TESTYANTRA

SOFTWARE SOLUTIONS (INDIA) PVT. LTD.

# JAVASCRIPT

EXPERIENTIAL learning factory

## Introduction to JavaScript

TESTYANTRA SOFTWARE SOLUTIONS (INDIA) PVT. LTD.

EXPERIENTIAL learning factory

- JavaScript is a lightweight scripting language aka programming language.
- JavaScript is case sensitive.
- Brendan Eich creator of JavaScript language in 1995.
- Use script tag to include js in html.
  ```
  <script src = “index.js”></script>
  ```
- JavaScript v/s EcmaScript.

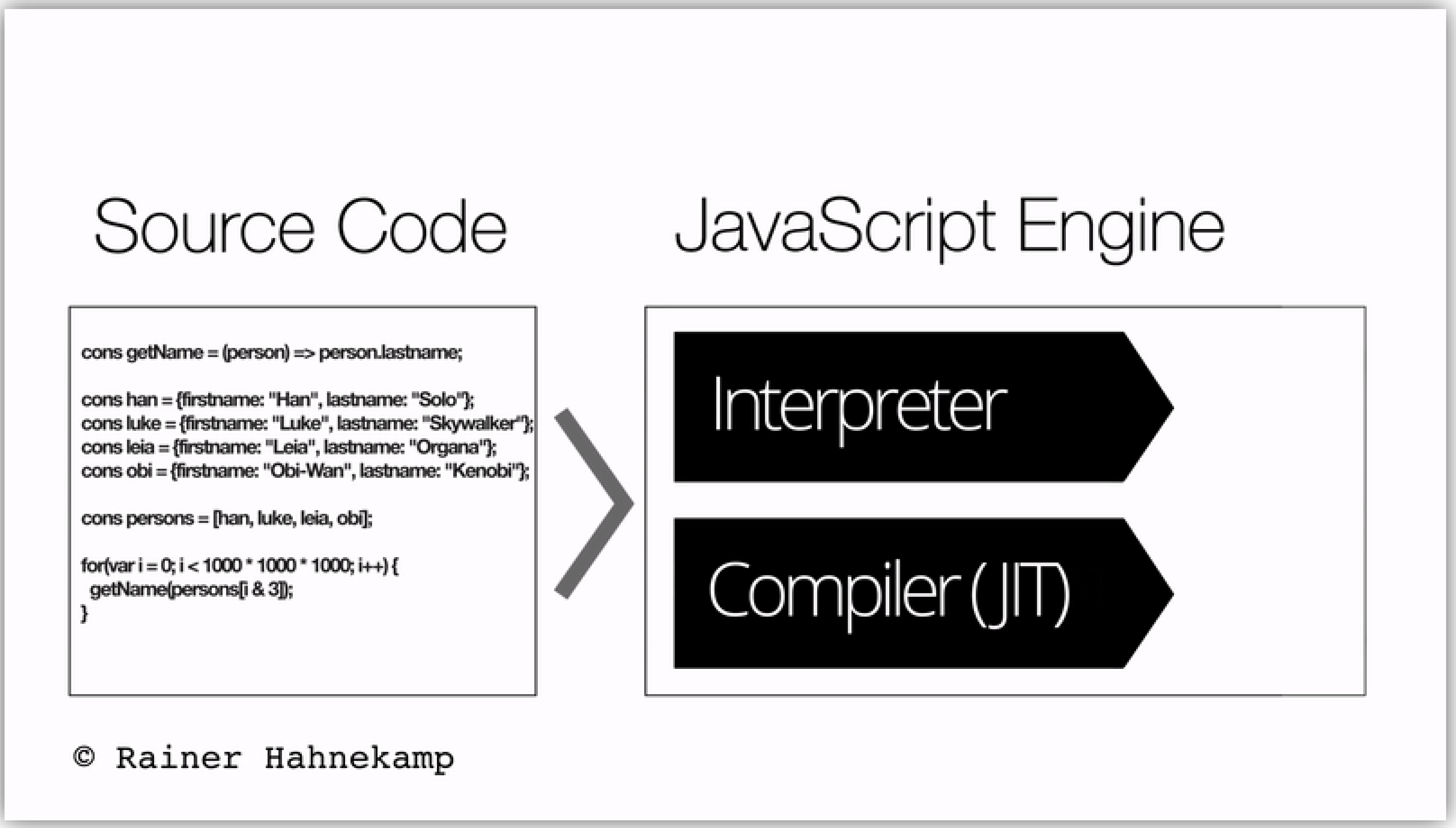
## JavaScript Engine

- A JavaScript Engine is a computer program that executes JavaScript code.

- JavaScript Engines are developed by web browser vendors, and every major browser has one.

  - Firefox – SpiderMonkey.
  - Chrome – V8.
  - Microsoft Edge – Chakra.
  - Safari – JavaScriptCore.

