**3. Javascript Foundation II**

**2. Execution Context**

**STATEMENT** -- Whenever code is run in JS it is run inside of an execution context.

function() {

this is an execution context

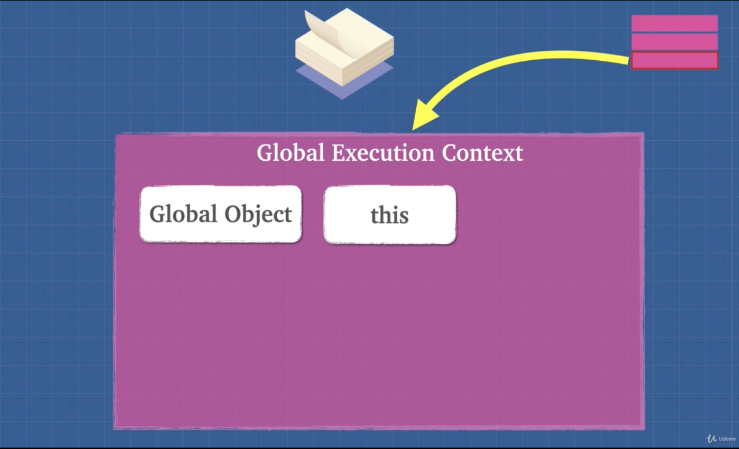
}

* Every function() creates a new execution context and it is pushed on to the stack.

**global() execution context**

* Underneath the hood there is a "global() execution context". After every line is read it gets popped of from the stack.
* global() execution context gives us 2 objects, Global object & this object.

