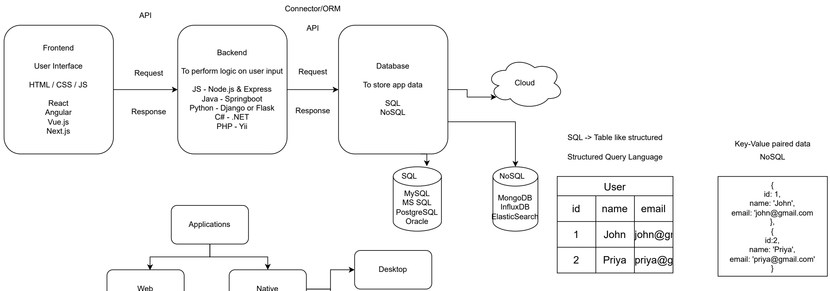
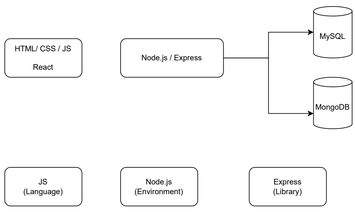


web applications dont need installation

web apps use browser resource to run

native apps require installing.

native apps use OS



Introduction to web Browser Wars

DOM tree CSSOM tree,
Browser internals - HTML parser,
CSS parser, JavaScript V8 engine,
internals

IP – MAC address – Ports &

Evolution of HTTP,
HTTP Methods

How the Server looks at the URL Request & Response cycle

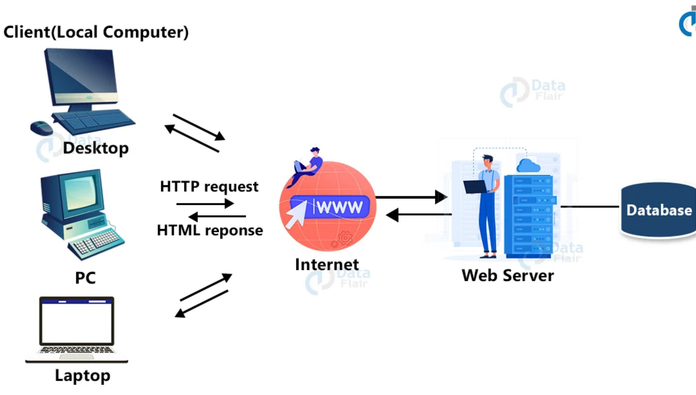
Browsers
(platforms to run web based applications
sends request to the server, gets back the response from the server)

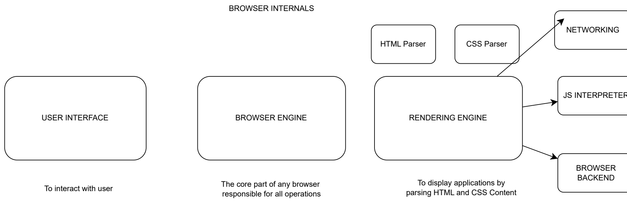
Google Chrome
Mozilla Firefox
Microsoft Edge
Safari

