



INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY

H Y D E R A B A D

JavaScript

Session 2

Switch

- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.
- SWITCH, CASE, BREAK, DEFAULT are the constructs

```
switch(expression) {  
    case x:  
        // code block  
        break;  
    case y:  
        // code block  
        break;  
    default:  
        // code block  
}
```

Functions

Functions are the main “building blocks” of the program. They allow the code to be called many times without repetition.

The name of the function

Parameters (empty here)

```
function showMessage() {  
  alert( 'Hello everyone!' );  
}
```

The body of the function
(the code)

this function

- The object that “this” refers changes every time execution context is changed.
- By default the execution context for an execution is global which means that if a code is being executed as part of a simple function call then “this” refers to global object.

Loops

- *Loops* are a way to repeat the same code multiple times.
- Brackets are not required for a single-line body
- We covered 3 types of loops:
 - ***while*** – The condition is checked before each iteration.
 - ***do.....while*** – The condition is checked after each iteration.
 - ***for (;;)*** – The condition is checked before each iteration, additional settings available.

Date and Time

```
new Date() - Thu Jan 26 2017 05:30:00 GMT+0530 (India Standard Time) //  
alert(date)
```

Get Methods

```
getFullYear(), getMonth(), getDate(), getHours(), getMinutes(), getSeconds(),  
getMilliseconds()
```

Set Methods

```
setFullYear(year [,month,date]), setMonth(month[,date]), setDate(date),  
setHours(hour [,min,sec,ms]), setSeconds(sec [,ms]), setSeconds(sec [,ms]),  
setMilliseconds(ms), setTime(milliseconds)
```

Type Conversion

DataTypes can be converted from one type to another based on few inbuilt methods. There are also cases when we need to explicitly convert a value to the expected type.

Explicit Conversion: stated clearly and in detail

Implicit Conversion: suggested though not directly expressed

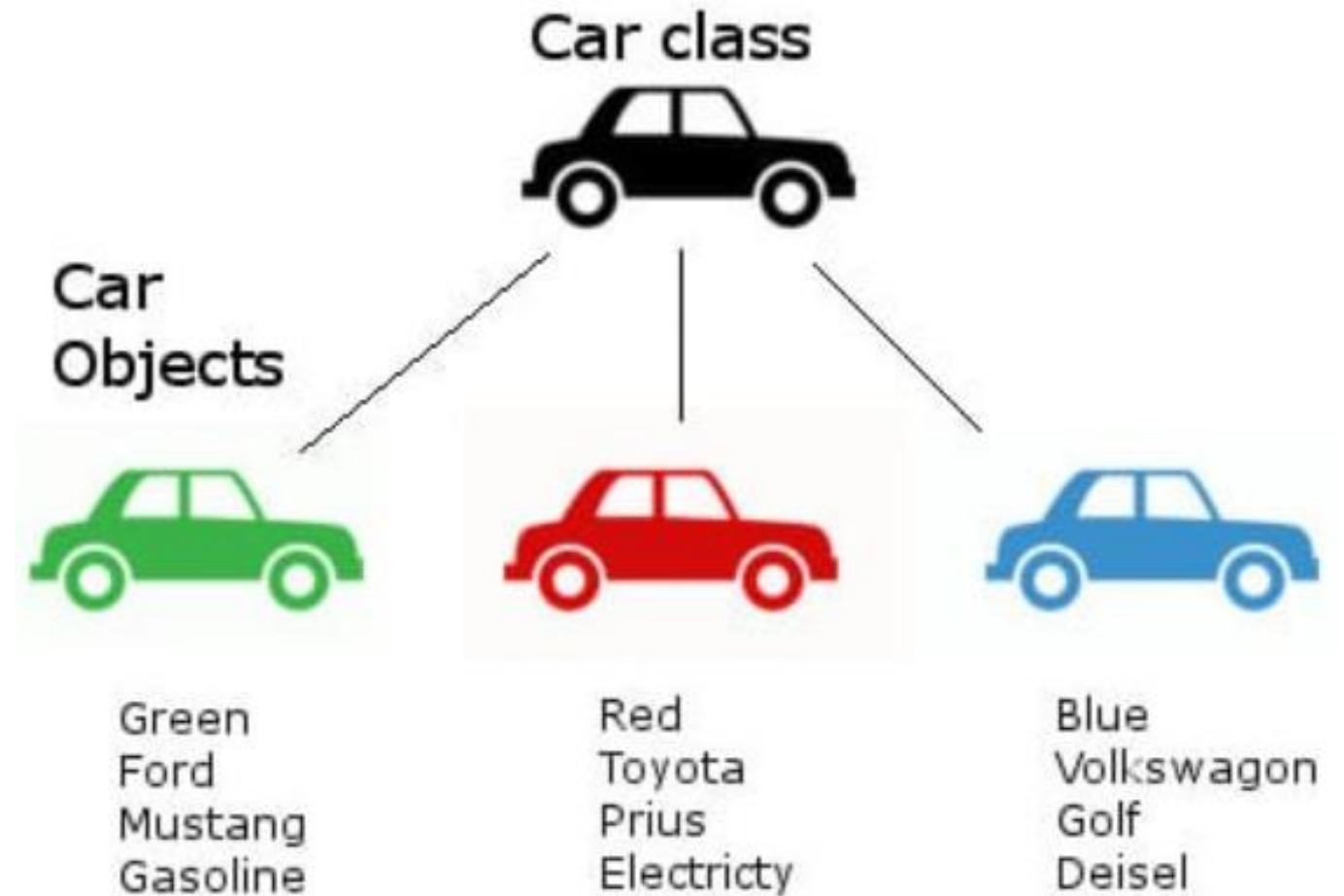
- ToString – Converting any value to String. Explicit conversion is usually required when we read a value from a string-based source like a text form but expect a number to be entered.
- ToNumber – Converting any value to Numeric Type. Undefined - NaN
- ToBoolean – Converting the value into 0 or 1

