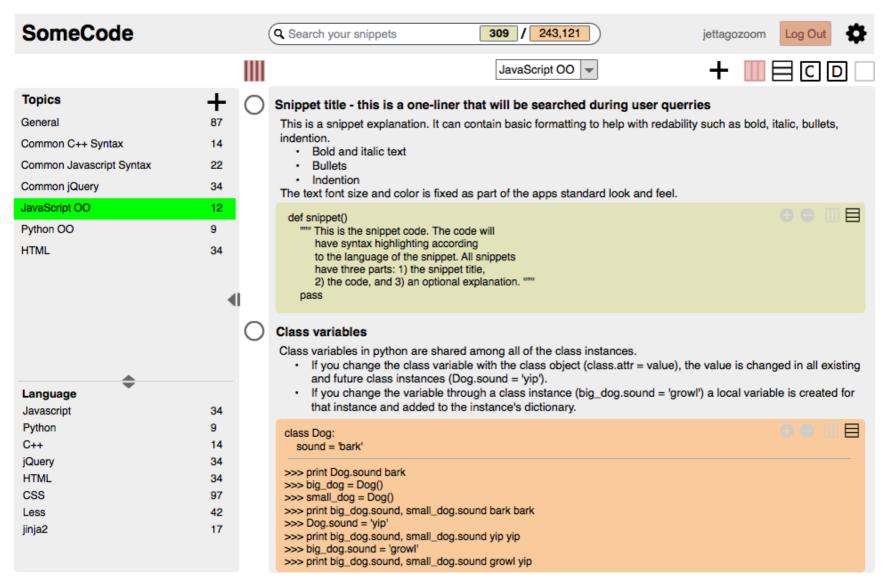
JavaScript OO ▼   ＋ ▋▋ ☰ C D ☐

Topics   ＋

General	87
Common C++ Syntax	14
Common Javascript Syntax	22
Common jQuery	34
JavaScript OO	**12**
Python OO	9
HTML	34

◀

Language

Javascript	34
Python	9
C++	14
jQuery	34
HTML	34
CSS	97
Less	42
jinja2	17

○ **Snippet title - this is a one-liner that will be searched during user querries**

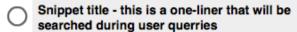This is a snippet explanation. It can contain basic formatting to help with redability such as bold, italic, bullets, indention.
- Bold and italic text
- Bullets
- Indention

The text font size and color is fixed as part of the apps standard look and feel.

```
def snippet()
    """ This is the snippet code. The code will
        have syntax highlighting according
        to the language of the snippet. All snippets
        have three parts: 1) the snippet title,
        2) the code, and 3) an optional explanation. """
    pass
```

○ **Class variables**

Class variables in python are shared among all of the class instances.
- If you change the class variable with the class object (class.attr = value), the value is changed in all existing and future class instances (Dog.sound = 'yip').
- If you change the variable through a class instance (big_dog.sound = 'growl') a local variable is created for that instance and added to the instance's dictionary.

```
class Dog:
    sound = 'bark'

>>> print Dog.sound bark
>>> big_dog = Dog()
>>> small_dog = Dog()
>>> print big_dog.sound, small_dog.sound bark bark
>>> Dog.sound = 'yip'
>>> print big_dog.sound, small_dog.sound yip yip
>>> big_dog.sound = 'growl'
>>> print big_dog.sound, small_dog.sound growl yip
```

JavaScript OO ▾ **+** ▤ C D ▨

○ **Snippet title - this is a one-liner that will be searched during user querries**

This is a snippet explanation. It can contain basic formatting to help with redability such as bold, italic, bullets, indention.
- Bold and italic text
- Bullets
- Indention

The text font size and color is fixed as part of the apps standard look and feel.

```
def snippet()
    """ This is the snippet code. The code will
        have syntax highlighting according
        to the language of the snippet. All snippets
        have three parts: 1) the snippet title,
        2) the code, and 3) an optional explanation. """
    pass
```

○ **Class variables**
Class variables in python are shared among all of the class instances.
- If you change the class variable with the class object (class.attr = value), the value is changed in all existing and future class

```
class Dog:
    sound = 'bark'

>>> print Dog.sound bark
>>> big_dog = Dog()
>>> small_dog = Dog()
>>> print big_dog.sound, small_dog.sound bark
bark
>>> Dog.sound = 'yip'
>>> print big_dog.sound, small_dog.sound yip yip
>>> big_dog.sound = 'growl'
```

basic formatting to help with redability such as bold, italic, bullets, indention.
- Bold and italic text
- Bullets
- Indention

The text font size and color is fixed as part of the apps standard look and feel.

```python
def snippet()
    """ This is the snippet code. The code will
        have syntax highlighting according
        to the language of the snippet. All snippets
        have three parts: 1) the snippet title,
        2) the code, and 3) an optional explanation. """
    pass
```

○ **Class variables**
Class variables in python are shared among all of the class instances.
- If you change the class variable with the class object (class.attr = value), the value is changed in all existing and future class

```python
class Dog:
    sound = 'bark'
```

```python
>>> print Dog.sound bark
>>> big_dog = Dog()
>>> small_dog = Dog()
>>> print big_dog.sound, small_dog.sound bark
bark
>>> Dog.sound = 'yip'
>>> print big_dog.sound, small_dog.sound yip yip
>>> big_dog.sound = 'growl'
```

Search your snippets

JavaScript OO ▼   **+**

- ○ **Snippet title - this is a one-liner that will be searched during user querries**
- ○ **Class definition**
- ○ **Class variables**
- ○ **Built-in Class Attributes**
- ○ **Built-in Instance Attributes**
- ○ **Instance Variables/Methods, Ctor/Dtor**
- ○ **Inheritance, Public/Private Attrs**

Account

open/create a bank account

Log in to bank account

create multiple savings buckets

associate a debit card with account

access bank balance from a banking app

be notified via text or email when account is running low

be able to close banking account

Banking App

Deposits / Transfers / Withdrawls

deposit money into checking account

transfer money from my checking account to my savings account buckets

do in-store purchases from a banking app

withdrawal money from checking or savings accounts

withdrawal money from accounts through an ATM

get an accurate monthly budget from a banking app

view all transactions

with each transaction associate the date, time, amount, and place

Transactions

Columnar Configuration

Row Configuration