Chrome
Google's V8 Engine
Chromium

Firefox
SpiderMonkey

Safari
WebKit





DATA PERSISTANCE

IP ADDRESS
IPv4/ IPv6

DNS

OSI Layers

NETWORK PROTOCOLS

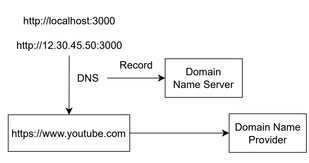
2^32 2^128

12.30.45.50 -> IP Address

127.0.0.1
localhost

2^16 ports - 0 - 65535

1000 -> App A
1001 -> App B
1002 -> App C

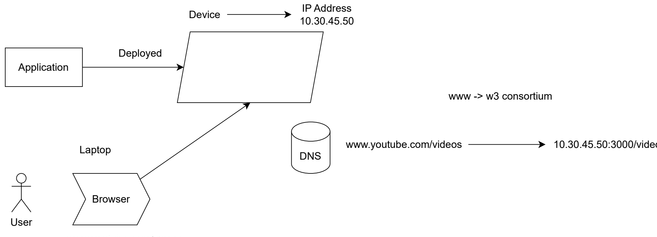


How server looks at the url

Uniform Resource Locator

https://www.example.com/products

www.hotstar.com/sports/cricket



CODEKATA

```
// JS - Javascript - Programming Language

// Scripting Language -> Javascript
// Compiled Language -> C++, Java etc

// Blocking - Synchronous
// Non-Blocking- Asynchronous

// Javascript
//  - JS is single threaded. (it will execute one execution at a time.)
//  - Non-Blocking I/O (Asynchronous) operations, because of a mechanism; (Event Loop)
//  - Used for Client and Server side programming.

// Threads - is used to carry out the process/execution one at a time.

//  - Datatype
//  - Variable
//  - Operators

//  - Looping Statements
//  - Conditional Statements
//  - Functions
//  - Objects and Arrays
//  - Class (OOPS)

//  Object Oriented programming Language

//  Typescript(OOPS) -> parent of Javascript

// - Datatype:

// data -> values or information

// Datatype ->

// Numbers - 1, 100, 50, -5, -10, 0.001, 0.6 (including integers, float, double)
// String - "priya", "john", "chennai", "india", "Cricket", "How are you?", "a", "b"
// Boolean - true,  false

// console.log("Hello")
// console.log(12345)
// console.log(true)

// // typeof - keyword used to find the type of the data

// console.log(typeof "hello");
// console.log(typeof  100);
// console.log(typeof false);

// console.log(typeof -123);
// console.log(typeof "1000");
// console.log(typeof "true")

// console.log(typeof 'Hey');


// Variable - is used to hold some values or data
// var, let, const

// Declaration and Assignment
// var username = 'John';
// let city = 'Bangalore';
// const gender = 'Male';


// console.log(username)
// console.log(city)
// console.log(gender)
// // // Declaration
// var myValue1;

// //Assignment:
// myValue1 = 1000;

// var myValue1 = 1000;


// var - can be redeclared and can be reassigned
// let - cannot be redeclared but can be reassigned
// const - cannot be redeclared and cannot be reassigned.

// var value1 = 'ABC';
// var value3 = 'PQR';
// var value1 = 'XYZ';

// value1 = 'Hello';
// console.log(value1);


// let value2 = '123';
// // let value2 = '456';
// value2 = 'How are you?';
// console.log(value2)

// const bloodGroup = 'O+ve'
// const bloodGroup = 'A-ve'
// bloodGroup = 'B +ve'

// OPERATORS:

// Arithmetic = +, -, *, /, %

// console.log(5 + 2)
// console.log(5 - 2);
// console.log(5 * 2);
// console.log(5 / 2);
// console.log(5 % 2);
// Logical
// AND (&&) - true and true = true, if any one input is false, the output will be false
// OR (||) - if anyone is true , the output will be true.if  both are false, then the output will be false.
// NOT (!) - if it is true, output will false. if it is false, output will be true.
// NAND - true and true = false, if any one input is false, the output will be true
// NOR - if anyone is true , the output will be false.if  both are false, then the output will be true.

// console.log(true && true); // true
// console.log(true || true) // true
// console.log(true || false); // true
// console.log(false && true); // false
// console.log(true); // false

// console.log(!(true || true))  // false
// console.log(!(true && false)); // true


// Comparison
// >
// <
// >=
// <=
// ==
// !=

// console.log(5 > 2) // true
// console.log(5 == 2); // false
// console.log(5 <= 5); // true
// console.log(5 != 4);  // true
// console.log(5 < 5); // false

// == -> checks only for the value
// === -> checks for the value and its datatype

// console.log(5 == '5'); // true
// console.log(5 === '5'); // false
// console.log(5 === 5); // true

// console.log( 5 != '5'); // false
// console.log(5 !== '5'); // true

// String
// console.log(5 + 5); //10
// console.log('5' + '5'); //55 -> concatenation -> appending;


// Conditonal or Ternary OPERATORS

// ? :
// condition ? statement1 : statement2

// 5 > 20 ? console.log('5 is greater') : console.log('5 is not greater');


let value1 = '1000';
console.log(typeof value1); // String


let value2 = parseInt(value1);

console.log(typeof value2)

let value3 = 'abc';

let value4 = parseInt(value3)

console.log(value4)

let value5 = 1234.44;
let value6 = parseInt(value5);

console.log(value5)

console.log(value5.toFixed(3)) // number of digits from decimal point

console.log(value5.toPrecision(9)) // number of digits from the start.


// Object / Arrays

// arrays -> []
// object -> {}

// arrays

let fruits = ['apple', 'mango', 'orange', 'watermelon', 'pineapple']; // collection of strings