# JavaScript Engine WorkFlow



```
cons getName = (person) => person.lastname;

cons han = {firstname: "Han", lastname: "Solo"};
cons luke = {firstname: "Luke", lastname: "Skywalker"};
cons leia = {firstname: "Leia", lastname: "Organa"};
cons obi = {firstname: "Obi-Wan", lastname: "Kenobi"};

cons persons = [han, luke, leia, obi];

for(var i = 0; i < 1000 * 1000 * 1000; i++) {
  getName(persons[i & 3]);
}
```

# Data Types and Variables

- Variables – Variables are containers that you can store values in it.

  Syntax :- var name(identifier) = ‘Raj’;(value)

- Primitive – Number, String, Boolean, null, undefined. (immutable)

- Reference – Object, Arrays, Date, Math, Function. (mutable)

# Operators

- Arithmetic Operators : +, -, *, /, %, ++, --

- Assignment Operators : =, +=, -=, *=, /=, %=

- Comparison Operators : ==, ===, !=, !==, >, <, >=,<=

- Conditional (Ternary) Operator : *variablename* = (*condition*) ? *value1*:*value2*

  *Ex :* var status = (age >= 18) ? “Adult“ : “Minor";

- Logical Operators : &&, ||, !

- typeof Operator : returns type of a variable, object, function or expression.

  ```
  Ex : typeof "John“     // returns string
       typeof(10)        // returns number
  ```

- Concatenation Operator(+) : used to concatenate strings and variable.

## JavaScript Object

- The values are written as **name : value** pairs (name and value separated by a colon).

  Ex : var person = {

```
firstName : “John”,
lastName : “Doe”,
age : 50,
eyeColor : “blue”,
fullName : function() {
        return this.firstName + " " + this.lastName;
    }
};
```

  Using **new** keyword :

```
var person = new Object();
person.firstName = "John";
person.lastName = "Doe";
person.age = 50;
person.eyeColor = "blue";
```

# JavaScript Arrays

- An array is a special variable, which can hold more than one value at a time.

  Syntax : var *array_name* = [*item1*, *item2*, ...];
  Ex : var fruits = [‘Apple’, ‘Banana’, ‘Orange’];
  Ex : var employee = [‘John’,45,null,,true]; // multiple data types supported
  Accessing values – fruits[0]

- Using **new** keyword

  var fruits = new Array(‘Apple’, ‘Banana’, ‘Orange’);

- Complex Array

```
var library = [
        {author : ’Bill Gates’ , title : ‘The Road Ahead’ , bookID : 1254 },
        {author : ‘Steve Jobs’ , title : ‘Walter Isaacson’ , bookID : 4264 },
        {author : ’Suzanne Collins’ , title : ‘Mockingjay : The Final Book of The
              Hunger Games’ , bookID : 3254 },
    ];
```

# Date Object

TESTYANTRA SOFTWARE SOLUTIONS (INDIA) PVT. LTD. EXPERIENTIAL learning factory

- Creating Date Objects

  - new Date()

    Ex : var date = new Date();

  - new Date(*year, month, day, hours, minutes, seconds, milliseconds*)

    Ex : var date = new Date(2018, 11, 24, 10, 33, 30, 0);

  - new Date(*milliseconds*)

    Ex : var date = new Date(0); Zero time is January 01, 1970 00:00:00 UTC.

  - new Date(*date string*)

    Ex : var date = new Date("October 13, 2014 11:13:00");

# Date Object Methods

| Method | Description |
|---|---|
| getFullYear() | Get the **year** as a four digit number (yyyy) |
| getMonth() | Get the **month** as a number (0-11) |
| getDate() | Get the **day** as a number (1-31) |
| getHours() | Get the **hour** (0-23) |
| getMinutes() | Get the **minute** (0-59) |
| getSeconds() | Get the **second** (0-59) |
| getMilliseconds() | Get the **millisecond** (0-999) |
| getTime() | Get the time (milliseconds since January 1, 1970) |
| getDay() | Get the weekday as a number (0-6) |
| Date.now() | Get the time. ECMAScript 5. |

## Math Object

- Math.PI – returns pi value.
- Math.round(x) - returns the value of x rounded to its nearest integer.
- Math.pow(x , y) - returns the value of x to the power of y.
- Math.sqrt(x) - returns the square root of x.
- Math.abs(x) - returns the absolute (positive) value of x.
- Math.ceil(x) -  returns the value of x rounded **up** to its nearest integer.
- Math.floor(x) - returns the value of x rounded **down** to its nearest integer.
- Math.min() and Math.max() - can be used to find the lowest or highest value in a list of arguments.
- Math.random() - returns a random number between 0 (inclusive), and 1 (exclusive).