Class, Object

Classes are special functions which can define function expressions and declarations.

Constructor - It's an ordinary function that is used with the new operator to produce a specialized type of object.

GET and SET Methods



Scope and Closure of Object

- Public
- Privileged
- Private
- Static

Iterables

Iterables object is a generalization of arrays. It helps loops, arrays run in iteration based on well-defined syntax.

Symbol.iterator – a special in-built symbol just for iterating the valuables used in Loops and Arrays

Array.from – This is a universal operator that brings array live value and makes it a real array so that it can be called in array methods

Array

- Objects allow you to store keyed collections of values.
- Trailing Commas declaration
- Can be used as Stack/Queue
- For stacks, the latest pushed item is received first, that's also called LIFO (Last-In-First-Out) principle. For queues, we have FIFO (First-In-First-Out).
- Array can be operated as Stack and Queue at the same time
 - Shift/Unshift – Queue Operations
 - **Shift** - Extracts the first element of the array and returns it
 - **UnShift** - Add the element to the beginning of the array
 - Push/Pop – Stack Operations
 - **push** - adds an element to the end.
 - **pop** - takes an element from the end.

Map

It is collection of Keyed data times, like an Object. But the main difference with Map data-structure is that it allows keys of any type

`new Map()` – creates the Map

`map.set(key,value)` – Stores the value by the key

`map.get(key)` – returns the value by the key. Undefined if key doesn't exist in the map

`map.has(key)` – returns true if Key exists, else False

`map.delete(key)` – removes the value by the key

`map.clear()` – clears the entire Map

`map.size` – returns current element count

Set

A Set is a collection of values, where each value may occur only once.

`new Set(iterable)` – creates the set, from an array of values

`set.add(value)` – adds a value, returns the set itself

`set.delete(value)` – removes the value, returns true if value exists during the call, else false

`set.has(value)` – returns true if the value exists in the set, else false

`set.clear()` – removes everything from the set

`set.size` – is the elements count

Error Handling – try, catch, finally

If a script fails to perform an action and if we wish to capture the error, we use Try – Catch and Finally blocks. This logic only works during run-time

try – all our existing logic. This is executed first.

catch(err) – Catch with error argument so as to handle the error. If there are no errors, this logic is ignored and moves to ***catch***

If error occurs, the try execution is stopped and control is pushed to catch(err). ***err*** variable contains error object with details about what went wrong.

finally – after ***try*** with no errors, after ***catch*** if there are errors