**Draw Global Execution Context for Below Codes and Explain Line by line.**

console.log(a);  
var a = 10;  
console.log(a);  
a = 20;  
console.log(a);

|  |  |  |
| --- | --- | --- |
| **MEMORY PHASE** | **CODE PHASE** | **OUTPUT** |
| Var a: undefined  10  20 | console.log(a); var a = 10; console.log(a); a = 20; console.log(a); | Undefined  10  20 |

* At the time of memory phase variable var is assigned with undefined.
* In memory phase variables, functions, classes are stored.
* In code phase our code is stored.
* Initially variable ‘a’ is assigned with undefined.
* The value 10 is initialized to ‘a’, now var ‘a’ is modified to 10 and stored in memory phase of GEC, but it is not printed in the output. i.e. var a = 10;
* On applying console.log(a); it will print the output value as 10.
* Then a is assigned with 20 and it is stored in memory phase.
* On applying console.log(a); it will print the output value as 20.

================================================================

console.log(b);  
var b = 5;  
var b = 15;  
console.log(b);  
b = b + 5;  
console.log(b);

|  |  |  |
| --- | --- | --- |
| **Memory phase** | **code phase** | **output** |
| Var b: undefined  5  15  20 | console.log(b); var b = 5; var b = 15; console.log(b); b = b + 5; console.log(b); | Undefined  15  20 |

* At the time of memory phase variable var is assigned with undefined.
* In memory phase variables, functions, classes are stored.
* In code phase our code is stored.
* Initially variable ‘b’ is assigned with undefined.
* On applying console.log(b); the output will be printed as undefined.
* The value 5 is initialized to ‘b’, now var ‘b’ is modified to 5 and stored in memory phase of GEC, but it is not printed in the output.
* The value 15 is initialized to ‘b’, now var ‘b’ is modified to 15 and stored in memory phase of GEC, but it is not printed in the output.
* On applying console.log(b); it will print the output value as 15.
* Now b = b + 5; is assigned to b, the value 20 is stored in memory phase. But it is not printed in output.
* On applying console.log(b); it will print the output value as 20.

================================================================

console.log(c);  
c = 30;  
var c;  
console.log(c);  
c = c \* 2;  
console.log(c);

|  |  |  |
| --- | --- | --- |
| **MEMORY PHASE** | **CODE** **PHASE** | **OUTPUT** |
| Var c: undefined  30  60 | console.log(c); c = 30; var c; console.log(c); c = c \* 2; console.log(c); | Undefined  30  60 |

* At the time of memory phase variable var is assigned with undefined.
* On applying console.log(c); the output is printed as undefined.
* The value 30 is assigned to variable c and the value 30 is stored in memory phase.
* Var c; is declared, but it will be ignored.
* On applying console.log(c); the output will be printed as 30.
* Now the value 60 (c=30, 30\*2=60 i.e.(c\*2)) is assigned to c and this value is stored in memory phase.
* Output is printed as 60 on applying console.log (c);

=================================================================

var d;  
console.log(d);  
d = 50;  
console.log(d);  
d = d + 10;  
console.log(d);  
var d = 100;  
console.log(d);

|  |  |  |
| --- | --- | --- |
| **MEMORY PHASE** | **CODE PHASE** | **OUTPUT** |
| Var d: undefined  50  60  100 | var d; console.log(d); d = 50; console.log(d); d = d + 10; console.log(d); var d = 100; console.log(d); | Undefined  50  60  100 |

* At the time of memory phase variable is assigned with undefined.
* Variable d is declared as a variable.
* By applying console.log(d); it will print output as undefined.
* The value 50 is assigned to d and stored in memory phase.
* On applying console.log(d); the value 50 is printed as output.
* The variable d is re-initialized with 60.
* The value 60 is printed as output.
* Variable d is again re-initialized with 100.
* The value 100 is printed as output.

=================================================================

var e = 1;  
console.log(e);  
e = e + 1;  
console.log(e);  
var e = 10;  
console.log(e);  
e = e \* 2;  
console.log(e);

|  |  |  |
| --- | --- | --- |
| **MEMORY PHASE** | **CODE PHASE** | **OUTPUT** |
| Var e: undefined  1  2  10  20 | var e = 1; console.log(e); e = e + 1; console.log(e); var e = 10; console.log(e); e = e \* 2; console.log(e); | 1  2  10  20 |

* At the time of memory phase variable is assigned with undefined.
* Variable e is declared as a variable.
* Variable e is initialized with 1 and stored in memory phase
* The value is printed as 1 output.
* Variable e is re-initialized with value 2.
* On applying console.log(e) the value 2 is printed as output.
* Variable e is re-initialized with 10, memory phase stores this value.
* On applying console.log(e) 10 will be printed in its output.
* Variable e is again re-initialized to 20 (e\*2=10\*2).
* On applying console.log(e) the value 20 is printed in its output.