JAVASCRIPT - Core Concept

Variables:

Var:

Scope: Global level, Block level scope

Hoisted: variable hoisted to the top and assigned the value to undefined

Declaration and Initialize: Can be re-declared as well as re-initialize.

Example:

console.log(a);

var a = 43;

console.log(a);

var a = 65

console.log(a);

|  |  |
| --- | --- |
| Global execution Context | |
| Memory phase | Code execution phase |
| var a = undefined (**Hoisted & initialized with undefined**) | Log a => undefined  a=43  log a => 43;  a = 65  log a => 65 |

Let:

Scope: Block level scope

Hoisted: variable hoisted to the top, but cannot access the value before initialization as it creates a temporal dead zone, it gives the **# ReferenceError** error**.**

Declaration and Initialization: Can re-initialize but not re-declared, it gives the **#** **SyntaxError** error.

Example:

console.log(a);

let a = 43;

console.log(a);

a = 65

console.log(a);

|  |  |
| --- | --- |
| Global execution Context | |
| Memory phase | Code execution phase |
| let a (**Reserved in memory but NOT initialized → TDZ**) | Log a => referenceError Temporal dead zone  a=43  log a => 43;  a = 65  log a => 65 |

Const:

Scope: Block level scope

Hoisted: Same as Let

Declaration and Initialization: Cannot be re-declared, gives **#TypeError,** nor re-initialize, gives **#SyntaxError,** but it is still mutable(will learn while studying array and objects) **.** Const doesn’t allowed you to just declaration and then initialize later **#SyntaxError**, you have to declare as well as initialize at the same time.

Example:

const a = 43;

console.log(a);

|  |  |
| --- | --- |
| Global execution Context | |
| Memory phase | Code execution phase |
| const a (**Reserved in memory but NOT initialized → TDZ**) | Temporal Dead Zone  a=43  log a => 43; |

Data Types:

Number: integer, floats, Ex.: let a = 45.33

Methods:

1. a.toString() => “45.33”
2. Number.parseInt(a) => 45