JS basics

1. What is JavaScript?

JavaScript is a programming language that adds interactivity to your website. Propeties of JavaScript:

- High-level
- Just-in-time compiled (JIT)
- Multi-paradigm
- Prototype-based
- Garbage-collected
- Dynamically-typed
- Single-threaded
- Non-blocking event loop

2. Variables and Data Types

2.1. Variables

```
var firstName = "John";
let middleName = "Doe";
const firstName = "Jack";

// you can reassign the value of a variable

firstName = "Jane";
middleName = "Doe";

// you can't reassign the value of a constant

firstName = "Jill"; // error
```

2.2. Data Types

```
// Primitive data types

// Number: Floating point numbers, for decimals and integers
let age = 23;
const PI = 3.14;

// String: Sequence of characters, used for text

const firstName = 'John';
const lastName = 'Doe';



// Boolean: Logical data type that can only be true or false
const fullAge = true;

// Undefined: Data type of a variable that does not have a value yet
```

```
let children;
console.log(children); // undefined

// Null: Also means 'non-existent'
let children = null;


// non primitive data types

// Object: Reference data type that represents the object

const family = ['Jane', 'Mark', 'Bob'],


const john = {
 firstName: 'John',
 lastName: 'Doe',
 age: 23,
};
```

3. Operators

3.1. Arithmetic Operators

```
// + - * / ** %

const now = 2021;

const ageJohn = now - 1998; // 23
const ageMark = now - 1996; // 25

console.log(ageJohn, ageMark); // 23 25

console.log(ageJohn * 2, ageJohn / 10, 2 ** 3); // 46 2.3 8

const firstName = "John";
const lastName = "Doe";

console.log(firstName + " " + lastName); // John Doe

// Assignment operators

let x = 10 + 5; // 15

x += 10; // x = x + 10 = 25
x *= 4; // x = x * 4 = 100

x++; // x = x + 1
x--; // x = x - 1

// Comparison operators

console.log(ageJohn > ageMark); // >, <, >=, <=
const isFullAge = ageJohn >= 18; // true

console.log(now - 1998 > now - 1996); // false
```

3.2. Operator Precedence

```
const now = 2021;
const ageJohn = now - 1998; // 23
const ageMark = now - 1996; // 25

console.log(now - 1998 > now - 1996); // false

let x, y;

x = y = 25 - 10 - 5; // x = y = 10, x = 10

console.log(x, y); // 10 10

const averageAge = (ageJohn + ageMark) / 2;
console.log(ageJohn, ageMark, averageAge); // 23 25 24

const a = 2;
const b = 3;

console.log(a == b); // false
console.log(a != b); // true

console.log(a === b); // false
console.log(a !== b); // true

console.log(a == "2"); // true
console.log(a === "2"); // false
```

3.3. Strings and Template Literals

```
const firstName = "John";
const job = "teacher";
const birthYear = 1998;
const year = 2021;

const john =
 "I'm " + firstName + ", a " + (year - birthYear) + " years old " + job + "!";
console.log(john); // I'm John, a 23 years old teacher!

const johnNew = `I'm ${firstName}, a ${year - birthYear} years old ${job}!`;

console.log(johnNew); // I'm John, a 23 years old teacher!

console.log(`Just a regular string...`); // Just a regular string...

console.log(
 "String with \n\
multiple \n\
lines"
); // String with
// multiple
// lines
```

3.4. If-Else Statements

```
const firstName = "John";
const civilStatus = "single";

if (civilStatus === "married") {
 console.log(`${firstName} is married!`); // John is married! }
else {
 console.log(`${firstName} is single!`); // John is single!
}

const isMarried = true;

if (isMarried) {
 console.log(`${firstName} is married!`); // John is married! }
else {
 console.log(`${firstName} is single!`); // John is single!
}

const markMass = 78;
const markHeight = 1.69;

const johnMass = 92;
const johnHeight = 1.95;

const markBMI = markMass / markHeight ** 2;

const johnBMI = johnMass / johnHeight ** 2;

if (markBMI > johnBMI) {
 console.log(`Mark's BMI (${markBMI}) is higher than John's (${johnBMI})!`); }
else {
 console.log(`John's BMI (${johnBMI}) is higher than Mark's (${markBMI})!`); }
```

