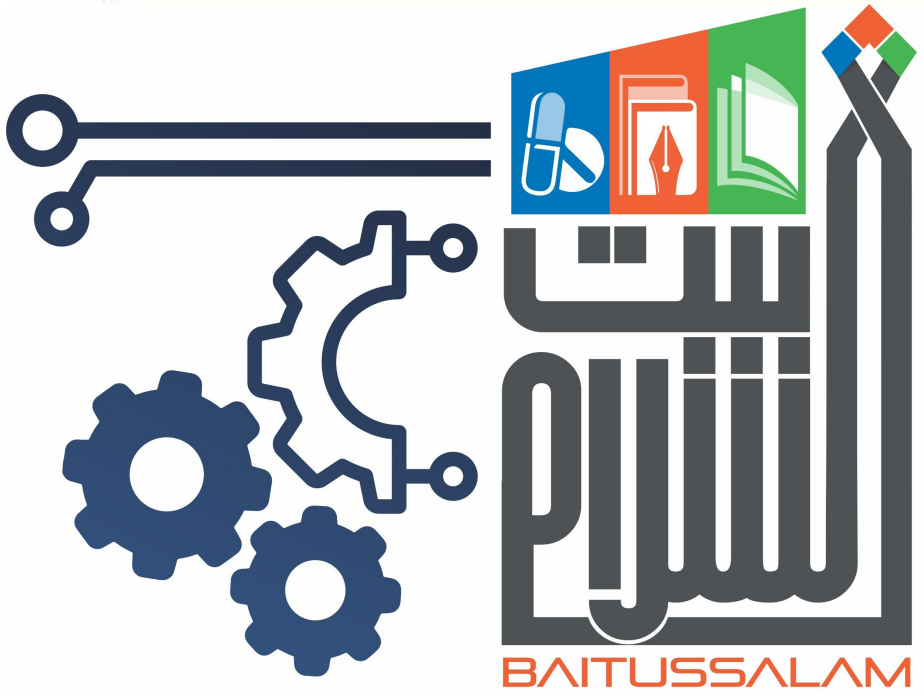

BAITUSSALAM
—TECH PARK—

Class Agenda

BAITUSSALAM
—TECH PARK—

Numbers and Object Methods, Math Functions, Problem Solving Patterns

JavaScript Numbers Methods

BAITUSSALAM
—TECH PARK—

toString() Method
change number type to string

toFixed() Method
Format a number to a fixed number of decimal places

isNaN() Method
Check if a value is NaN (Not-a-Number)

Math.round() Method
Round a number to the nearest integer

toPrecision() Method
Format a number to a specified length

toLocaleString Method
ormat a number according to local language and region

```
let num = 123.4567
let formattedNum = num.toFixed(2) // "123.46"

let result = isNaN('Hello') // true
let result2 = isNaN(123) // false

let num = 123.56
let roundedNum = Math.round(num) // 124

let num = 123.456
let preciseNum = num.toPrecision(4) // "123.5"

let num = 1234567.89

let localNumUS = num.toLocaleString('en-US') // "1,234,567.89"
let localNumArabic = num.toLocaleString('ar-EG') // ١,٢٣٤,٥٦٧,٨٩
let localNumInd = num.toLocaleString('en-IN') // 12,34,567.89

console.log(localNumUS) // Output: "1,234,567.89"
console.log(localNumArabic) // Output: ١,٢٣٤,٥٦٧,٨٩
console.log(localNumInd) // Output: 12,34,567.89
```

Numbers Exercise

Write a JavaScript function `reverseInteger` that accepts an integer `number` and returns its **reverse**.

The function should handle both **positive and negative integers**, and it should **remove any leading zeros** in the reversed number.

Examples

1. `reverseInteger(981)` should return `189`.
2. `reverseInteger(500)` should return `5`.
3. `reverseInteger(-15)` should return `51`.
4. `reverseInteger(-60)` should return `6`.

Maths Methods

Math.abs(): Returns the absolute value of a number
Math.floor(): Rounds a number down to the nearest integer
Math.ceil(): Rounds a number up to the nearest integer
Math.random(): Generates a random number between 0 and 1

app.js > ...

```
1   let num1 = -456.789;
2   let num2 = 123.456;
3
4   let absNum1 = Math.abs(num1); // 456.789
5   let floorNum2 = Math.floor(num2); // 123
6   let ceilNum2 = Math.ceil(num2); // 124
7
8   let randomNum = Math.random(); // e.g., 0.726
9
```

Maths Exercise #1

BAITUSSALAM
—TECH PARK—

Create a function `getRandomInteger` that generates a **random whole number** within a specified range.

The function should accept two parameters, `min` and `max`, indicating the lowest and highest values possible, both inclusive.

