	PYTHON	**JAVASCRIPT**
1. dictionary	- keys must be "hashable" (ex. *strings, numbers, boolean, None, tuples* (only if all values are immutable)	- keys must be strings! Primitive keys are implicitly coerced into strings
	- indexing into a nonexistent key results in ERROR	- indexing into a nonexistent key does NOT result in error, but undefined
	Ex1.	Ex1.
	dic = { }	obj = { }
	dic["a"] # ERROR- KeyError: 'a'	obj["a"] // undefined
	- can NOT use dot . notation to set/get key value pairs (will get error)	- CAN use dot . notation to set/get key value pairs
	Ex2.	Ex2.
	dic = { "b": 2 }	obj = { "b": 2 }
	dic.a = 1 # ERROR- AttributeError: 'dic' object has no attribute 'a'	obj.a = 1 // { a: 1, b: 2 }
	dic.b # ERROR- AttributeError: 'dic' object has no attribute 'b'	obj.b // 2
	- creating a key using an undefined variable results in error	- creating a key with an undefined variable results in the variable implicitly coerced into a string!
	Ex3.	Ex3.
	dic = { a: 1 } # ERROR- NameError: name 'a' not defined	obj = { a: 1 } // { "a": 1 }
2. functions	**- functions are NOT hoisted**	**- function declarations ARE HOISTED. function expressions are not hoisted**
	Ex1.	Ex1.
	def function_name(arguments):	function printName(arguments) {
	print('yo")	console.log('yo")
	- indentation can be 2+ spaces, but must be uniform for each code block	}
	- if no return value, default return is **None**	- if no return value, default return is **undefined**
	- variables defined in function are **function scoped** (ie. not accessible outside function)	- variables defined in function can be **function or global scoped** depending on keyword (if no keyword, scope is global)
	Ex2.	Ex2.
	def test():	function test(){
	x = 10	x = 10
		}
	test()	test()
	print(x) # ERROR- NameError: name 'x' not defined	console.log(x) // 10
	- function **positional arguments** must be called in order (like JS functions)	
	- function **Keyword Arguments** (aka **Named arugments**) can be called out of order	- does **NOT** have Keyword / Named arguments!! Though we can mimick this by using object destructuring
	Ex3.	
	def print_name_age(name, age):	
	print("hello " + name + "! you are " + str(age) + " years old")	
	print_name_age(age=12, name="harry")	

	PYTHON	**JAVASCRIPT**
	- calling a function with more parameters than defined results in ERROR	- calling a function with more parameters than defined does NOT error
	Ex4.	Ex4.
	def test(x):	function test(x) {
	print(x)	console.log(x)
	test(1, 2, 3) # TypeError: test() takes 1 positional argument but 3 were given	test(1, 2, 3) // 1
	- calling a function with less parameters than defined results in ERROR	- calling a function with less parameters than defined does NOT error
	Ex5.	Ex5.
	def test(x):	function test(x) {
	print("hi")	console.log("hi")
	test() # TypeError test() missing 1 required positional argument: 'x'	test()