

TESTYANTRA

SOFTWARE SOLUTIONS (INDIA) PVT. LTD.

JAVASCRIPT

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.

- 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.

Source Code

```
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]);  
}
```

JavaScript Engine

Interpreter

Compiler (JIT)

© Rainer Hahnekamp

- 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)

- 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.

- 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";
```

- 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');`

- Example Array with multiple Objects

```
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 },  
];
```


- 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");

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.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).

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

for loop	forEach method	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

- 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 :
Before Hoisting :-

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

After hoisting :-

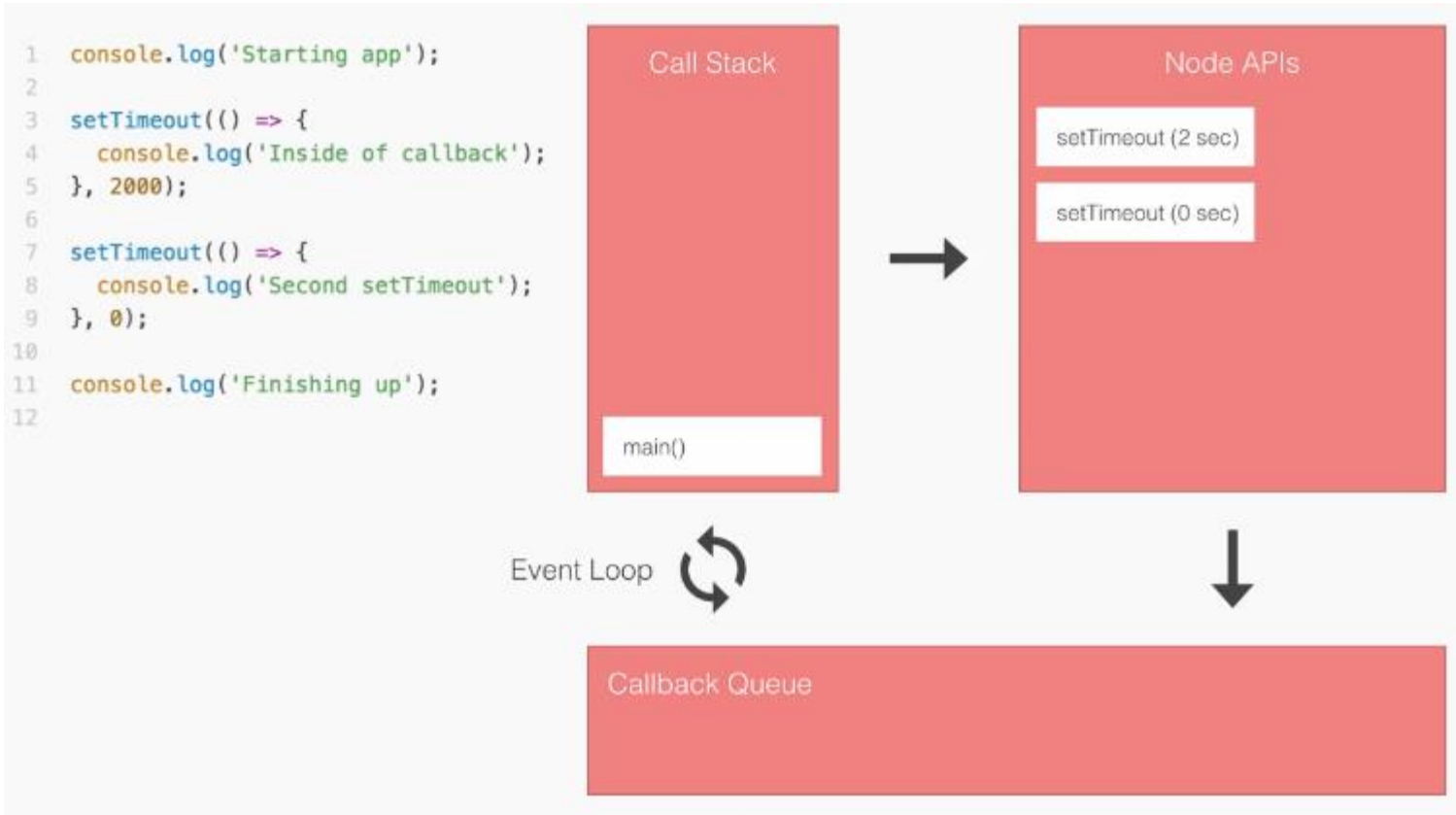
```
var hoist;  
console.log(hoist);  
hoist = 'The variable has been hoisted';
```

- 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();
```



