1.What is Java Script ?

JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

2.Data types in JavaScript ?

Primitive data type =>

Number : let age =32;
String : let name='manish'; let name="manish"
Boolean: let fullage=true;
Undefined: let children;
Null : {} or empty value
Symbol: value that is unique and can't be changed
BigInt : to hold larger integers than the Number type.

Object data type =>

Let myObject={
firstName:'Manish'
};

3.Different ways of declaring variables in JS?

let : can't be re-declared but can be updated within same scope. let is block scoped.
Var: can be re- declaredand can be updated within same scope .var is function scoped.
Const : block scope, with const requiring an initial value and preventing reassignment.

4.Type Conversion and Type Coercion

```
// Type Conversion
const age='32';
console.log(typeof(age));
const ageNew=Number(age);
console.log(typeof(ageNew)); // type conversion

// Type coercion

const stradd='I am ' + 23 + 'years old'; // converts integer into string
console.log(stradd);
console.log('23'-10-'3'); // converts string into integer

let n='1'+1; // this will become 11 with type String
console.log(typeof(n));
n-=1; // this will become 10 with type integer
console.log(n);
console.log(typeof(n));
```

5.Falsy Values : NaN , null , undefined , 0 , ‘ ‘.

6.== vs ===

"==" operator compares values after performing type conversion,
“===” operator compares values without type conversion.

```
const a=20;
const b=21;
const c='20';
const d='21';



if(a==c){ // operator compares values after performing type conversion,
   console.log("Equal NUmbers"); // this will get printed in console
}
else{
   console.log("Unequal Numbers");
}

if(a===c){ //operator compares values without type conversion,
   console.log("Equal NUmbers");
}
else{
   console.log("Unequal Numbers"); // thiswill get printed in console
}
```

7.Statement vs Expression

```
let a=10;
let b=20;

console.log(a*b); // expression
if(a<b){
   console.log("statement");
}
```

8.Conditional Ternary Operator

```
const age=18;

age >=18?console.log("You can drink"):console.log("You can’t drink")
```

9.Function Declaration vs Function Expressions

```

// function declaration
function printName(fullName){
   console.log(fullName);
   return true;
}

fLname="Manish Patwal";
//function expression
fun=printName(fLname);
fun==true?console.log("correct"):console.log("Incorrect");
```

10.Arrow Function

```

// function expression
const birthYear = 1991;
const calcAge2 = function(birthYear){
   return 2024-birthYear;
}

// Arrow Function equivalent to above function

const calcAge3 = birthYear => 2024-birthYear
```

11. Array

```
// typed array  - contains similar data type and dynamic in nature
const fattuGang = ['Bisht','Verma','kanojia','Joshi','Nautiyal'];
console.log(typeof(fattuGang));//object
```

```
//Javascript array - contains different data type and dynamic in nature
const karz =['manish',1467900 ,"priyakshi" , 153000,true];
console.log(typeof(fattuGang)); //object
```

12. Objects

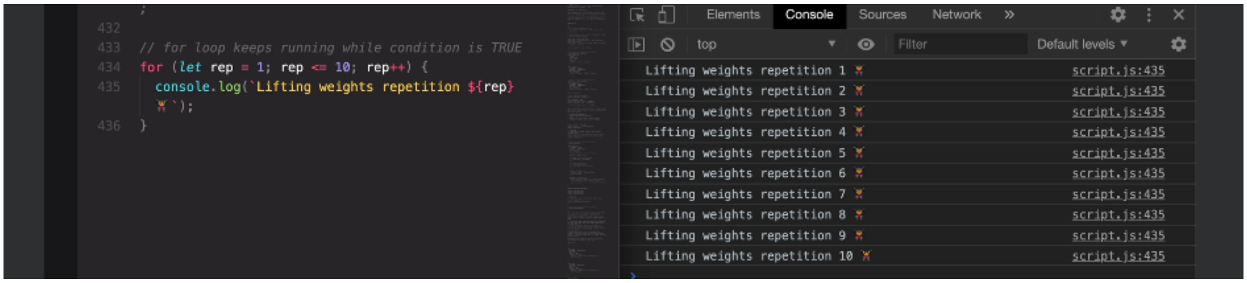
In JavaScript, objects can be seen as a collection of properties.
Object properties are equivalent to key-value pairs.

```
// Creation of object
const firstObject={
   fullName:"Manish",
   spouseName:"Priyakshi",
   Age:32,
   address_current:"Soorya splendor Bengaluru"
};

console.log(firstObject);
console.log(typeof(firstObject));

// Accessing the object via  and
console.log(firstObject.fullName); //DOT notation output: Manish
console.log(firstObject['Age']); //BRACKET notation , output : 32
```