## Control Structures, Loops

TESTYANTRA SOFTWARE SOLUTIONS (INDIA) PVT. LTD. EXPERIENTIAL learning factory

- if, if-else, if else-if, for, switch, while, do-while, continue, break, for-of, for-in, forEach

| for loop | forEach | for of | for in |
|---|---|---|---|
| Does not work with object | Does not work with object, only use with arrays | Does not work with object | Works with object and arrays |
| Does not ignore empty elements | Ignores empty elements | Does not ignore empty elements | Ignores empty elements |
| break statement is supported | break statement is not supported coz it’s a method | break statement is supported | break statement is supported |
| Ignores extra properties which does not have index | Ignores extra properties which does not have index | Ignores extra properties which does not have index | Does not ignore extra properties which does not have index |

# JavaScript Functions

TESTYANTRA SOFTWARE SOLUTIONS (INDIA) PVT. LTD.

EXPERIENTIAL learning factory

- Named Functions :

```
function funcname( args ) {
        //statements
    }
```

- Function Expression(Anonymous Function) :

```
var getName = function( args ) {
        //statements
    }
```

- IIFE(Immediately Invoked Function Expression) :

```
(function( args ){
        //statements
    })();
```

- ES6 Arrow Function :

```
( args ) => {
        //statements
    }
```

# Variable Hoisting

TESTYANTRA SOFTWARE SOLUTIONS (INDIA) PVT. LTD. EXPERIENTIAL learning factory

- Variable hoisting :

  Before Hoisting :-

```
console.log(hoist);
var hoist = ‘The variable has been
hoisted’;
```

  After hoisting :-

```
var hoist;
console.log(hoist);
hoist = ‘The variable has been hoisted’;
```

- Variable Hoisting in Function :

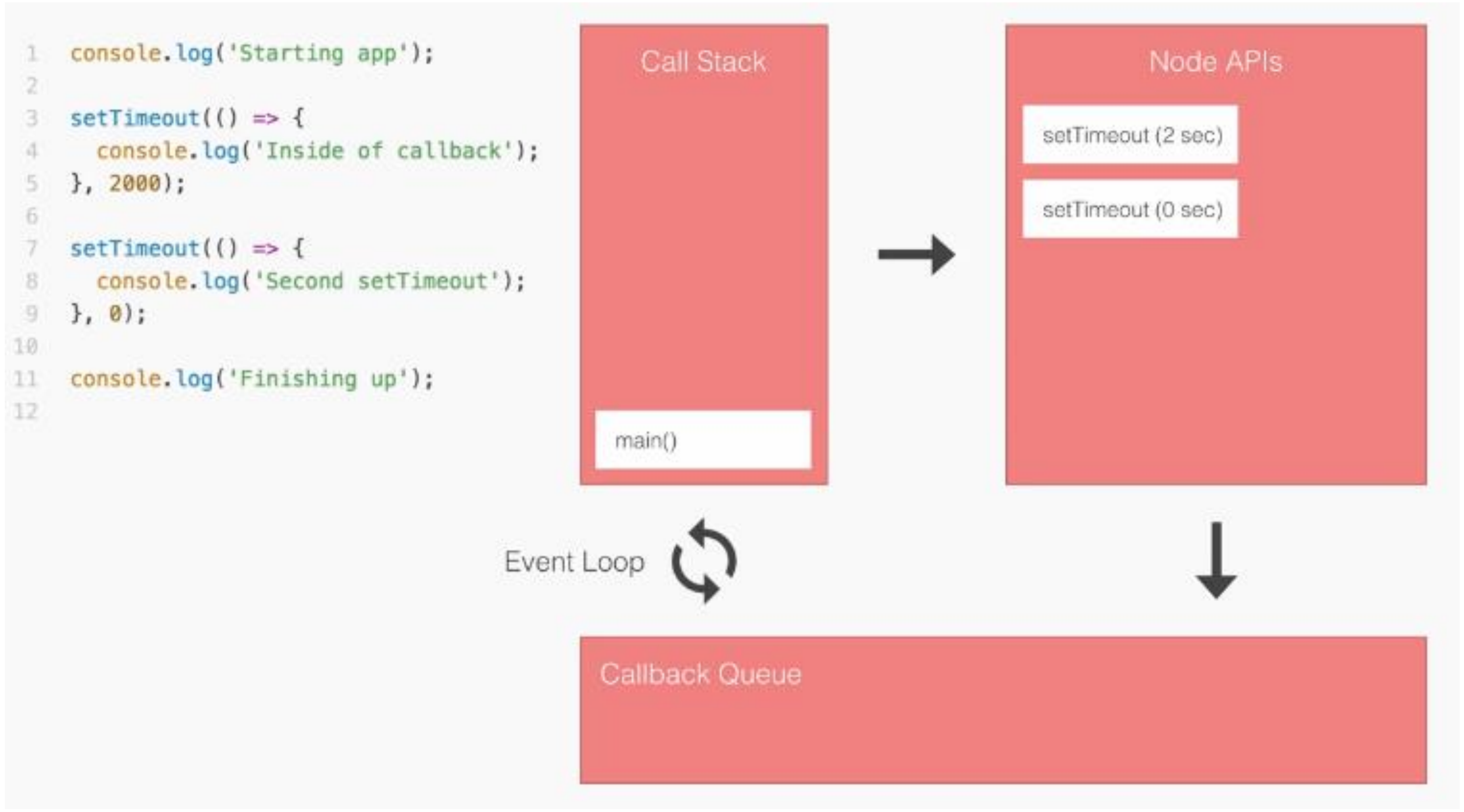
  Before Hoisting :-

```
function hoist() {
        console.log(message);
        var message = ‘Hoisting’
}
hoist();
```

