JS EQUIVALENTS

#	*//*	**is**	*===*
and	*&&*	**isnt**	*!==*
or	*||*	**not**	*!*
true, yes, on	*true*	**false, no, off**	*false*
@	*this*	@prop	*this.prop*
v **||=** x	*v = v || x*	**	*exponential operator*

ARRAYS

```
arr = [
   1
   2
   3
]
```

```
[1..6]
```
***creates a range of [1, 2, 3, 4, 5, 6]*

```
[1...6]
```
***non-inclusive: [1, 2, 3, 4, 5]*

```
myArr[2..3]
```
***selects the second and third items*

OBJECTS

```
customer = name:"Bob", age:34

client =
   name:"Anne"
   age:27
```

NEAT TRICKS

```
console.log "Hey #{name}, what's up? I
   was hoping I would find you here."
```
***easy variable escaping with double quotes,*
block text ignores whitespace.

```
elvis?
```
***true if it exists (i.e. not undefined or null)*

```
couldBeFalseOr0 ?= default
```

```
`
write.javascript();
`
```

CONDITIONALS

```
if name is "Steve"
   console.log name

if 4 is 4 then alert "The same!"

mood = "happy" unless monday

coat = if cold
   "parka"
else
   "jacket"
```

LOOPS

```
for number in array
   number + 1
```
***returns an array with each iteration's result*

```
nums = (num + 1 for num in arr)

for key, value of myObject
   console.log key, value

count += 1 until count is 10
```
***'until' is 'while not', loop body can proceed loop*

FUNCTIONS

```
sum = (x, y) ->
   x + y
```
***can return last logically executed line without 'return'*

```
total = sum x, y
```
***can call without parentheses*

```
returnDefault = (default=0) ->
   default
```
***can set defaults in the parameters list*

```
list = (first, others...) ->
   alert first, others
```
***splats (var…) are the tail end of JS's 'arguments' array*

```
thisFunc = =>
   console.log @
```
***fat arrow passes in 'this' from outer scope*

CLASSES

```
class Car
   drive: (mpg) ->

class Hybrid extends Car
   drive: ->
      super 50
```
***extend both parent classes and parent methods,*
uses pseudoclassical class style