It should return a random integer between these values, inclusive

```
// Object.keys() => Returns an array of a given object's property names
let user = { name: 'Alice', age: 25 }
let keys = Object.keys(user) // ["name", "age"]

// Object.values() => Returns an array of a given object's property values
let values = Object.values(user) // ["Alice", 25]

// Object.entries() => Returns an array of a given object's key-value pairs
let entries = Object.entries(user) // [["name", "Alice"], ["age", 25]]

// Object.freeze() => Freezes an object, preventing new properties from being added
// or existing properties from being removed or changed
let obj = { name: 'Bob' }
Object.freeze(obj)
obj.name = 'Alice' // No effect, obj is frozen

// Object.hasOwnProperty() => Checks if an object has a specific property
//as its own property (not inherited from its prototype chain)
let car = { brand: 'Toyota', model: 'Corolla', year: 2020 }

let hasBrand = car.hasOwnProperty('brand') // true
let hasColor = car.hasOwnProperty('color') // false

// in Operator => Checks if a property exists in an object
// (including properties from the prototype chain)
let hasBrand = 'brand' in car; // true
let hasColor = 'color' in car; // false
```

JavaScript Object Methods

Function + Object Exercise #1

Write a function `decodeCipher` which will take `str` as parameter to decode or encode it.

The GADERYPOLUKI is a simple substitution cypher used in scouting to encrypt messages. The encryption is based on short, easy to remember key. The key is written as paired letters, which are in the cipher simple replacement.

The most frequently used key is "GA-DE-RY-PO-LU-KI".

```
encode("ABCD")          // => GBCE
encode("Ala has a cat") // => Gug hgs g cgt
encode("gaderypoluki"); // => agedyropulik
decode("Gug hgs g cgt") // => Ala has a cat
decode("agedyropulik")  // => gaderypoluki
decode("GBCE")          // => ABCD
```

Frequency Counter Pattern

BAITUSSALAM
—TECH PARK—

- A strategy to solve problems by counting frequencies of elements
- Helps avoid nested loops, making code more efficient
- Uses objects or maps to store counts

```
const products = [
  'mouse', 'speaker', 'keyboard', 'laptop',
  'charger', 'headphone', 'speaker', 'keyboard',
  'charger',
]

const frequencyCounter = {}

for (let product of products) {
  if (!frequencyCounter[product]) {
    frequencyCounter[product] = 1
  } else {
    frequencyCounter[product] = frequencyCounter[product] + 1
  }
}

console.log(frequencyCounter)
// {mouse: 1, speaker: 2, keyboard: 2, laptop: 1, charger: 2, headphone: 1}
```

Anagram Exercise | Frequency Counter

BAITUSSALAM
—TECH PARK—

Time Complexity:
O(n)

Given two strings, write a function `validAnagram` that will take two strings as parameters to determine if the second string is the anagram of first.

An anagram is a word, phrase or name formed by rearrangin the letters of another, cinema fromed from iceman.

```
validAnagram("","") // true
validAnagram("aaz","zza") // false
validAnagram("rat","car") // false
validAnagram("awesome","awesom") // false
validAnagram("qwerty","qeywrt") // true
```

2D Array Exercise | Chunky Monkey

Define the function `chunkyMonkey` which takes two parameters: an array `arr` and a number `size`.

It splits an array (first argument) into groups the length of `size` (second argument) and returns them as a `two-dimensional` array.

```
function chunkyMonkey(arr, size) {
    // Your code here
}

chunkyMonkey(["a", "b", "c", "d"], 2);
// should return [["a", "b"], ["c", "d"]]

chunkyMonkey([0, 1, 2, 3, 4, 5], 3);
// should return [[0, 1, 2], [3, 4, 5]]

chunkyMonkey([0, 1, 2, 3, 4, 5], 4)
// should return [[0, 1, 2, 3], [4, 5]]
```

Function + Object Exercise #2

Write a function `createCounter`. It should accept an initial integer `init`. It should return an object with three functions.

The three functions are:

- `increment()` increases the current value by 1 and then returns it.
- `decrement()` reduces the current value by 1 and then returns it.
- `reset()` sets the current value to `init` and then returns it.

Example

Input: init = 5, calls = ["increment","reset","decrement"]
Output: [6,5,4]
Explanation:
const counter = createCounter(5);
counter.increment(); // 6
counter.reset(); // 5
counter.decrement(); // 4

For Each Loop

- forEach is a built-in array method in JavaScript.
- It allows you to execute a function once for each array element.
- Useful for performing actions on each item in an array.

Syntax

```
array.forEach(function(element, index, array) {
  // code block
})
```

- `element` : The current element being processed in the array.
- `index` : (Optional) The index of the current element.
- `array` : (Optional) The array `forEach` was called upon.

Bubble Sort | Sorting Algorithms

Bubble sort algorithm is an algorithm that sorts an array by comparing two adjacent elements and swapping them if they are not in the intended order. Here order can be anything like increasing or decreasing.

The End

BAITUSSALAM
—TECH PARK—