* Use of 'this' Keyword *

→ Overview

*1. In an object, this refers to the object.

~~* In o~~

*2. Alone, this refers to the global object.

*3. In a function, this refers to the global object.

*4. In a function, in strict mode, this is undefined.

*5. In an event, this refers to the element that received the event.

*6. Methods like call(), apply(), and bind() can refer this to any object.

*3.

```
> function callme() {
        console.log(this);
  }
  callme();
```

output

```
▷ Window {.....}
<· undefined
```

*2

```
> this        => output ⇒ ▷ Window {.....}
```

*4. "use strict"; Defines that Javascript code should be executed in "strict mode".

The purpose of "use strict" is to indicate that the code should be executed in "strict mode".

It is not a statement, but a literal expression.

With strict mode, you can not, for eg:, use undeclared variables

We can't call function ~~f~~ before defining it in strict mode. This is because of "Hoisting"

eg:-
```
<script>
  "use script";
  x = 3.14;   // this will cause an error (x is not defined)
</script>
```

*1.
```
const obj = {
   name: "Aman",
   greet : function () {
      console.log(this);
   },
};
```

```
obj.greet();
→ { name: 'Aman', greet: f}
```

So, this refers to object obj.

Note :- If greet method is defined using the "Method Shorthand" syntax. Then again, this refers to object itself

Note :- If we define function using Fat Arrow function then this refers to Window object