3.5. Type Conversion and Coercion

```
// Type Conversion

const inputYear = "1998";
console.log(Number(inputYear), inputYear); // 1998 '1998'
console.log(Number(inputYear) + 18); // 2016

console.log(Number("John")); // NaN
console.log(typeof NaN); // number

console.log(String(23), 23); // '23' 23

// Type Coercion
```

```
console.log("I am " + 23 + " years old"); // I am 23 years old
console.log("23" - "10" - 3); // 10
console.log("23" + "10" + 3); // 23103
console.log("23" * "2"); // 46
console.log("23" / "2"); // 11.5
console.log("23" > "18"); // true

let n = "1" + 1; // '11'
n = n - 1; // 10
console.log(n); // 10
```

3.6. Truthy and Falsy Values

```
// 5 falsy values: 0, '', undefined, null, NaN

console.log(Boolean(0)); // false
console.log(Boolean(undefined)); // false
console.log(Boolean("John")); // true
console.log(Boolean({})); // true
console.log(Boolean("")); // false

const money = 0;

if (money) {
 console.log("Don't spend it all"); // Don't spend it all
} else {
 console.log("You should get a job!"); // You should get a job! }

let height;

if (height) {
 console.log("YAY! Height is defined"); // YAY! Height is defined }
else {
 console.log("Height is UNDEFINED"); // Height is UNDEFINED }
```

3.7. Equality Operators: == vs. ===

```
const age = "18";

if (age === 18) console.log("You just became an adult! (strict)"); // undefined if (age == 18)
console.log("You just became an adult! (loose)"); // You just became an adult! (loose)

const favourite = Number(prompt("What's your favourite number?"));

console.log(favourite); // 23
console.log(typeof favourite); // number

if (favourite === 23) {
 console.log("Cool! 23 is an amazing number!"); // Cool! 23 is an amazing number! }
else if (favourite === 7) {
 console.log("7 is also a cool number"); // 7 is also a cool number
} else {
 console.log("Number is not 23 or 7"); // Number is not 23 or 7
}

if (favourite !== 23) console.log("Why not 23?"); // undefined
```

3.8. Boolean Logic

```
const hasDriversLicense = true;
const hasGoodVision = true;

console.log(hasDriversLicense && hasGoodVision); // true
console.log(hasDriversLicense || hasGoodVision); // true
console.log(!hasDriversLicense); // false

const shouldDrive = hasDriversLicense && hasGoodVision; // true

if (shouldDrive) {
 console.log("Sarah is able to drive!"); // Sarah is able to drive!
} else {
 console.log("Someone else should drive..."); // Someone else should drive... }

const isTired = false;

console.log(hasDriversLicense && hasGoodVision && isTired); // false

if (hasDriversLicense && hasGoodVision && !isTired) {
 console.log("Sarah is able to drive!"); // Sarah is able to drive!
} else {
 console.log("Someone else should drive..."); // Someone else should drive... }
```

3.9. The switch Statement

```
const day = "monday";

switch (day) {
 case "monday": // day === 'monday'
 console.log("Plan course structure"); // Plan course structure
console.log("Go to coding meetup"); // Go to coding meetup  break;
 case "tuesday":
 console.log("Prepare theory videos");
 break;
 case "wednesday":
 case "thursday":
 console.log("Write code examples");
 break;
 case "friday":
 console.log("Record videos");
 break;
 case "saturday":
 case "sunday":
 console.log("Enjoy the weekend :D");
 break;
 default:
 console.log("Not a valid day!");
}

if (day === "monday") {
 console.log("Plan course structure"); // Plan course structure
console.log("Go to coding meetup"); // Go to coding meetup } else if
(day === "tuesday") {
 console.log("Prepare theory videos");
} else if (day === "wednesday" || day === "thursday") {
 console.log("Write code examples");
} else if (day === "friday") {
 console.log("Record videos");
} else if (day === "saturday" || day === "sunday") {
 console.log("Enjoy the weekend :D");
} else {
 console.log("Not a valid day!");
}
```