■ 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 -

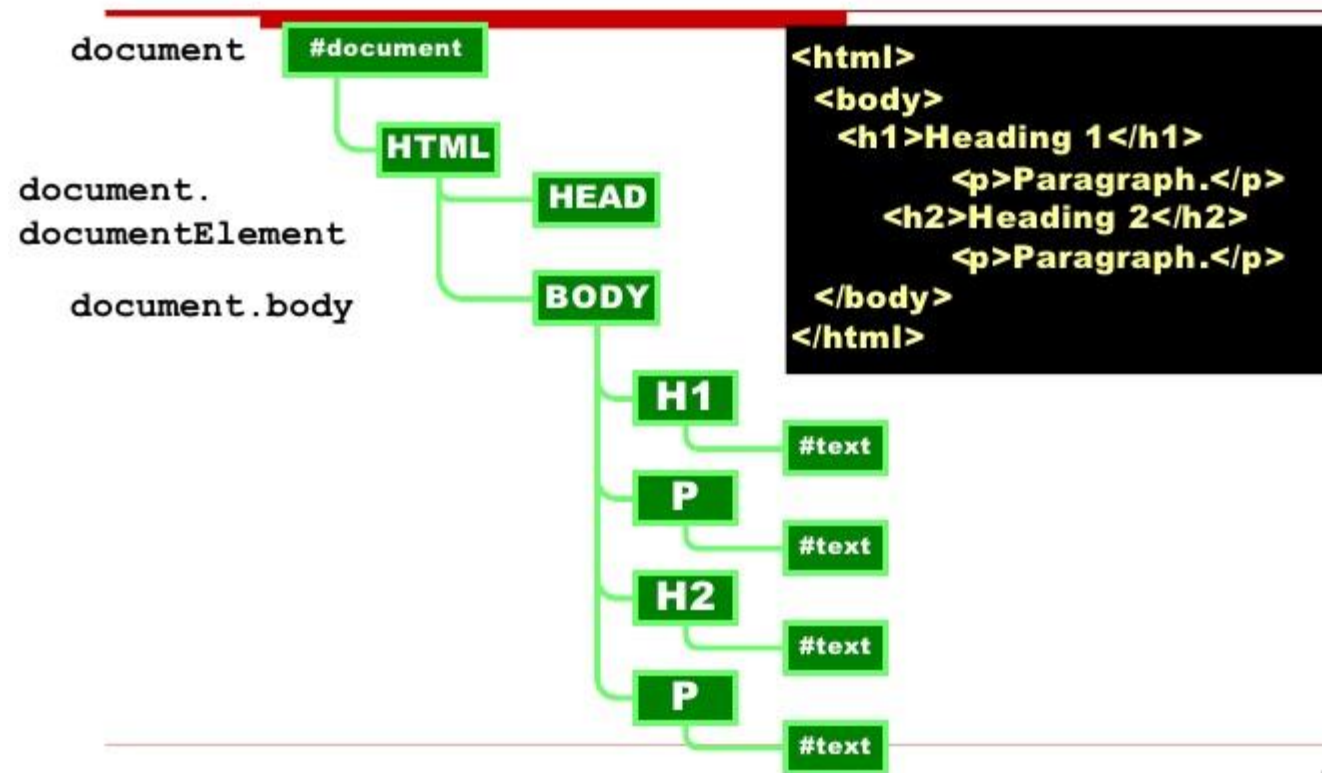
length

Methods -

toLowerCase(), toUpperCase(), charAt(position), indexOf(searchString, position), concat(...strings), includes(search String, start position),replace(search Value, replaceValue), substr(start, length), substring(start, end(not inclusive)) , trim()