let marks = [80, 90, 95, 92, 78]; // collection of number
var myArray = ['Blue', 1200, true, -10]

console.log(fruits);
console.log(marks);

// array index starts with 0 and goes upto (length - 1)

console.log(fruits[3]); //watermelon
console.log(marks[0]). // 80

console.log(marks[5]) //undefined

console.log(fruits.length)

fruits[3] = 'lemon';

fruits[8] = 'jackfruit'

console.log(fruits)
console.log(fruits[7]); //undefined

console.log(fruits.length)


// Object

// {
//     key: value
// }

let userDetails = {
```

```
    name: 'John',
    city: 'Bangalore',
    email: 'john@gmail.com',
    mobileNumber: '+91 1234567890'
}

console.log(userDetails)

console.log(userDetails['email'])

console.log(userDetails.mobileNumber)

userDetails['city'] = 'Mumbai';
userDetails.mobileNumber = '+91 0987654321'

console.log(userDetails)

// let key = 'email';

// console.log(userDetails[key]); // userDetails['email']

// console.log(userDetails.key);
```

```
// JS - Javascript - Programming Language

// Scripting Language -> Javascript
// Compiled Language -> C++, Java etc

// Blocking - Synchronous
// Non-Blocking- Asynchronous

// Javascript
//  - JS is single threaded. (it will execute one execution at a time.)
//  - Non-Blocking I/O (Asynchronous) operations, because of a mechanism; (Event Loop)
//  - Used for Client and Server side programming.


// Threads - is used to carry out the process/execution one at a time.


//  - Datatype
//  - Variable
//  - Operators

//  - Looping Statements
//  - Conditional Statements
//  - Functions
//  - Objects and Arrays
//  - Class (OOPS)

//  Object Oriented programming Language

//  Typescript(OOPS) -> parent of Javascript

// - Datatype:

// data -> values or information

// Datatype ->

// Numbers - 1, 100, 50, -5, -10, 0.001, 0.6 (including integers, float, double)
// String - "priya", "john", "chennai", "India", "Cricket", "How are you?", "a", "b"
// Boolean - true,  false

// console.log("Hello")
// console.log(12345)
// console.log(true)

// // typeof - keyword used to find the type of the data

// console.log(typeof "hello");
// console.log(typeof 100);
// console.log(typeof false);

// console.log(typeof -123);
// console.log(typeof "1000"); // '100.25' -> parseInt('100.25') // parseFloat('100.25')
// console.log(typeof "true")

// console.log(typeof 'Hey');


// Variable - is used to hold some values or data
// var, let, const

// Declaration and Assignment
// let username = 'John';
// username = 'Preeti';
// let city = 'Bangalore';
// const gender = 'Male';


// console.log(username)
// console.log(city)
// console.log(gender)


// // // Declaration
// var myValue1;

// //Assignment:
// myValue1 = 1000;

// var myValue1 = 1000;


// var - can be redeclared and can be reassigned
// let - cannot be redeclared but can be reassigned
// const - cannot be redeclared and cannot be reassigned.

// var value1 = 'ABC';
// var value3 = 'PQR';
// var value1 = 'XYZ';

// value1 = 'Hello';
// console.log(value1);


// let value2 = '123';
// // let value2 = '456';
// value2 = 'How are you?';
// console.log(value2)

// const bloodGroup = 'O+ve'
// const bloodGroup = 'A-ve'
// bloodGroup = 'B +ve'

// OPERATORS:

// Arithmetic = +, -, *, /, %

// console.log(5 + 2)
// console.log(5 - 2);
// console.log(5 * 2);
// console.log(5 / 2);
// console.log(5 % 2);


// Logical

// AND (&&) - true and true = true, if any one input is false, the output will be false
// OR (||) - if anyone is true , the output will be true.if  both are false, then the output will be false.
// NOT (!) - if it is true, output will false, if it is false, output will be true.
// NAND - true and true = false, if any one input is false, the output will be true
// NOR - if anyone is true , the output will be false.if  both are false, then the output will be true.

// console.log(true && true); // true
// console.log(true || true) // true
// console.log(true || false); // true
// console.log(false && true); // false
// console.log(!true); // false

// console.log(!(true || true))  // false
// console.log(!(true && false)); // true


// Comparison
// >
// <
// >=
// <=
// ==
// !=

// console.log(5 > 2) // true
// console.log(5 == 2); // false
// console.log(5 <= 5); // true
// console.log(5 != 4);  // true
// console.log(5 < 5); // false

// == -> checks only for the value
// === -> checks for the value and its datatype

// console.log(5 == '5'); // true
// console.log(5 === '5'); // false
// console.log(5 === 5); // true

// console.log( 5 != '5'); // false
// console.log(5 !== '5'); // true

// String
// console.log(5 + 5); //10
// console.log('5' + '5'); //55 -> concatenation -> appending;


// Conditonal or Ternary OPERATORS

// ? :
// condition ? statement1 : statement2

// 5 < 20 ? console.log('5 is greater') : console.log('5 is not greater');