13.For loop

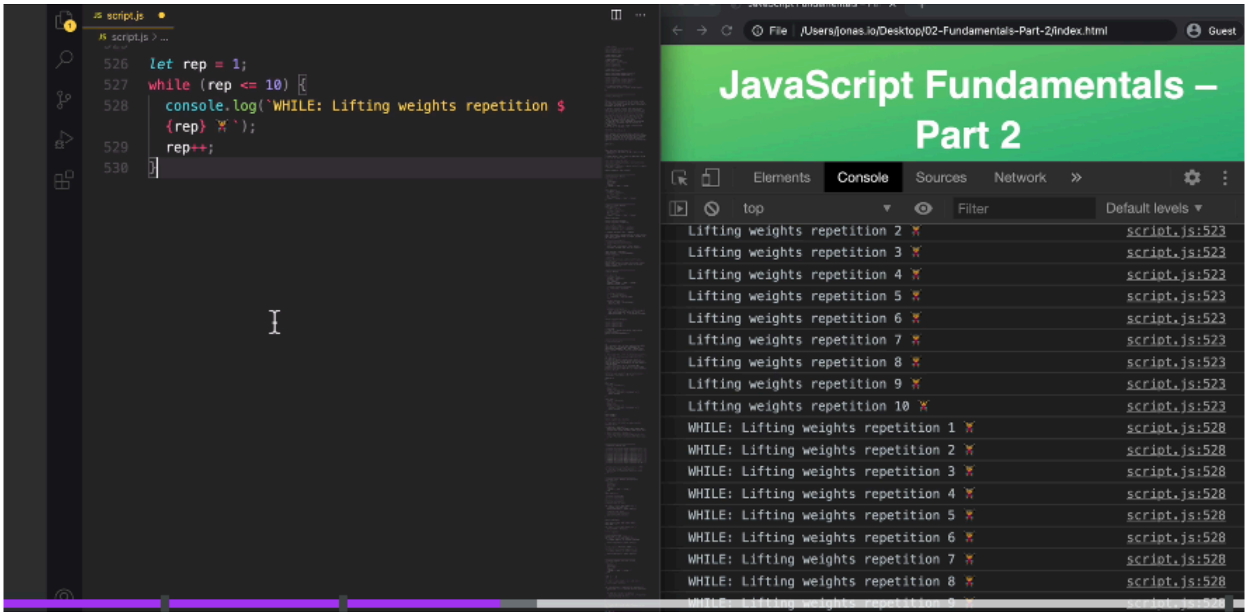


```
;

// for loop keeps running while condition is TRUE
for (let rep = 1; rep <= 10; rep++) {
  console.log(`Lifting weights repetition ${rep}
  🏋️`);
}
```

```
Lifting weights repetition 1 🏋️                    script.js:435
Lifting weights repetition 2 🏋️                    script.js:435
Lifting weights repetition 3 🏋️                    script.js:435
Lifting weights repetition 4 🏋️                    script.js:435
Lifting weights repetition 5 🏋️                    script.js:435
Lifting weights repetition 6 🏋️                    script.js:435
Lifting weights repetition 7 🏋️                    script.js:435
Lifting weights repetition 8 🏋️                    script.js:435
Lifting weights repetition 9 🏋️                    script.js:435
Lifting weights repetition 10 🏋️                   script.js:435
```