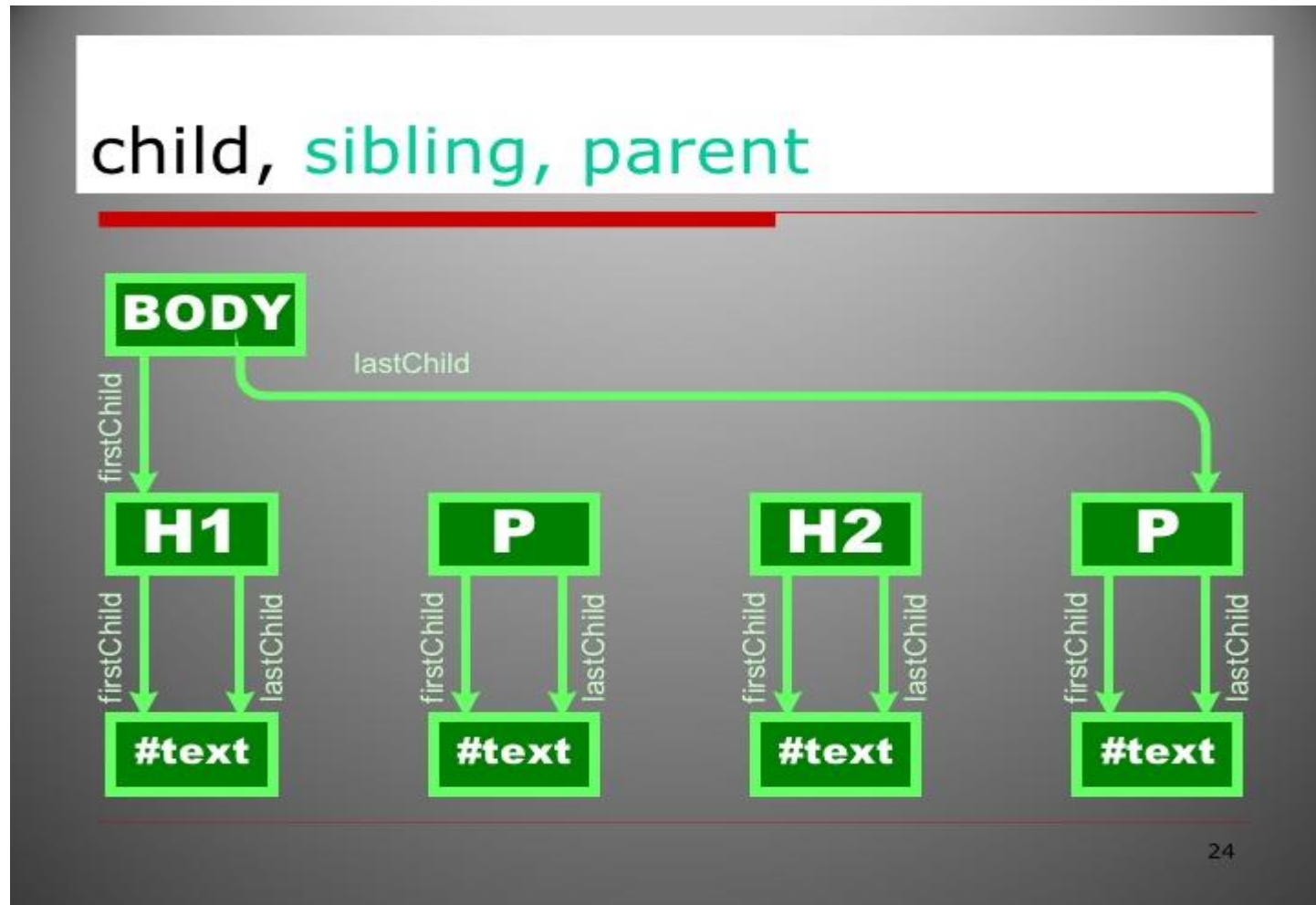
- The Browser Object Model (BOM) allows JavaScript to "talk to" the browser.
- **Window Object :**
The window object is supported by all browsers. It represents the browser's window tab.
- **Properties :**
innerHeight, innerWidth.
- **Methods :**
prompt, alert, confirm, open, close, console.
- **Location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

