|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Var** | **Let** | **Const** |
| * Declare and Initialize. | can declare no need of initialize at declaration  Var A;  A=10; possible | can declare no need of initialize at declaration  Let A;  A=10; possible | can declare but needed to initialize  at declaration  Const A;  A=10; error need initialized A first |
| * Re-declaration | Var A=20;  Var A=10;  possible | Let A=10;  Let A=20;  Not possible | Const A=10;  Const A=20;  Not possible |
| * Re-initialization | Var A=20;  A=10;  Possible | Let A=10;  A=20;  Possible | Const A=10;  A=20;  Not possible |
| * Block Scopes | Var declared inside can be accessed outside the scope | Let declared cannt be accessed outside the scope | Const declared can be accessed outside the scope |
| * Function scopes | Var declared inside cannt be accessed outside the scope | Let declared cannt be accessed outside the scope | Const declared can be accessed outside the scope |
| * Scope type | Var is function scope | Let is block scope | Const is block scope |
| * Memory allocation | Get allocation of memory in heap | Get allocation of memory in call stack/ GEC(window obj) itself | Get allocation of memory in call stack/ GEC(window obj) itself |
| * Accesing before initialization | Variable declare using var will not be temporal dead zone cause var stored into window obj | Variable declare using let will be in temporal dead zone. So we cannot access them before initialization.Cause let stored into call stack | Variable declare using const will be in temporal dead zone. So we cannot access them before initialization cause const stored into call stack |