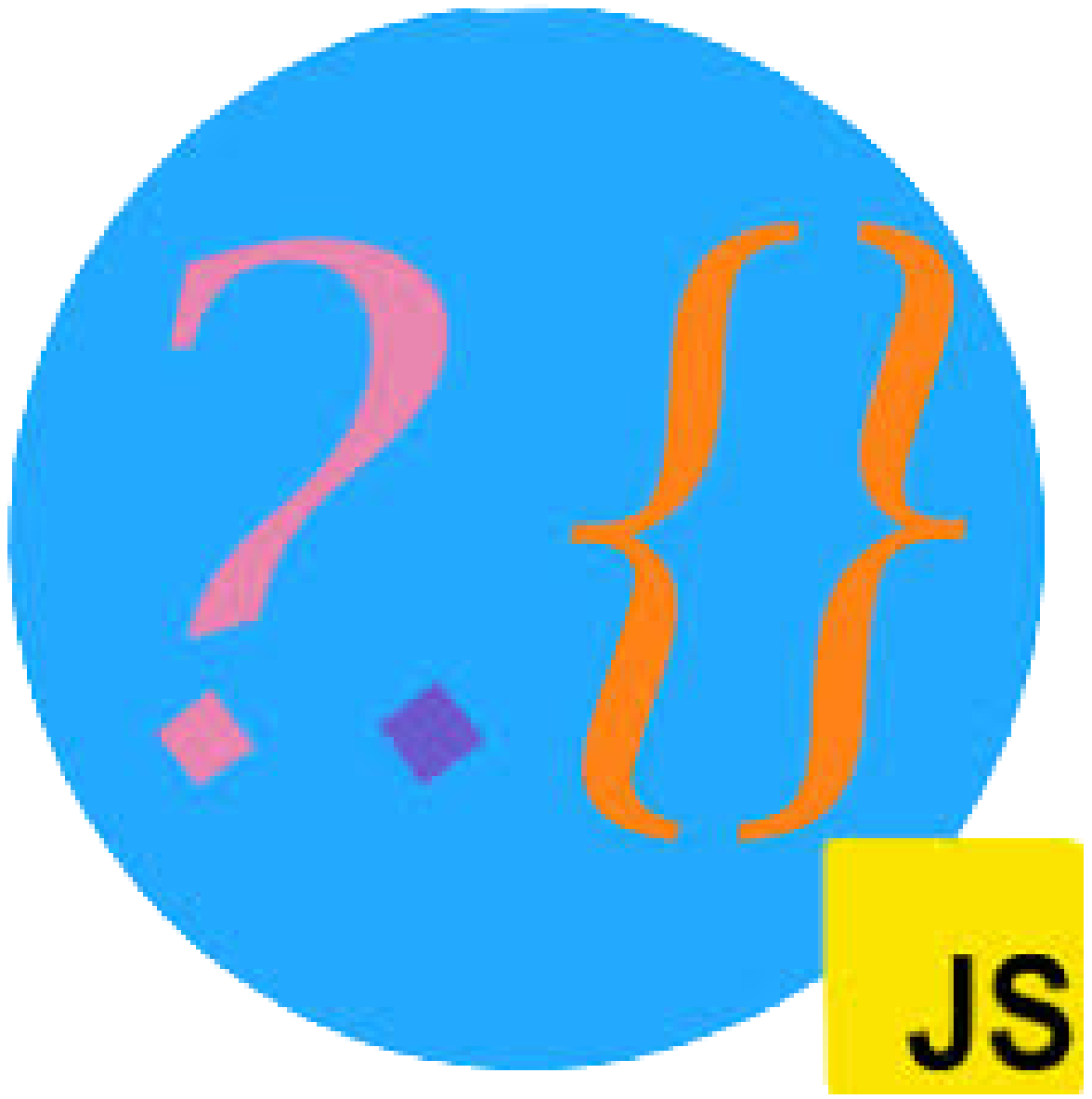
SCOPES IN JAVASCRIPT

The scope is a policy that manages the accessibility of variables, functions, or objects during runtime.



global scope

```
// global scope is the outermost scope
const str = "I'm in a global scope"

//window and document are global objects
//supplied by the browser
window.innerHeight
document.characterSet

const obj = {}
function addObjectProperty(key, value){
  //variable obj is from global scope
and/here it is accessed and modified
fromthe local/function scope
  obj[key] = value
}
addObjectProperty('name', 'abc')
obj   //Object {name: 'abc'}
```

local scope

```
function getAbcFirstName(){
  //variables defined inside a function
  //are in the local/function scope
  const firstname = 'Abc'
  return firstname
}
function getDefFirstName(){
  //new function = new local scope - that's the rule
  const firstname = 'Def'
  return firstname
}
getAbcFirstName()  //"Abc"
getDefFirstName()  //"Def"

// outer scope can't access variables from it's
// children scopes, in this case we can't access
// local variables from global scope
firstname   // ReferenceError: firstname is not defined
```

block scope

```
// block scope - created with for, while, if
// and etc. statements
// using let and const keywords we can declare
// local scope for variables inside block
// statements

if(true){
   const s1 = 'available in block scope only'
   // a code block does not create a scope for var
   // variables (but function body does!)
   var s2 = "available in and outside block"
}
s1  // ReferenceError: s1 is not defined
s2  // "available in and outside block"

for (let i=0; i<5; i++){
  // do something
}
i   // ReferenceError: i is not defined
```

lexical scope

```
// when a function within another function, the
// inner function has access to the scope in the
// outer function, this is called lexical scope

function weather(temp){
  const city = 'Indore'
  function getWeather(){
    function toCelsius(){
      // javascript starts at the innermost
      // scope and searches outwards until it
      // finds the variable/object/function
      // it was looking for
      return (5/9)*(temp-32);
    }
    return `today in ${city} is
${toCelsius()}C`
  return getWeather();
}
weather(68) // "today in Indore is 20C"
```