In Javascript, every thing is represented as an object. And functions and methods execute inside the context of these objects.

In JS, we have two contexts -:

1. A global context
2. A local context (function)

The this keyword is a reference to the current object (context), and is used to refer the properties and methods of the current context.

From a background like Java, PHP or other standard language, this is the instance of the current object in the class method. this cannot be used outside the method and such a simple approach does not create confusion.

Let us break down how to determine the value of this in different scenarios. Consider this as a rule of thumb for determining the value of this -:

## A. For regular functions

I. When **regular functions are executed in the global scope**, the value of **this is the window object**.

II. When **regular functions are methods of objects**, the value of **this is the object itself**.

III. When **regular functions are called inside another function**, the value of **this is the window object**.

IV. When **in strict mod**e, the value of **this in regular function is undefined**.

V. I**nside a constructor function, the value of this is the newly created objec**t.

VI. **When using call, bind or apply with regular function, the value of this is the newly created binding.**

## B. For arrow functions

I. When arrow functions are executed in the global scope, this points to the window object.
**If the arrow function is defined in the topmost scope (outside any function), the context is always the global object (window in a browser)**

II. When **arrow functions are executed as methods, they always point to the window object**. Even if the functions are nested inside the object, it will always point to the window object.

III. When arrow functions are **invoked using call, bind and apply, the context does not change and the value of this still points to the old this value**.

Don’t use arrow functions to call constructor functions, as the constructor will never bind to the object created from the class.

For a more detailed explaination of the this keyword, read through the below article (Reading time ~ 20 min)

https://dmitripavlutin.com/gentle-explanation-of-this-in-javascript/#41-this-in-a-constructor-invocation

This topic is extremely important for JS interviews.