Python to Javascript

Syntax Reference

Syntax is the word used to define the rules a language uses to define itself. For example some English syntax rules would be;

1. All sentences start with a capital letter.
2. All sentences end with a period (full stop).
3. A period(.) denotes a rest when spoken.

Often there are multiple ways to get some similar behavior between the languages. You will find the first option most similar between the languages first in their table cell.

Name	Python	Javascript
Comment	`# A single line comment`	`// A JS single line comment`
Variable assignment	`name = 1`	`var name = 1`
Boolean	`True` `False`	`true` `false`
Print to screen	`print('Example something')`	`console.log('Example Something')`
Statement separator	`newline` `;`	

String literal	`"hello world"`
	`'hello world'`
	`"""hello world"""`
	`'''hello world'''`
		`"hello world"`
		`'hello world'`

Array literal	`[1, 2, 'etc']`
	`(1, 2, 'etc')`
	`{1, 2,'etc'}`
		`[1, 2, 'etc']`

Object literal	`{"key": "string value", "number": 1}`	
String index	`"Some Words"[0]`	
Array index	`[1,2,3, "four"][0]`	

Object property	`myObjct = {"name": "demo"}`
	`myObjct["name"]`
		`var myObjct = {"name": "demo"}`
		`myObjct["name"]`
		`o.name`

Name	Python	Javascript

Conditionals	`test = True`
	`if test == True:`
	`  print("correct")`
	`elif test == "never":`
	`  print('never going to run')`
	`else:`
	`  print('Test is false')`
		`var test = true`
		`if(test == true){`
		`  console.log("correct")`
		`} else if (test == "never"){`
		`  console.log("never going to run")`
		`} else {`
		`  console.log("Test is false")`
		`}`

Function definition	`def funcName():`
	`  # Code inside the function must be indented`
		`function funcName(){`
		`  //Code must be inside code blocks closing bracket`
		`}`

Language methods	all lower case, usually short e.g. "hi".upper() # HI	camelCase, more wordy e.g. "hi".toUpperCase() // HI

Code examples

This example is to show how similar yet different javascript is from python

PYTHON

```
def shout(word):
return str(word).upper()
```

JAVASCRIPT

```
function shout(word){
return word.toString().toUpperCase()
}
```

Note how Javascript uses more syntax to create the statement, it only happens to look similar because it can be written that way. The following is exactly the same function. Look at the brackets locations and white-space.

JAVASCRIPT

```
function shout(word) { return word.toUpperCase() }
```

JAVASCRIPT

```
function shout(word)
{      return word.toUpperCase()   }
```

JAVASCRIPT

```
     function shout(word)

{return word.toUpperCase()}
```

All these examples produce this result when called

JAVASCRIPT

```
shout('hi there') // HI THERE
```

Glossary

Memorizing these words isn't important to programming, what is important is recognizing the things that they do. These words become important when you find yourself having to explain your code or thoughts to another human.

Index

Like a book, there's an index field used to quickly reference specific parts, often numbered by page, sometimes a section number. In data structures like Objects and Arrays that hold multiple fields of information you might find the index useful to access.

code	`['a','b','c','d','e']`				
index	0	1	2	3	4
value	a	b	c	d	e

If you want to access an index by name instead of number, you now understand the difference between and array and an object.

Method, definition, function

A block of code that can be re-used, if it's given a name to be reused by we'd start to call it a method. Generally any of those names are fine to use.

Literal

You don't always have to put something in a named variable to use it, especially if it's not going to be reused. Take the following identical code as an example;

JAVASCRIPT

```
var line = "Example sentence." // Declare the variable named "line" and assign it the value of a string literal
console.log(line) // Print out the value of the variable line, which would be a string, assigned earlier
```

JAVASCRIPT

```
console.log("Example sentence.") // Print the literal string of example sentence.
```