

MapReduce Simulator Cheat Sheet

Basics

```
var myVar;  
    declare variable in current scope  
  
var myObj = {  
  'property': value,  
  'otherProperty': otherValue  
};  
    create a simple object  
  
var myArr = [ el1... ];  
    create an array  
  
function abc( param*... ) {}  
    function defintion
```

Type Conversions

```
typeof myVar  
    get type of variable  
  
myStr = '' + myVar  
    convert variable to string  
  
myNumber = parseInt( myStr, radix* )  
    parse integer from a string  
  
myNumber = parseFloat( myStr )  
    parse float from a string  
  
myBool = !!myVar  
    convert variable to boolean
```

Objects

```
myArr = Object.keys( obj )  
    get a list of object properties  
  
myBool = obj.hasOwnProperty( prop )  
    test for property existence
```

Loops

```
while( condition ) { }  
    while loop  
  
do { } while( condition )  
    do while loop  
  
for( start; cond; final ) { }  
    for loop
```

Array Methods

Non Modifying
myArr.length size of the array
newArr = myArr.concat(otherArr) concatenate arrays
myString = myArr.join(glue*) join the elements of an array
mySubArr = myArr.slice(start*, end*) get shallow copy of an part of the array
myPos = myArr.indexOf(el) get first index of element within array
myPos = myArr.lastIndexOf(el) get last index of element within array

Modifying
myArr.pop() remove and get the last element
myArr.push(el) add an element to the end
myArr.unshift() remove and get the first element
myArr.shift(el) add an element to the start
myArr.sort(cb*) sort the array
myArr.reverse() reverse the order
myArr.splice(start, count, el*...) remove elements and replace by new ones

String Methods

Non Modifying
myStr.length length of string
myStr.charAt(pos) get character at position
myNewString = myStr + myOtherStr; myNewString = myStr.concat(myOtherStr) string concatenation
myPos = myStr.indexOf(char) get first position of character
myPos = myStr.lastIndexOf(char) get last position of character
myNum = myStr.localeCompare(myOtherStr) compare two strings
myBool = myStr.match(regexp) match a regular expression against a string
myNewStr = myStr.replace(needle regexp, replacement) replace a subtring by a new substring
myNewStr = myStr.slice(start, end*) myNewStr = myStr.substring(start, end*) myNewStr = myStr.substr(start, length*) extract a substring
myArr = myStr.split(needle* regexp*, limit*) split a string into substrings
myNewStr = myStr.toLowerCase() myNewStr = myStr.toUpperCase() convert to lower/upper case
myNewStr = myStr.trim() remove whitespace characters at start and end

Regular expressions

```
var myRegExp = /.../;  
    create a regular expression  
  
myBool = regexp.test( str )  
    test a string for the given expression  
  
myArr = str.match( regexp )  
    retrieve the matches in a string  
  
myNum = str.search( regexp )  
    search for the first match inside the string
```

Serializing

```
myObj = JSON.parse( myStr )  
    parse JSON string  
  
myStr = JSON.stringify( myObj,  
                        filter*, spacer* )  
    serialize object to string
```

MapReduce emits

```
emit( key, value )  
    general emit function  
  
emit( line )  
    emit in outputFormat
```

Legend

parameter*
optional parameter

parameter...
undefined number of instances

Links & Resources

www.webplatform.org
WebPlatform

developer.mozilla.org
Mozilla Developer Network

Credits

Sirko Schindler
<http://mr.sirko.org>