  After hoisting :-

```
function hoist() {
        var message;
        console.log(message);
        message = ‘Hoisting’
}
hoist();
```

# JavaScript Call Stack



```
console.log('Starting app');

setTimeout(() => {
  console.log('Inside of callback');
}, 2000);

setTimeout(() => {
  console.log('Second setTimeout');
}, 0);

console.log('Finishing up');
```

## Array and String Methods

TESTYANTRA
SOFTWARE SOLUTIONS (INDIA) PVT. LTD.
EXPERIENTIAL learning factory

- **Array** :

  Properties -

  length

  Methods -

  forEach(( callback( value, index) )),boolean isArray( array ),boolean includes( searchElement, fromIndex ), number push(items),string pop(), string shift(), number unshift(items), array splice(start index, delete count, items), array slice( start index, end index), string join( separator ), number indexOf( searchElement, fromIndex ),array map(( callback( value, index) )),array filter(( callback( value, index) ))

- **String** :

  Properties -

  length

  Methods -

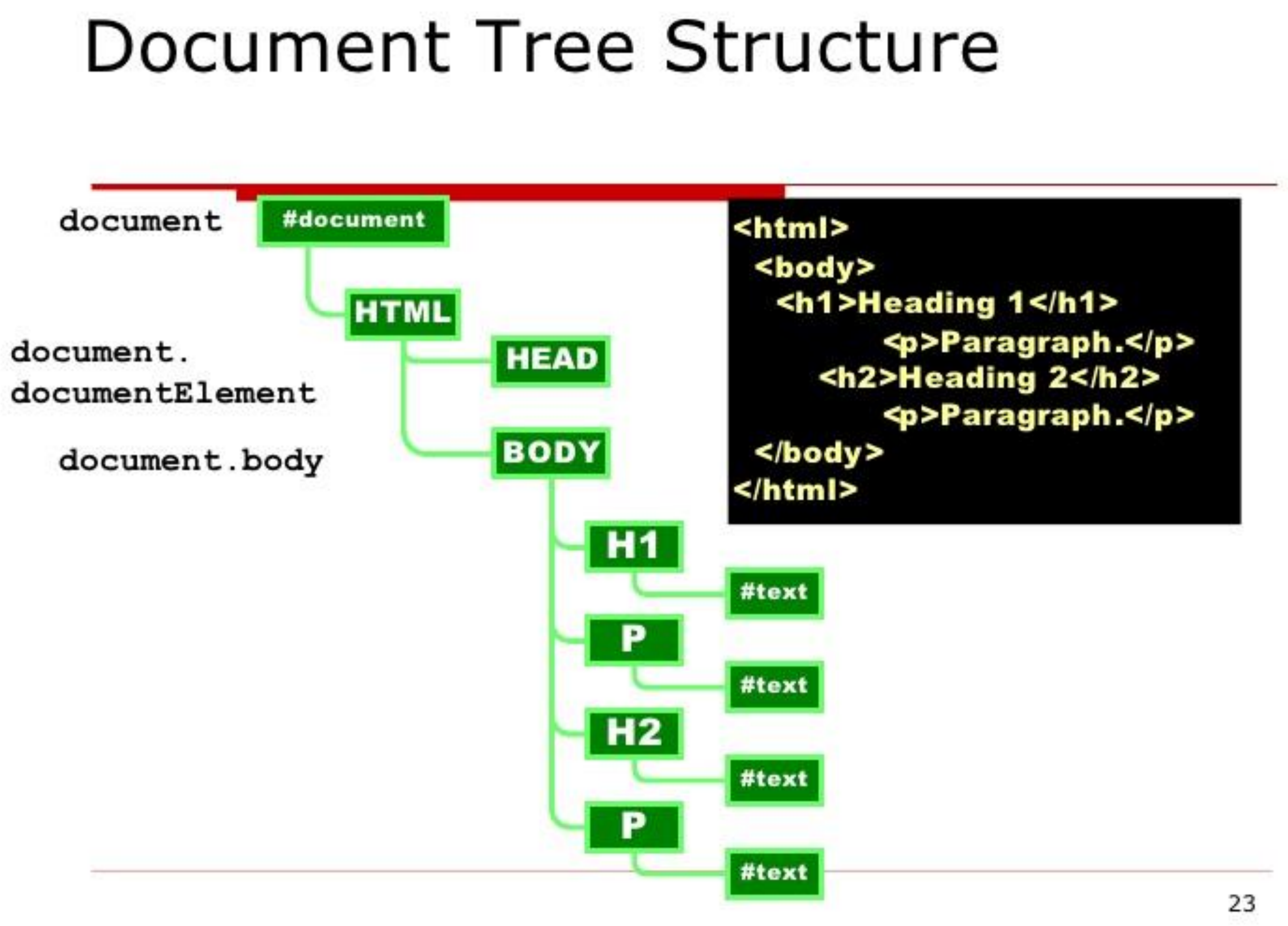
  toLowerCase(), toUpperCase(), charAt( position ), indexOf( searchString, position ), concat(…strings), includes(search String, start position),replace(search Value, replaceValue), substr( start, length ),trim()

