

this : general rules

① refer as the object when you call like `obj.func()` ;

② You can change this to be other thing using `call`, `apply`, `bind`

① `func.bind(null)` \leftarrow window
`func.bind(undefined)`

② `func.bind(obj, arg1, arg2);` \leftarrow this in func = obj.
`func.call(obj, arg1, arg2);`
`func.apply(obj, [arg1, arg2, ...]);`

③ `bind`, `apply`, `call`
 - `bind` return another function, `apply`, `call` immediately invoke
 - `bind`, `call`, pass args separated using comma
`apply` pass args as a whole - array

Not
in
Exam

\rightarrow ④ event handler : using normal function
 this refers as the object you attached the event

⑤ arrow function : doesn't have its own
 this keyword
 always refer to its surrounding
 LE (Closest function scope)

"use strict", doesn't change this

⑥ Function declaration - not arrow

this = window

'use strict' \Rightarrow undefined.