window === this



cntd

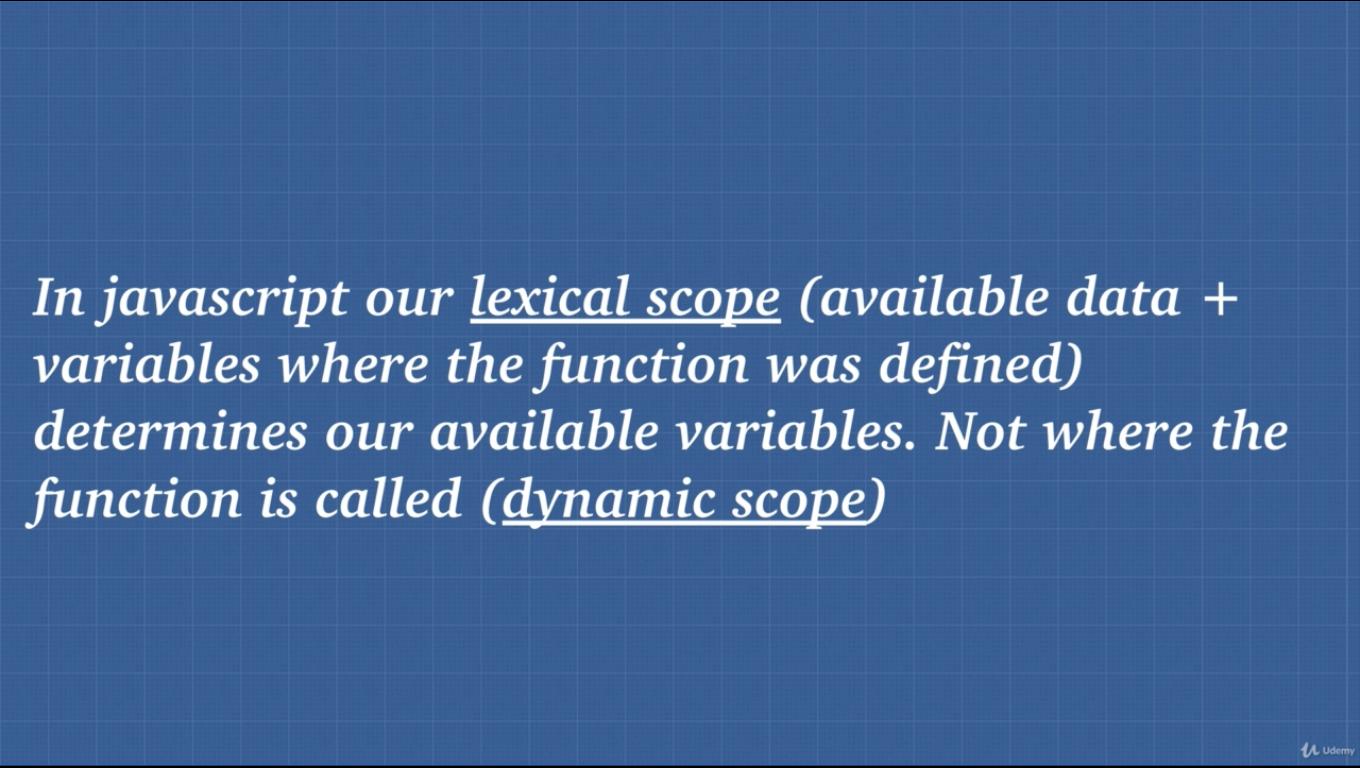
**3. Lexical Environment**

**Lexical Environment** is simply where we write our code.

**STATEMENT** – Execution context tells us which lexical environment is currently running or which planet is currently running.

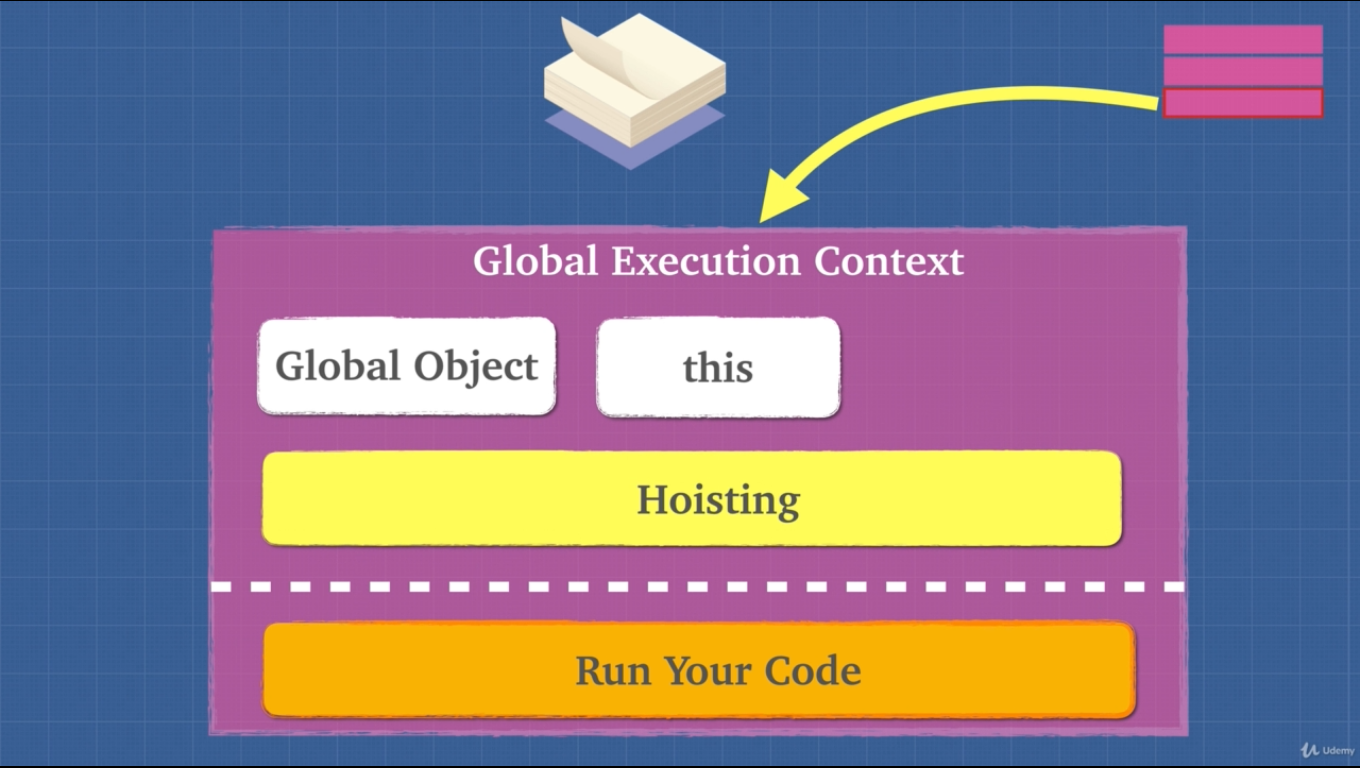


* Every function creates a new lexical environment.
* Lexical scope determines the available data, variables etc. not where the function is called.



cntd

**4. Hoisting**



**Hoisting**: Hoisting is moving the var or function declaration to the top of their respective environment during compilation phase.