// let value1 = '1000';
// console.log(typeof value1); // String


// let value2 = parseInt(value1);

// console.log(typeof value2)

// let value3 = 'abc';

// let value4 = parseInt(value3)

// console.log(value4)

// let value5 = 1234.44;
// let value6 = parseInt(value5);

// console.log(value5)

// console.log(value5.toFixed(3)) // number of digits from decimal point

// console.log(value5.toPrecision(9)) // number of digits from the start.


// // Object / Arrays

// // arrays -> []
// // object -> {}

// // arrays

// let fruits = ['apple', 'mango', 'orange', 'watermelon', 'pineapple']; // collection of strings
// let marks = [80, 90, 95, 92, 78]; // collection of number
// var myArray = ['Blue', 1200, true, -10]

// console.log(fruits);
// console.log(marks);

// // array index starts with 0 and goes upto (length - 1)

// console.log(fruits[3]); //watermelon
// console.log(marks[0]); // 80

// console.log(marks[5]) //undefined

// console.log(fruits.length)

// fruits[3] = 'lemon';

// fruits[8] = 'jackfruit'

// console.log(fruits)
// console.log(fruits[7]); //undefined

// console.log(fruits.length)


// // Object

// // {
// //     key: value
// // }

// let userDetails = {
//     name: 'John',
//     city: 'Bangalore',
//     email: 'johnl@gmail.com',
```

```
//     mobileNumber: '+91 1234567890'
// }

// console.log(userDetails)

// console.log(userDetails['email'])

// console.log(userDetails.mobileNumber)

// userDetails['city'] = 'Mumbai';
// userDetails.mobileNumber = '+91 0987654321'

// console.log(userDetails)

// // let key = 'email';

// // console.log(userDetails[key]); // userDetails['email']

// // console.log(userDetails.key);

// // Looping
// //Conditional Statements
// // // Functions -> Methods
// // Array Function or methods

// // Looping

// let myName = 'Ajay';

// console.log(myName);

// let myArrayValues = [1, 2, 3 ,4 ,5 ,6 ,7 ,8 ,9 ,10];

// console.log(myArrayValues)

// // Looping -> to access each value inside dynamic array;

// // initialization;
// // condition;
// // incrementation;

// // for Loop
// // while
// // do...while

// // For Loop

// for(initialization; condition; incrementation){

// // }
// // i++
// // // i = i + 1
// console.log('Using For Loop')
// for(let i = 0; i <= 5; i++){
//     console.log(i);
// // }


// console.log('Using While Loop')
// let x = 1;
// while(x <= 10){
//     console.log(x)
//     x = x+ 5;
// // }



// console.log('Using Do While Loop')


// let y = 1;

// do{
//     console.log(y)
//     y++;
// }while(y < 1);


// console.log('Printing numbers: ')

// let myNum = [10, 20, 30, 40, 50];

// for(let i = 0; i < myNum.length; i++){

//     console.log('Value at index ' + i + ' is ' + myNum[i]);

// // }

// // Conditional Statements:

// // // if

// // if(5 > 2){
//     console.log('5 is greater')
// // }

// // if(5 < 2){
//     console.log('5 is greater')
// // }else{
//     console.log('5 is not greater')
// // }

// let myNewValue = 22;

// // if(myNewValue % 2 === 0){
//     console.log('It is even number');
//     // if(myNewValue % 3 === 0){
//         console.log('It is divisible by 3')
//     // }else{
//         console.log('It is not divisible by 3')
//     // }
// // }else{
//     console.log('It is an odd number')
// // }




// // // switch

// let month = 11;

// // let monthInString = 'JAN';

// // switch(month){
//     // case 'JAN':
//         console.log('January');
//         // break;
//     // case 2:
//         console.log('Feb');
//         // break;
//     // case 'MAR':
//         console.log('Mar');
//         // break;
//     // case 4:
//         console.log('Apr');
//         // break;
//     // case 5:
//         console.log('May');
//         // break;
//     // case 6:
//         console.log('Jun');
//         // break;
//     // case 7:
//         console.log('Jul');
//         // break;
//     // case 8:
//         console.log('Aug');
//         // break;
//     // case 9:
//         console.log('Sept');
//         // break;
//     // case 10:
//         console.log('Oct');
//         // break;
//     // case 11:
//         console.log('Nov');
//         // break;
//     // case 12:
//         console.log('Dec');
//         // break;

//     // default:
//         console.log('Not a valid number;')
// // }


// // if..else
// // nested if
// // switch

// // Functions:

// // Syntax

// // function functionName(){
//     //logic
// // }

// // // Normal Function
// // function myFunc(){
//     console.log('Hi, How are you!')
// // }

// // let myValue1 = 'Hello';
// // let myValue2 = 100;
// // // arrow function
// // // let myArrowFunction = () => {
// //     console.log('Hi, How are you!')
// // // }

// // // myArrowFunc();



// // myFunc();

// // // Parameterized Function
// // // function greetings(name){