# Browser Object Model(BOM)

- The Browser Object Model (BOM) allows JavaScript to "talk to" the browser.
- **Window Object** :
  The window object is supported by all browsers. It represents the browser's window tab.
- **Properties** :
  innerHeight, innerWidth.
- **Methods** :
  prompt, alert, confirm, open, close, console.
- **Location Object** :
  href, hostname, pathname, protocol, port.
- **History Object** :
  back(), forward().

## DOM Tree Structure

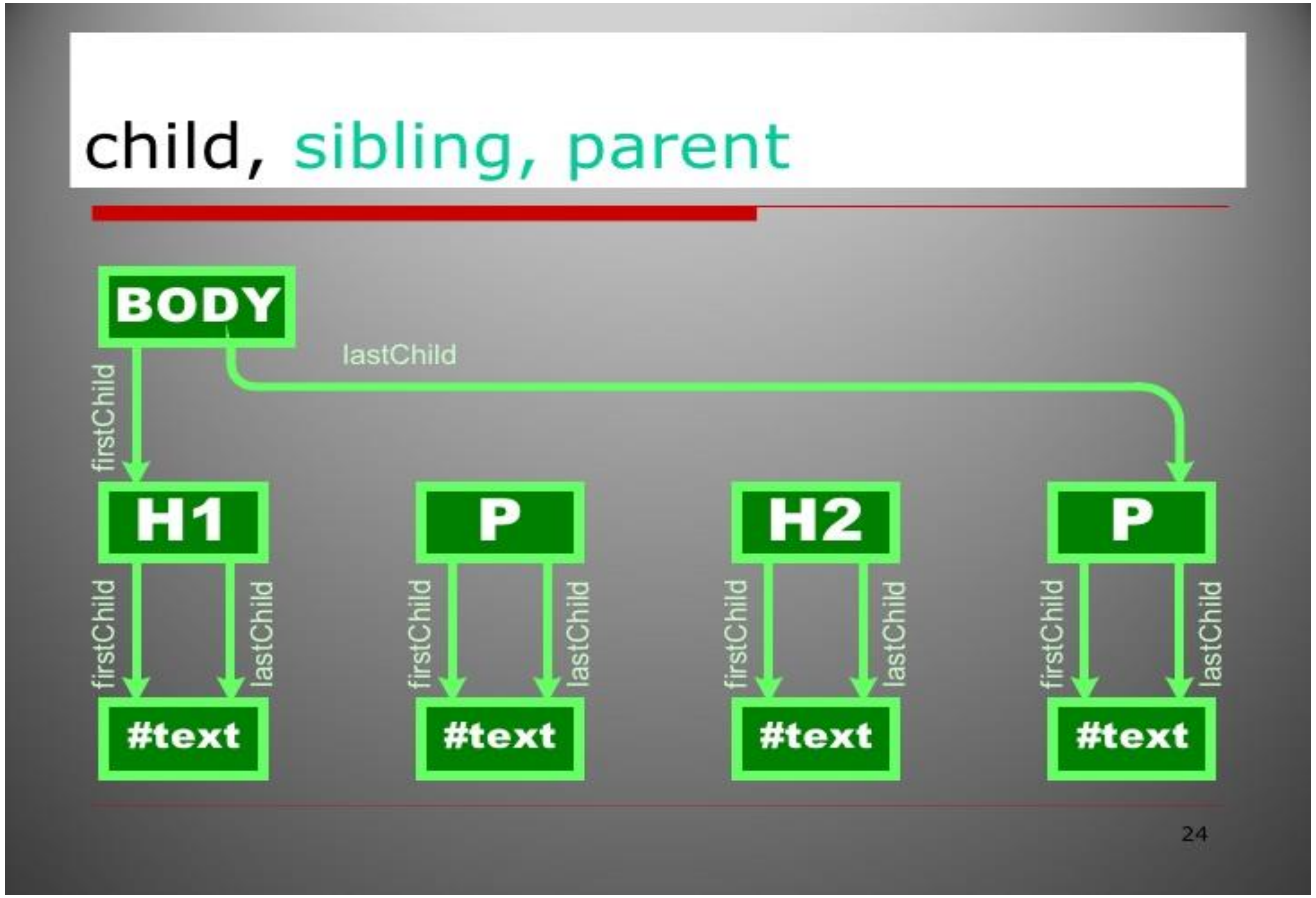
TESTYANTRA SOFTWARE SOLUTIONS (INDIA) PVT. LTD. EXPERIENTIAL learning factory



