Cheatography

COSC-3353-Ch3at Sheet Cheat Sheet

by FooFax (foobar) via cheatography.com/16755/cs/1667/

JavaScript

```
// Define a new class
function Person() {}
// Create a new instance
var person1 = new Person();
// Inheri tance
Person.pr oto typ e.walk =
function() {
        ale rt( 'Wa lki ng!');
};
```

Simple Alert

```
<html>
          <bo dy>
                    <script
type="t ext /ja vas cri pt">
                              -
ale rt( 'Hello JavaSc rip t!');
                    </s cri pt>
          </b ody>
</h tml>
```

Simple JavaScript

```
<html>
          <bo dy>
                    <script
type="t ext /ja vas cri pt">
                              -
ale rt( 'Hello JavaSc rip t!');
                              -
doc ume nt.w ri te( 'Ja vaS -
cript rulez!');
                    </s cri pt>
          </b ody>
</h tml>
```

Steps of Ajax Operation

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the web server.
5. The web server returns the result containing XML document.
6. The XMLHttpRequest object calls the callback() function and processes the result.
7. The HTML DOM is updated.

Accessing Elements

```
// By Id
docume nt.g et Ele men tBy -
Id( " som e_i d");
// By class name
docume nt.g et Ele men tsB yCl -
ass Nam e("s ome _cl ass ");
// By tag name
el.get Ele men tsT agN ame -
("im g");
```

CSS

Inline - by using the style attribute inside HTML elements

Internal - by using a <style> element in the <head> section

External - by using a <link> element to link to an external CSS file

Firebug

* Supports breakpoints, watches, JavaScript, CSS, HTML
* Useful for CSS and HTML
* Edit documents in realtime
* Shows how CSS rules apply to elements
* Shows Ajax requests and responses
* Firebug is written mostly in JavaScript

Notes

This is called encapsulation, by which every class inherits the methods of its parent and only needs to define things it wishes to change

Abstraction is a mechanism that permits modeling the current part of the working problem. This can be achieved by inheritance (specialization), or composition. JavaScript achieves specialization by inheritance, and composition by letting instances of classes be the values of attributes of other objects.

Just like all methods and properties are defined inside the prototype property, different classes can define methods with the same name; methods are scoped to the class in which they’re defined. This is only true when the two classes do not hold a parent-child relation