

# JavaScript Assignment – Topic-wise Notes (ES6+)

## Core JavaScript

**1. let and const:** Block scoped. let allows reassignment; const does not.

```
let x = 10; x = 20; const y = 5;
```

**2. Data Types:** Primitive (number, string, boolean, null, undefined, NaN) and Reference (object, array).

**3. Truthy/Falsy:** false, 0, "", null, undefined, NaN are falsy.

**4. Operators:** == vs ===, &&, ||, ?:

```
let res = age >= 18 ? 'Adult' : 'Minor';
```

**5. Conditionals:** if-else and ternary.

## Functions

**6. Declaration vs Expression:** Declaration is hoisted.

```
function add(a,b){return a+b;} const sub=function(a,b){return a-b;}
```

**7. Arrow Functions:** Short syntax, lexical this.

```
const mul=(a,b)=>a*b;
```

**8. Default Parameters:** Used when value not passed.

```
function greet(name='Guest'){return name;}
```

**9. Return:** Sends value back from function.

## Objects & Arrays

**10. Objects:** Key-value pairs.

```
const user={name:'A',age:25};
```

**11. Arrays:** Ordered list.

```
const arr=[1,2,3];
```

**12. Array Methods:** map, filter, find, reduce, some, every.

```
arr.map(x=>x*2); arr.filter(x=>x>1); arr.reduce((a,b)=>a+b,0);
```

## Modern JavaScript (ES6+)

**13. Destructuring:** Extract values easily.

```
const {name}=user; const [a]=arr;
```

**14. Spread:** Copy/merge.

```
const nArr=[...arr,4];
```

**15. Rest:** Collect args.

```
function sum(...n){return n.length;}
```

**16. Template Literals:**

```
`Hello ${name}`
```

## Modules & Classes

**17. import/export:** Share code.

```
export const x=5; import {x} from './file.js';
```

**18–21. Classes:** constructor, extends, super, this.

```
class A{constructor(n){this.n=n;}}
```

## Async JavaScript

**22–25:** Callbacks, Promises, async/await, try-catch.

```
async function load(){await fetch(url);}
```

## Advanced Concepts

**26–29:** Scope, Closures, Immutability, Higher-order functions.

## Browser & Runtime

**30–34:** DOM, Events, Bubbling, Storage, JSON.