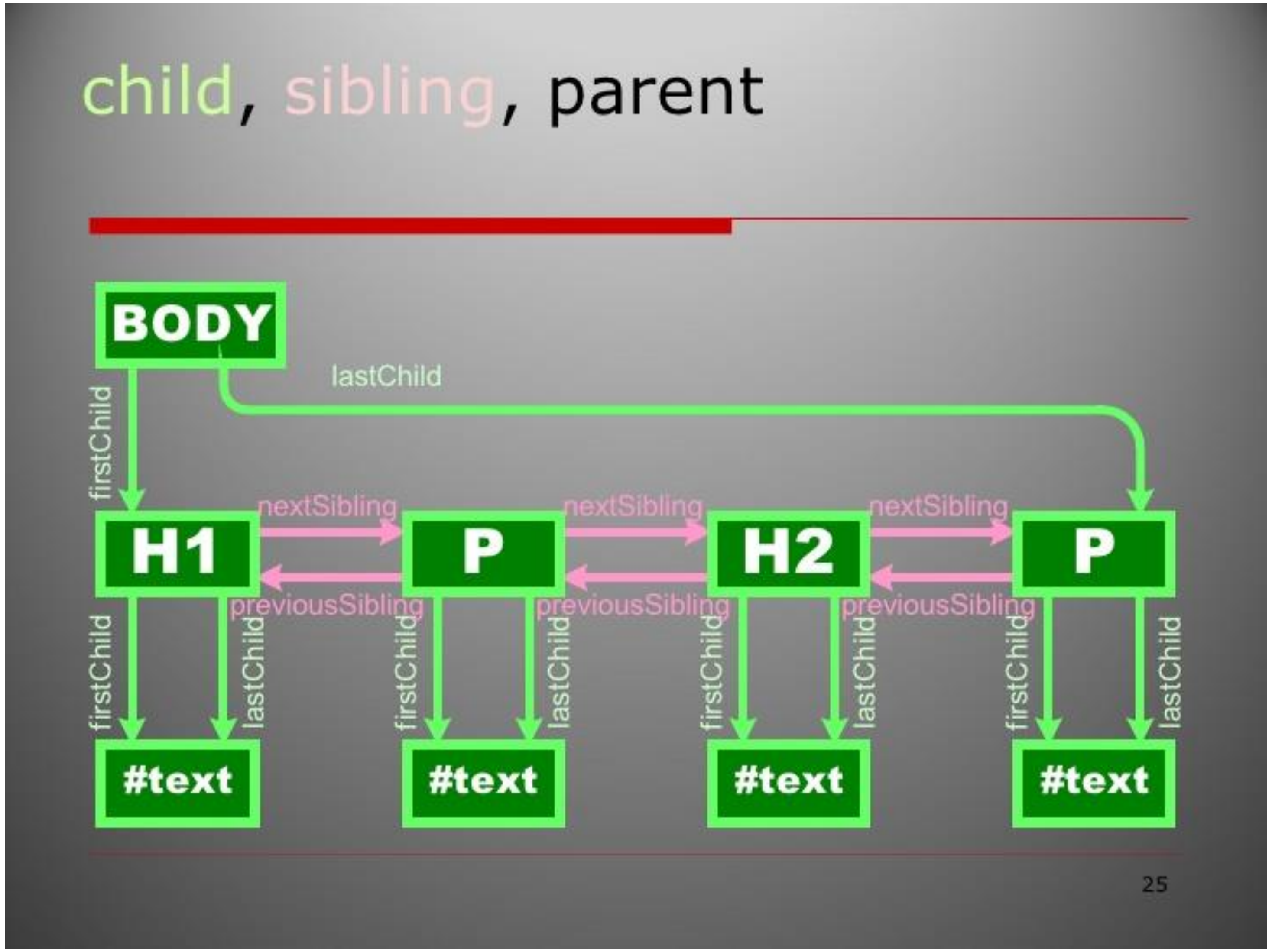
```
<html>
 <body>
  <h1>Heading 1</h1>
        <p>Paragraph.</p>
    <h2>Heading 2</h2>
        <p>Paragraph.</p>
 </body>
</html>
```

# DOM

- The Document Object Model (**DOM**) is a programming interface for HTML and XML documents, which can be modified with a scripting language such as **JavaScript**