// //     console.log('Hi' + name);
// //     console.log('Hope you are good doing!')
// // // }

// // function greetings(birthDate, age){
//     console.log('Seems like you are celebrating your birthday on ', birthDate);
//     console.log('Wishing you a Happy ', age, 'th Birthday!');
// // }

// // greetings('Priya');

// // greetings('25th Oct', 30);


// // greetings('Rahul')

// // function printUserDetails(name, email, address, mobileNumber){
//     console.log('Please find the user information:');
//     console.log('Name: ', name);
//     console.log('Email: ', email);
//     console.log('Address: ', address);
//     console.log('Contact Number : ', mobileNumber)
// // }


// // //Function call
// // printUserDetails('Karthik', 'karthik@gmail.com', 'Mumbai', '+91 9999999999');

// // printUserDetails('Ramya');


// // function sumOfTwoValues(a, b){
// //     return a + b;
// // }

// // function findOddorEven(num){
// //     if(num % 2 === 0){
// //         return 'Even';
// //     }else{
// //         return 'Odd'
// //     }
// // }

// // let IsEvenOrOdd = findOddorEven(156);
// // console.log('The number is ', IsEvenOrOdd);



// // let result = sumOfTwoValues(5, 3);

// // console.log(result);

// // let myName = 'Deepika Padukone';

// // console.log(myName);

// // console.log(myName[4]);

// // console.log(myName.length)


// // Array Functions: (Built in functions);

// // let myValues = [100, 200, 300, 400, 500, 600, 700];

// // // ForEach  -> to iterate each value in an array
// // myValues.forEach(() => {

// // });


// // // map -> to iterate each value in an array
// // myValues.map()
```

```
// myValues.map();

// myValues.split();

// myValues.join();


// forEach
// map

// split
// join


//Difference between function and generator in JS:


// Function - if it called it will be executed only once.

// Generator - if it is subscribed, it will be keep on executed.
// yield
```

```
// Array function

let myValues = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100];


let outputMap = myValues.map((value, index) => {
// console.log('My Map Value: ', value)
console.log('Index: ', index)
return value * 5;
});

myValues.forEach((value, abc) => {
// console.log('My ForEach Value: ', value)
console.log('Output:', value * 5)

})


console.log('Map OutPut: ', outputMap)


let filterOutput = myValues.filter((value) => {
if(value > 50){
return value
}
})

let findOutput = myValues.find((value) => {
if(value > 50){
return value
}
})

// let filteredMapOutput = myValues.map((value) => {
//  if(value > 50){
//  return value
//  }
// })


console.log('Filter Output: ', filterOutput);

console.log('Find Output: ', findOutput);
// console.log('Filtering using Map Output: ', filteredMapOutput);
// myValues.map(function (){

// })
// forEach
// map
// filter
// find


let fewNames = ['AB-De-Villiers', 'Mahendra-Singh-Dhoni', 'Shah-Rukh-Khan', 'VVS-Laxman'];


let convertedNames = fewNames.map((name) => {
let splittedName = name.split('-');
let joinedName = splittedName.join(' ');
return joinedName
})

console.log('Converted Name: ', convertedNames)


// split is used to convert string into array by performing some partition
let myName = 'AB-De-Villiers';
console.log(myName)

let result = myName.split('-');

console.log(result);
// ['AB', 'De', 'Villiers']

// join is used to convert array into string with the character in between

let outputString = result.join(' ')
// AB____De____Villiers

console.log('Output after join: ', outputString)

//split
// join

// myValues.map();
// myValues.filter();
// myValues.forEach();
// myValues.split();
// myValues.join();


// Map - iterates each value in array, and returns a new array of same length.
// ForEach = iterates each value in array, but doesnt return anything
// Filter - iterates each value and returns an array of values that are matching the condition.
// find - iterates each values and returns only the first occuring value that is matching the condition.

// split - splits string into array using a character
// join = joins all elements in array using a character mentioned.


// Function  using function keyword
// Arrow Function


// function myFunc(){
//  console.log('Hello');
// }

// function myFunc2(param1, param2){
//  console.log('Param 1', param1);
//  console.log('Param 2', param2)
// }


// // Arrow Function:

// // Anonymous Arrow Function:

// function myFunction(){
//  console.log('Heyyyy')
// }

// let myArrowFunction = () => {
//  console.log('Hello');
// }


// myFunction();
// myArrowFunction();


// function myParameterizedFunction(param){
//  console.log('params: ', param)
// }


// let myParameterizedArrowFunction = (param) => {
//  console.log('Params: ', param);
// }

// myParameterizedFunction('How are you!')
// myParameterizedArrowFunction('Hope you are doing good!')
```