* Hoisting is unique only to javascript.
* var and function expression is partially hoisted and function declaration is hoisted.

**5. Exercise Hoisting**

* For var hoisting ignores same variable name if found, only takes the first one because it is partially hoisted.
* For function, hoisting happens fully. If same function is found it overrides the previous function in memory.

**6. Exercise Hoisting 2**

* When we run a function a new execution context is created.

function() {

Here is a new execution context.

}

**STATEMENT** – Hoisting is bad practice. We can avoid hoisting using “const” or “let”.

**8. Function Invocation**

Function invocation/Call/Execution are the same thing

function\_name()

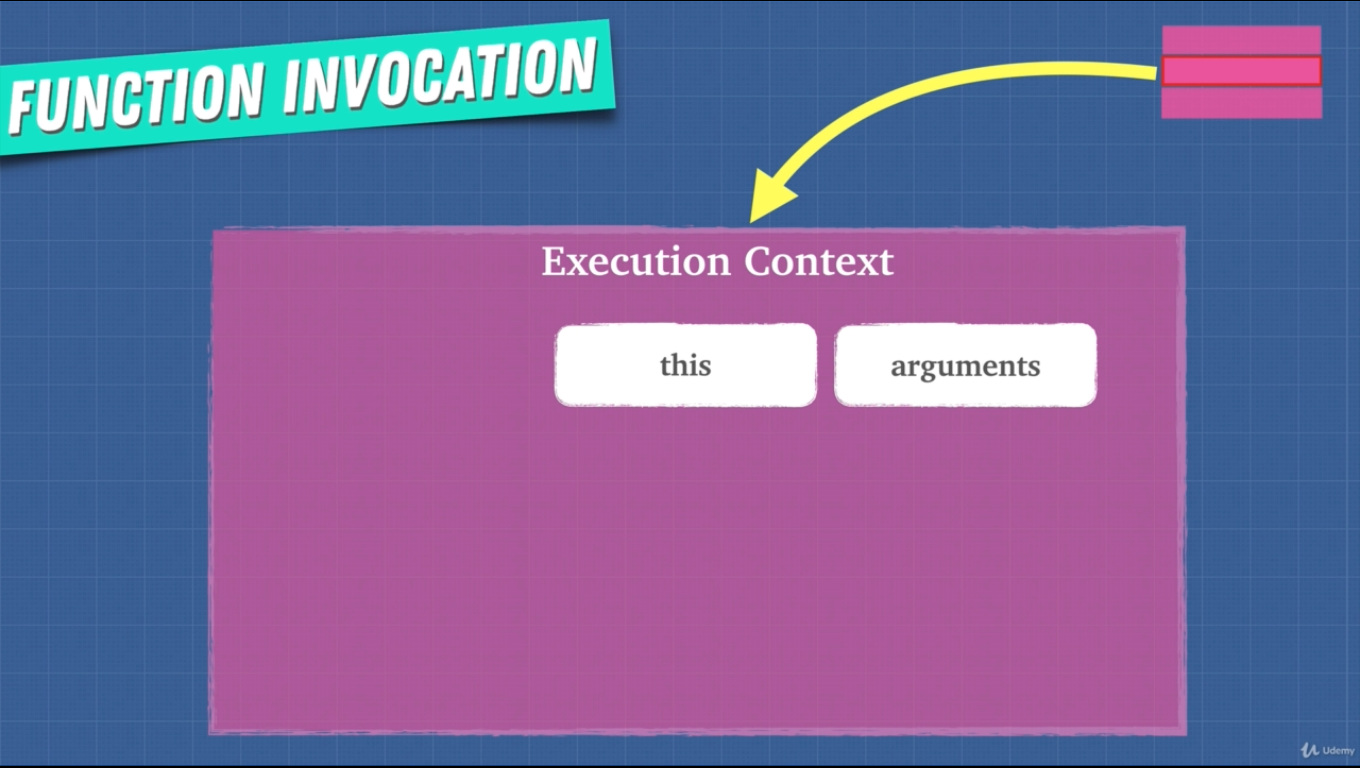
* Function expression is defined at runtime.
* Function declaration is defined at parse time (reading the code and allocating mamory).

function() {

Here is a new execution context.

We get this and arguments objects.

}



**9. arguments Keyword**

* Using arguments is bad practice. JS engine or compiler may not be able to optimize our code.
* arguments is not really an array, it’s an object.
  + Work around to this is