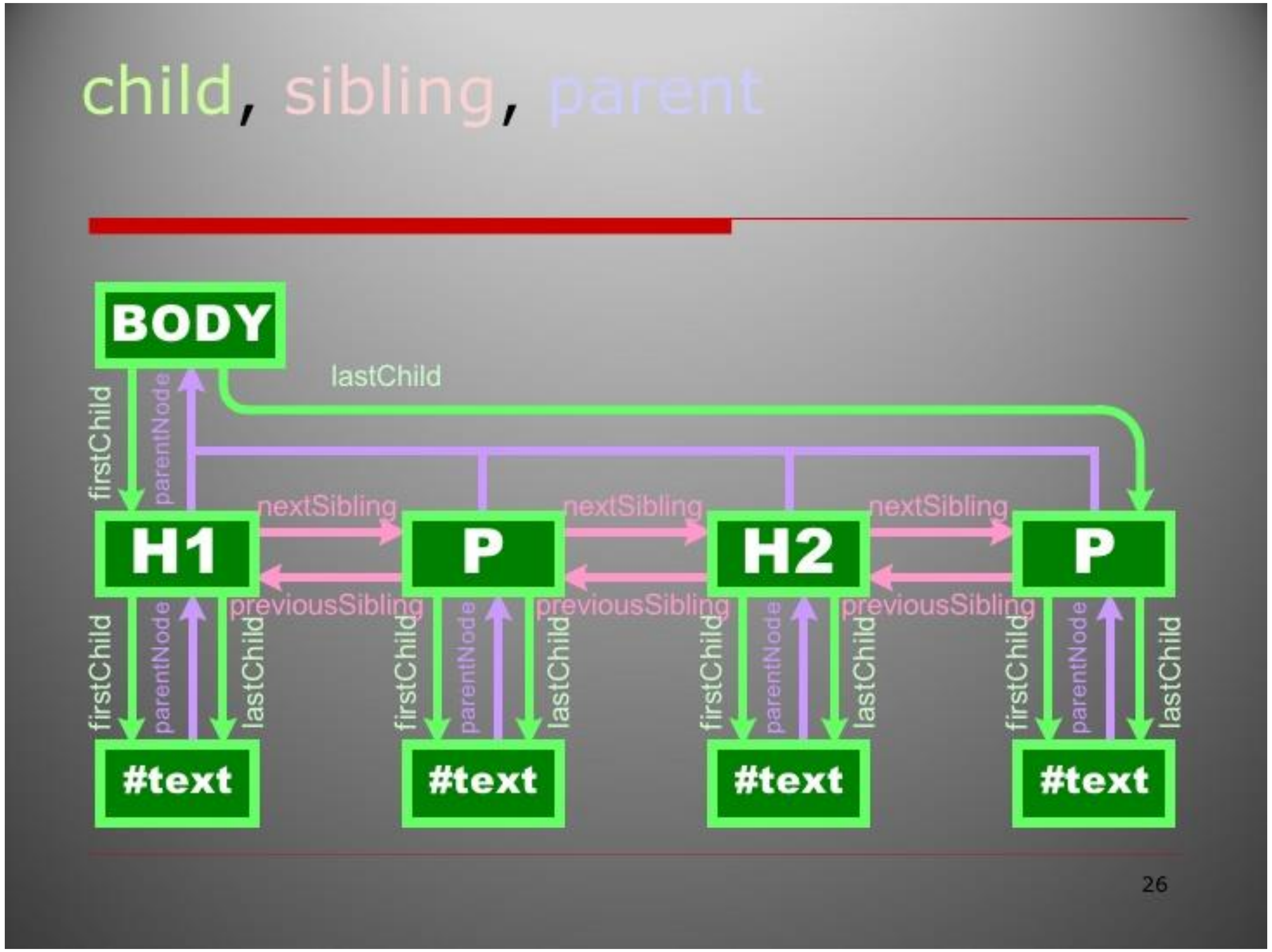


# DOM


child, sibling, parent
BODY
lastChild
firstChild
H1
P
H2
P
nextSibling
nextSibling
nextSibling
previousSibling
previousSibling
previousSibling
firstChild
lastChild
firstChild
lastChild
firstChild
lastChild
firstChild
lastChild
#text
#text
#text
#text
25