14.While loop



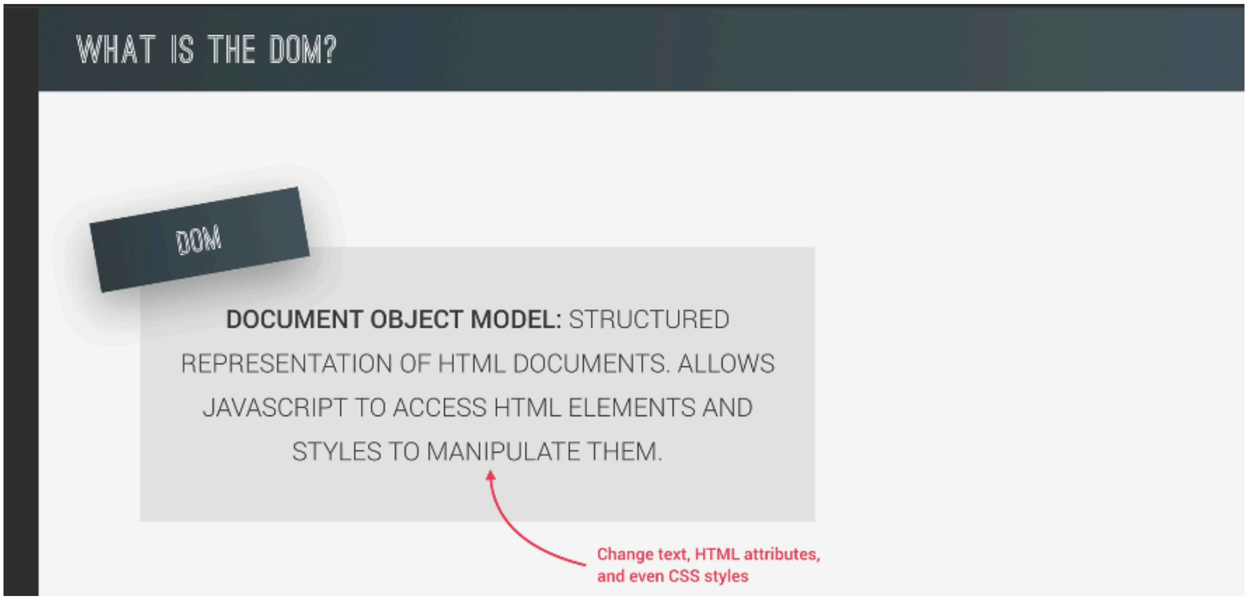
```javascript
let rep = 1;
while (rep <= 10) {
  console.log(`WHILE: Lifting weights repetition ${rep} 🏋️`);
  rep++;
}
```

/Users/jonas.io/Desktop/02-Fundamentals-Part-2/index.html

JavaScript Fundamentals – Part 2

```
Lifting weights repetition 2 🏋️                  script.js:523
Lifting weights repetition 3 🏋️                  script.js:523
Lifting weights repetition 4 🏋️                  script.js:523
Lifting weights repetition 5 🏋️                  script.js:523
Lifting weights repetition 6 🏋️                  script.js:523
Lifting weights repetition 7 🏋️                  script.js:523
Lifting weights repetition 8 🏋️                  script.js:523
Lifting weights repetition 9 🏋️                  script.js:523
Lifting weights repetition 10 🏋️                 script.js:523
WHILE: Lifting weights repetition 1 🏋️           script.js:528
WHILE: Lifting weights repetition 2 🏋️           script.js:528
WHILE: Lifting weights repetition 3 🏋️           script.js:528
WHILE: Lifting weights repetition 4 🏋️           script.js:528
WHILE: Lifting weights repetition 5 🏋️           script.js:528
WHILE: Lifting weights repetition 6 🏋️           script.js:528
WHILE: Lifting weights repetition 7 🏋️           script.js:528
WHILE: Lifting weights repetition 8 🏋️           script.js:528
WHILE: Lifting weights repetition 9 🏋️           script.js:528
```

14.DOM



WHAT IS THE DOM?

DOCUMENT OBJECT MODEL: STRUCTURED REPRESENTATION OF HTML DOCUMENTS. ALLOWS JAVASCRIPT TO ACCESS HTML ELEMENTS AND STYLES TO MANIPULATE THEM.

Change text, HTML attributes, and even CSS styles