JavaScript

Session 1&2

JavaScript

- High level
- Interpreted
- Event-driven

Why JavaScript

- Client-side execution
- Interactivity: dynamic UI and animations
- Versatility: frontend (React, Angular, Vue), backend (Node.js)
- Wide ecosystem: libraries & frameworks

Variables

	Scope	Can be re-assigned?	Hoisting
var	Function-wide	✓	✓ With undefined
let	Block-scoped	✓	✗
const	Block-scoped	✗	✗

```
{
  let x = 2;
}
// x can NOT be
used here
```

```
{
  Var x = 2;
}
// x CAN be
used here
```

Lists

```
let places = ["Dubai", "Riyadh", “Cairo"];
```

```
let mixedArray = ["Cairo", 5, true, null];
```

List Methods

- push(..);
- pop();
- unshift(..);
- shift();
- splice(1, 1, ..);
- length;
- sort(); Sorts as strings by default
- reverse();
- indexOf(..);
- includes(..);
- join(", ");
- slice(1, 3);
- concat([………]); OR Spread Operator […listtname, new items]

Conditionals

- If Else
- Ternary Operator
- Switch Case

Loops

- For loop
- While loop
- Do-while loop
- For-in loop

Functions

- Defining
- Default Parameters
- With Return
- Arrow Function
- Function Expression

Objects

- Accessing properties
- Adding/deleting properties
- Object with constructor
- Object prototype

Spread Operator

- With lists
- With objects
- With variables

Advanced List Methods

- For-each
- Map
- Filter
- Reduce