FREQUENCY OF VALUES IN ARRAY

```
// Getting input via STDIN
const readline = require("readline");

const inp = readline.createInterface({
input: process.stdin
});

const userInput = [];

inp.on("line", (data) => {
userInput.push(data);
});

inp.on("close", () => {
//start-here
//Your code goes here ... replace the below line with your code logic
let n = parseInt(userInput[0]); // converting string value into integer
let values = userInput[1].split(" ");
console.log('Values: ', userInput[1]);
let parsedValues = values.map((value) => {
return parseInt(value)
});
console.log('n: ', n)
console.log('values: ', parsedValues);

let freqObj = {};


// for(let i = 0; i < parsedValues.length; i++){
// let value = parsedValues[i];
//  if(value in freqObj){
//  freqObj[value]++
//  }else{
//  freqObj[value] = 1
//  }
// }

parsedValues.forEach((value) => {
if(value in freqObj){
freqObj[value]++
}else{
freqObj[value] = 1
}
})

console.log('Frequecy: ', freqObj)
const keys = Object.keys(freqObj);

let arr = [];

keys.forEach((key) => {
arr.push({
value: key,
count: freqObj[key]
})
})

console.log('arr: ', arr)



function mySorting(param1, param2){
if(param1.count === param2.count){
return param1.value - param2.value;
}else{
return param1.count - param2.count;
}
}

arr.sort(mySorting);

console.log('Sorted Array: ', arr);

let output = [];

arr.forEach((sortedValue) => {
output.push(sortedValue['value'])
})

let result = output.join(' ');
```

```
let result = output.join("  ");

console.log('Result: ', result);
//end-here
});
```

SUM OF ALL VALUES

```
// Getting input via STDIN
const readline = require("readline");

const inp = readline.createInterface({
    input: process.stdin
});

const userInput = [];

inp.on("line", (data) => {
    userInput.push(data);
});

inp.on("close", () => {

let value = parseInt(userInput[0]);
    let sum = 0;

for(let i = 1; i<=value; i++){
    sum += i;
    }
//start-here
//Your code goes here ... replace the below line with your code logic

console.log(sum);

//end-here
});
```