	PYTHON	JAVASCRIPT
1. if	**if condition:**	**if (condition) { }**
	Ex.	Ex.
	if 1 < 2:	if (1 < 2) {
	print("ye") # "ye"	console.log("ye"); // "ye"
		}
2. if shorthand	**if condition: __**	**if (condition) __ ;**
	__ = code to run if condiiton True	__ = code to run if condiiton True
	Ex.	Ex.
	if 1 < 2: print("ye") # "ye"	if (1 < 2) console.log("ye"); // "ye"
3. if else	**if condition:**	**if (condition) {**
	else:	**} else {**
	- can NOT have empty code blocks (will get error)	**}**
		- CAN have empty code blocks
	Ex.	Ex.
	if 1 > 2:	if (1 > 2) {
	print("ye")	console.log("ye");
	else:	} else {
	print("nop") # "nop"	console.log("nop"); // "nop"
		}
4. if else shorthand	value_true **if condition else** value_false	**condition ?** value_true **:** value_false
	- can NOT have expressions in value_true or value_false only statements!	- CAN have expressions in value_true or value_false
	- only available in python version 2.5+	- called ternary operator
	Ex1.	Ex1.
	print("ye") if 1 < 2 else print("nop") # "ye"	1 < 2 ? console.log('ye') : console.log('nop'); // "ye"
	Ex2.	Ex2.
	print("ye" if 1 < 2 else "nop") # "ye"	console.log(1 < 2 ? "ye" : "nop"); // "ye"
5. if elif	**if condition:**	**if (condition) {**
	elif condition:	**} else if (condition) {**
		}
	Ex.	Ex.
	if 1 > 2:	if (1 > 2) {
	print("ye")	console.log("ye");
	elif 1 < 2:	} else if (1 < 2) {
	print("elif") # "elif"	console.log("else if"); // "else if"
		}

	PYTHON	JAVASCRIPT
6. if elif else	**if condition:**	**if (condition) {**
	elif condition:	**} else if (condition) {**
	else:	**} else {**
		}
	Ex.	Ex.
	if 1 > 2:	if (1 > 2) {
	print("ye")	console.log("ye");
	elif 1 == 2:	} else if (1 == 2) {
	print("elif")	console.log("else if");
	else:	} else {
	print("else") # "else"	console.log("else"); // "else"
		}
7. Falsey/Truthy Values	**- objects by default are truthy unless empty or the object's bool method returns False**	**- ALL objects are truthy even if empty**
	TRUTHY "0", "-1", 1, -1, "a", [1], { "a": 1 }	TRUTHY "0", "-1", 1, -1, "a", [1], { "a": 1 }, **[ ], { }**
	FALSEY 0, 0.0, **[ ], { }**	**FALSEY 0, 0.0**
8. in	- used to iterate for __ in sequence/dict	**same as python**
	- used to check if a value (or key if dictionary) is present in a sequence, set, dictionary	
	- **DOES** check for objects in objects	- for objects does NOT check for values!
	Ex1.	Ex1.
	list = ["a", 2, [3]]	let arr = ["a", 2, [3]]
	print([3] in list) # True	console.log([3] in arr) // false
	Ex2.	
	list = ["a", [1] , 2]	
	print("b" in list) # False	
	print("a" in list) # True	
	print([1] in list) # True	
	Ex3.	
	dic = { "a": 2, "b": 3 }	
	print("z" in dic) # False	
	print("a" in dic) # True	
9. python expression vs statement	**Expressions**	
	- expressions evaluate to at least one value	
	- expressions are also statements, but not all statements are expressions	
	1 + 2	
	print("ye")	
	"hi"	
	Statements	

	PYTHON	JAVASCRIPT
	- made up of expressions	
	x = 10	
	print("ye")	
	if 1 < 2: x = "yep"	