Document Tree Structure

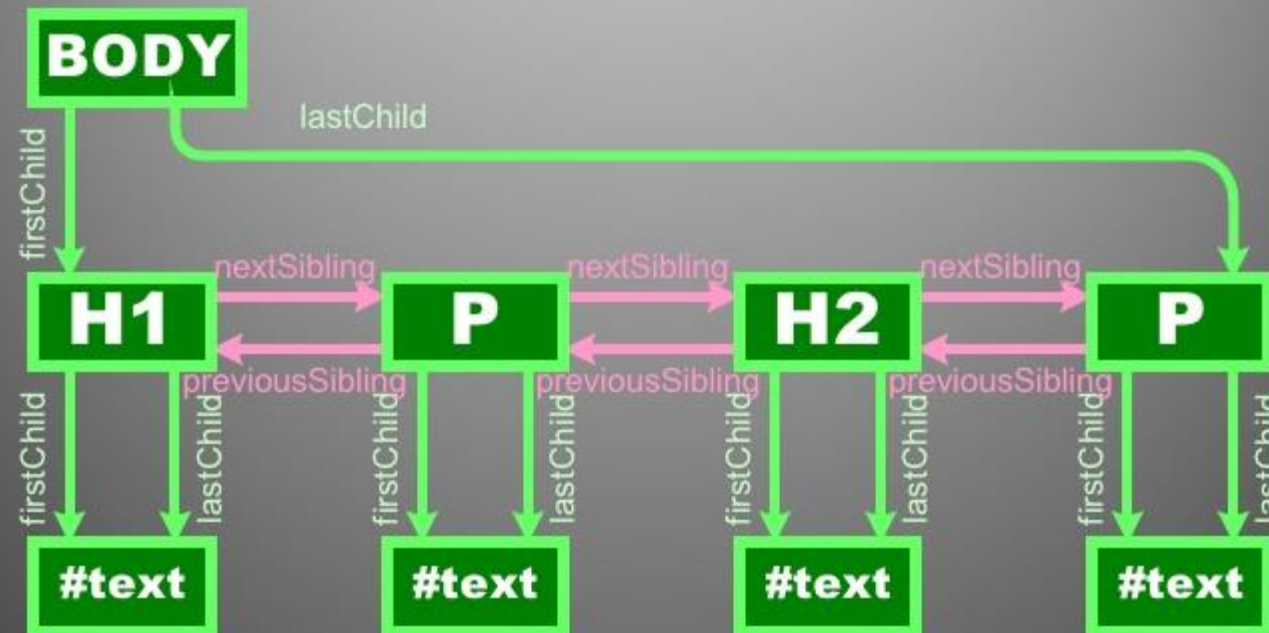


23

- 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**

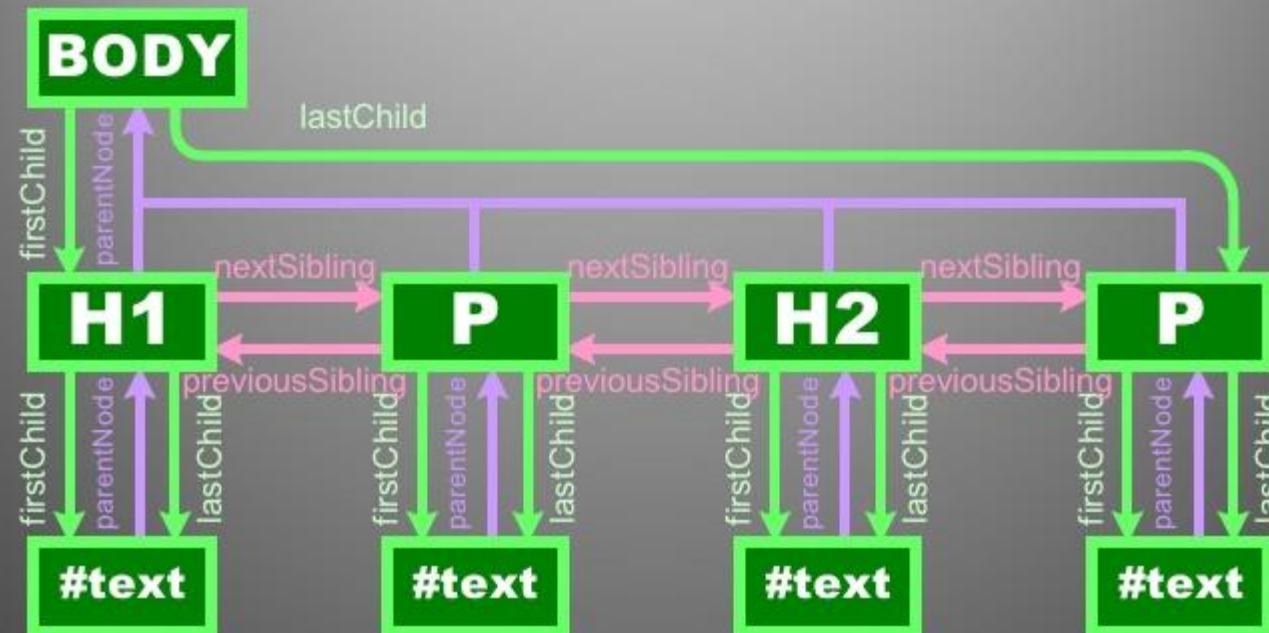


child, sibling, parent



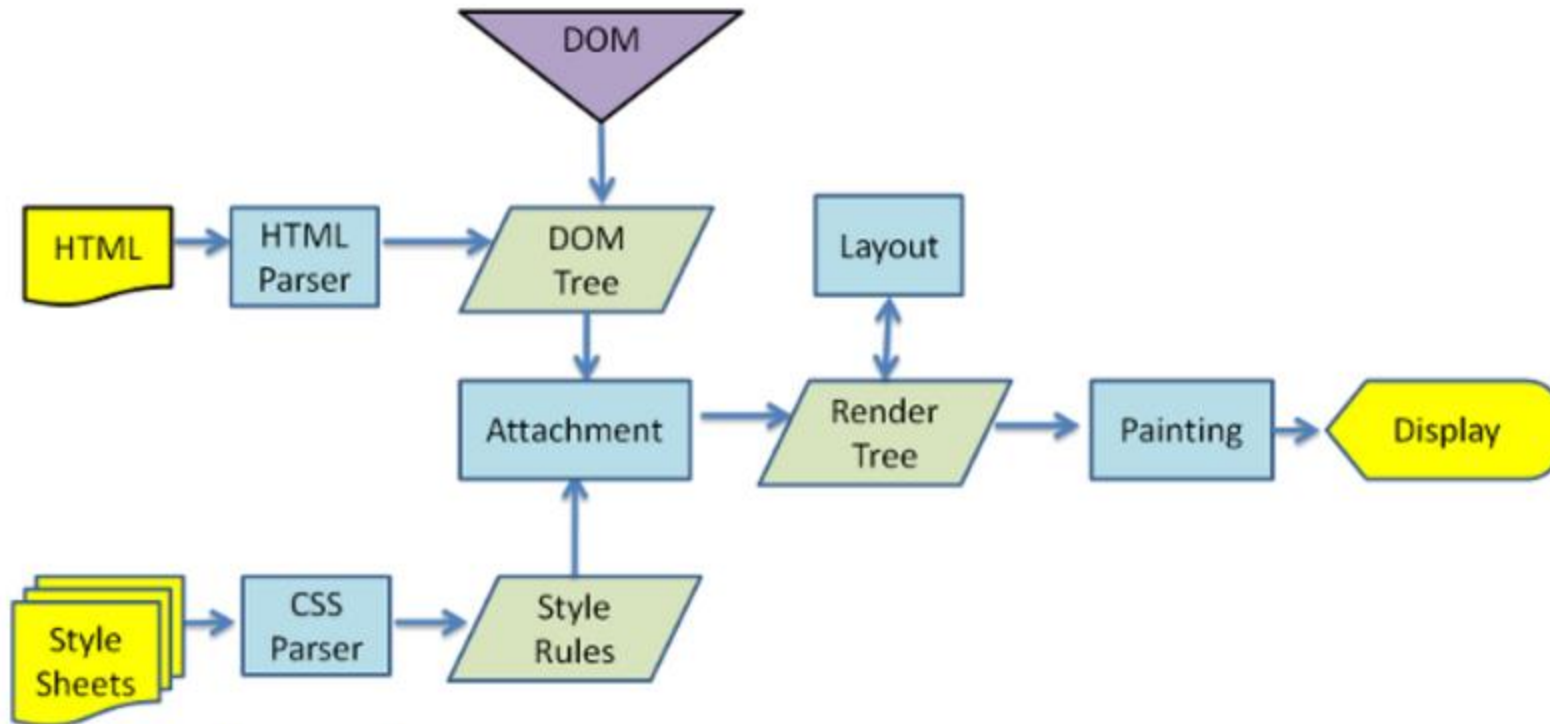
25

child, sibling, parent



26

- 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.



- 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 :
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.

- let, const keywords.
- Arrow functions.
- Template strings(``).
- Object and Array De-structuring.
- ...spread and ...rest operators.
- Promises – states (pending, fulfilled, rejected).
- Default parameters.

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