## DOM

TESTYANTRA
SOFTWARE SOLUTIONS (INDIA) PVT. LTD.
EXPERIENTIAL learning factory



## Document Object Model(DOM)

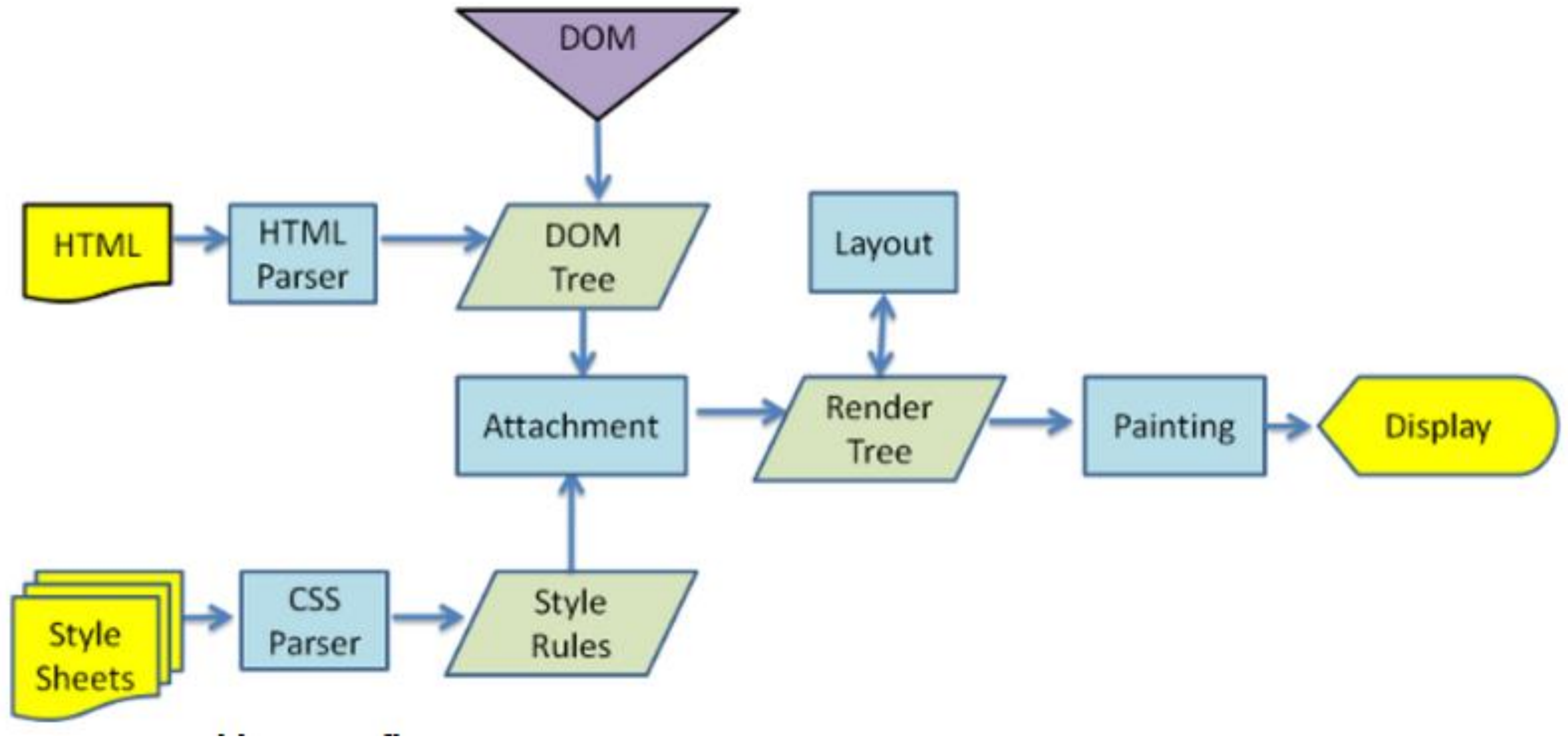
- The Document Object Model (**DOM**) is a programming interface for HTML and XML documents, which can be modified with a scripting language such as **JavaScript.**

- **Methods** :
  write(), createElement( element ), getElementById( id name ), getElementsByTagName( tag name ), getElementsByClassName( class name ), querySelector( tag or id or class ), querySelectorAll( tag or id or class ), appendChild( element ).

- **Properties** :
  firstChild, firstElementChild, lastChild, lastElementChild, removeChild, className, classList, childNodes.

- **DOM Events** :
  onclick, onmouseover, onmouseout, onkeyup, onkeydown.

# DOM Update Process



# Working with Form, JSON

- Data Validation.

- Create Element Dynamically, add Style Dynamically.

- JSON(JavaScript Object Notation) :
  JSON is a format for sharing data.

- JSON.stringify()

- JSON.parse()

- JSON Data Types :
  In JSON, values must be one of the following data types
  a string, a number, an object, an array, a boolean, null
  JSON values **cannot** be one of the following data types:
  a function, a date, undefined

## Closures, Callback, Recursio

- Closures :
  An inner function has always access to the parameters and variables of its outer function, even after the outer function has returned and removed from the stack.

- Callback :
  A callback function is a function passed into another function as an argument.
  A callback function is a function that is to be executed after another function has finished executing – hence the name ‘call back’.

- Recursion :
  A function calling itself repeatedly until it arrives at a result.

# ES6 Features

- let, const keywords.
- Arrow functions.
- Template strings(``).
- Object and Array De-structuring.
- …spread and …rest operators.
- Promises.
- Default parameters.
- ES6 Tooling : Babel.

TESTYANTRA
SOFTWARE SOLUTIONS (INDIA) PVT. LTD.

## Thank You !!!

No.01, 3rd Cross Basappa Layout, Gavipuram Extension, Kempegowda Nagar, Bengaluru, Karnataka 560019

praveen.d@testyantra.com

www.testyantra.com

EXPERIENTIAL learning factory