

Les scopes

L'objet des variables

Le context passe par 2 phases :

- **Phase de création (hoisting).**

C'est à se moment là que l'objet des variables se constitue avec:

Les paramètres des fonctions (Hoisting).

Les déclarations de fonctions (Hoisting).

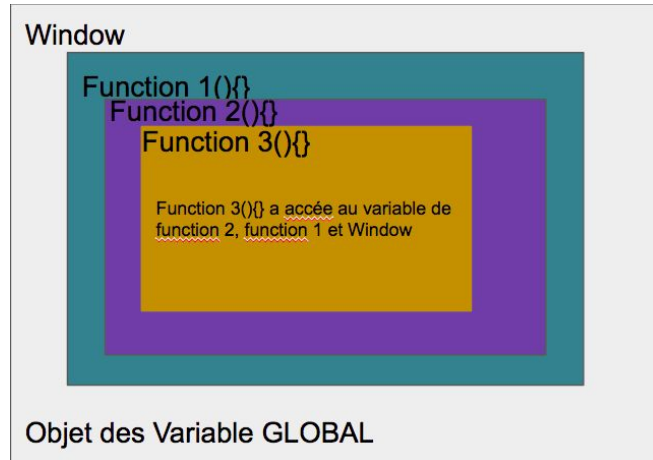
Les déclarations de variables (Hoisting)

- **Phase d'exécution.**

La chaîne des scopes.

Règle:

- Une fonction **déclarer** dans une autre fonction a accès au scope de ces parent, d'où le nom **Chaine** des Scopes.
- Si deux variable on le même nom dans deux fonction différentes, et dans le cas ou la variable est dans une des fonctions parent ou dans l'Objet des Variables Global, la variable la plus proche du scope de la fonction appelé sera utilisé, il ya donc un ordre de priorité.



Let et const Scope de Bloc

let et const respect les scopes de bloc contrairement à var qui ne respecte que les scopes de fonctions, c'est a dire que le principe d'héritage est le même un bloc enfants hérite des variables et des fonctions déclarées dans le bloc parents, mais surtout a retenir que le bloc parents n'a jamais accès à l'objet des variables enfants.

```
let i = 62;  
  
for(let i = 0; i < 10; i++){  
  console.log(i);  
}  
console.log(i)
```

Résultat →

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
62  
>
```

Dans se cas la le premier let i = 62 est declarer dans le bloc window et le deuxieme let i = 0 est declarer dans le bloc du for.