



BAITUSSALAM
—TECH PARK—



Class Agenda

**Regular Expression,
Recursion, Closure, ES6
Modules, and Webpack**

Regular Expression

Regular expression is a way to search string in text

Methods

test()

replace()

replaceAll()

match()

matchAll()

split()

search()

Flags

g (all)

i (case insensitive)

Quantifier

| (or)

* *Zero or more times*

+ *(match one or more)*

? *(use after character for optional)*

. *(matches any character except line break)*

Regex – Exact Matching

```
JS app.js > ...  
1  const str = 'I am learning JavaScript, We can build web app using Javascript'  
2  
3  const pattern = /JavaScript/gi  
4  
5  const result = pattern.test(str)  
6  
7  console.log(result)  
8
```

Exercise:

Replace all javascript (word) in text with python, using replace function

Output: I am learning python, We can build web app using python

Regular Expression

Do not match with ^ caret symbol

Metacharacters

\w (any word character)

\s (any whitespace)

\d (any decimal digit (0-9))

**Metacharacters with
capital works the opposite**

Characters with specific length

\w{4,6}/

[fc] matches any character inside set

[a-zA-Z] range a to z

[0-5] range 0-5

(c|C)at group, lower case or uppercase c following at

Regular Expression Exercises

- Match all the articles in paragraphs, article are “a, an, the”
- Select all the words in paragraph that start with vowel
- Select all words that are consist of 5 alphabets or more

Regex Exercise #1

Spinal Tap Case

Write a function *spinalCase(str)* to convert a string to spinal case. Spinal case is all-lowercase-words-joined-by-dashes.

For example

This Is Spinal Tap	=>	this-is-spinal-tap
The_Andy_Griffith_Show	=>	the-andy-griffith-show

Recursion

Recursion is a technique where a function calls itself to solve a problem.

- **Base Case:**

The condition under which the recursive function stops calling itself to prevent infinite loops.

- **Recursive Case:**

The part of the function where it calls itself with a modified argument, gradually approaching the base case.

Recursion Exercise

Solving Factorial using recursion

```
46  function factorial(num) {  
47      if (num === 0) {  
48          return 1  
49      }  
50  
51      console.log(num)  
52  
53      return num * factorial(num - 1)  
54  }  
55
```

Closure

A closure in JavaScript is a function that has access to variables in its parent scope, even after the parent function has returned. Closures are created when a function is defined inside another function, and the inner function retains access to the variables in the outer function's scope.

Closure Example

Closures can be used to create functions that generate other functions.

This is useful in scenarios where you need similar functions with different initial parameters.

```
64 function createGreeting(greeting) {  
65     return function (name) {  
66         return `${greeting} ${name}`  
67     }  
68 }  
69  
70 const sayHi = createGreeting('Hi')  
71 const sayHello = createGreeting('Hello')  
72  
73 const result1 = sayHi('Shariq')  
74 const result2 = sayHello('Hamza')  
75  
76 console.log(result1)  
77 console.log(result2)  
78
```

Closure Exercise

Create a function named **multiply** accept one parameter and return a function that also accept one parameter.

This will form a closure.

Now inner function will return product of main function and inner function parameter.

Create multiple versions of main function with different parameters, and store in variable.

Expected Output:

```
console.log(double(5)); // 10
```

```
console.log(triple(5)); // 15
```

Understanding ES6 Modules

ES6 modules allow us to break our code into separate files (modules) which can be imported and exported.

Benefits:

Improved code organization, reusability, and maintainability.

Exporting Modules

Default Export: A module can export one default value

```
77 export default function greet() {  
78 |   console.log('Hello, world!')  
79 }  
80
```

Named Export: A module can export multiple values

```
81 export const pi = 3.14159  
82  
83 export function multiply(a, b) {  
84 |   return a * b  
85 }  
86
```


package.json

package.json file helps developers manage project dependencies, automate common tasks, and provide important metadata about the project.

package-lock.json

serves as a lockfile that captures the exact versions of packages and their dependencies

Webpack

Webpack is used, to compile JavaScript modules.

It's a tool that lets you bundle your JavaScript applications (supporting both ESM and CommonJS), and it can be extended to support many different assets such as images, fonts and stylesheets.

The End