3.10. Statements and Expressions

```
// Expressions

3 + 4;
1991;
true && false && !false;

// Statements

if (23 > 10) {
 const str = "23 is bigger";
}

const me = "Jonas";
console.log(`I'm ${2037 - 1991} years old ${me}`); // I'm 46 years old Jonas
```

3.11. The Conditional (Ternary) Operator

```
const age = 23;

age >= 10
 ? console.log("I like to drink juice")
 : console.log("I like to drink water"); // I like to drink juice

const drink = age >= 10 ? "juice" : "water";

console.log(drink); // juice

let drink2;
if (age >= 10) {
 drink2 = "juice";
} else {
 drink2 = "water";
}

console.log(drink2); // juice

   console.log(`I like to drink ${age >= 10 ? "juice" : "water"}`); // I like to drink juice
```

4. Data Structures, Modern Operators and Strings

4.1. A Closer Look at Arrays

```
const friends = ["Michael", "Steven", "Peter"];
console.log(friends); // ['Michael', 'Steven', 'Peter']

const y = new Array(1991, 1984, 2008, 2020);

console.log(friends[0]); // Michael
console.log(friends[2]); // Peter

console.log(friends.length); // 3

console.log(friends[friends.length - 1]); // Peter

friends[2] = "Jay";

console.log(friends); // ['Michael', 'Steven', 'Jay']

const firstName = "Jonas";

const jonas = [firstName, "Schmedtmann", 2037 - 1991, "teacher", friends]; console.log(jonas); //

['Jonas', 'Schmedtmann', 46, 'teacher', ['Michael', 'Steven', 'Jay']] console.log(jonas.length);

// 5

// Exercise
const years = [1990, 1967, 2002, 2010, 2018];

console.log(years[0]);
console.log(years[1]);
console.log(years[years.length - 1]);
```

4.2. Basic Array Operations (Methods)

```
const friends = ["Michael", "Steven", "Peter"];

// Add elements

const newLength = friends.push("Jay");
```

```
console.log(friends); // ['Michael', 'Steven', 'Peter', 'Jay']

console.log(newLength); // 4

friends.unshift("John");

console.log(friends); // ['John', 'Michael', 'Steven', 'Peter', 'Jay'] //

Remove elements

friends.pop(); // Last

console.log(friends); // ['John', 'Michael', 'Steven', 'Peter']

const popped = friends.pop();

console.log(popped); // Peter

console.log(friends); // ['John', 'Michael', 'Steven']

friends.shift(); // First

console.log(friends); // ['Michael', 'Steven']

console.log(friends.indexOf("Steven")); // 1

console.log(friends.indexOf("Bob")); // -1

friends.push(23);

console.log(friends.includes("Steven")); // true

console.log(friends.includes("Bob")); // false

console.log(friends.includes(23)); // true

if (friends.includes("Steven")) {
 console.log("You have a friend called Steven"); // You have a friend called Steven }
```

4.3. Introduction to Objects

```
const jonasArray = [
 "Jonas",
 "Schmedtmann",
 2037 - 1991,
 "teacher",
 ["Michael", "Peter", "Steven"],
];

const jonas = {
 firstName: "Jonas",
 lastName: "Schmedtmann",
 age: 2037 - 1991,
 job: "teacher",
 friends: ["Michael", "Peter", "Steven"],
};
```

4.4. Dot vs. Bracket Notation

```
const jonas = {
 firstName: "Jonas",
 lastName: "Schmedtmann",
 age: 2037 - 1991,
 job: "teacher",
 friends: ["Michael", "Peter", "Steven"],
};

console.log(jonas); // {firstName: "Jonas", lastName: "Schmedtmann", age: 46, job: "teacher", friends: Array(3)}

console.log(jonas.lastName); // Schmedtmann

console.log(jonas["lastName"]); // Schmedtmann

const nameKey = "Name";

console.log(jonas["first" + nameKey]); // Jonas

console.log(jonas["last" + nameKey]); // Schmedtmann

// console.log(jonas.'last' + nameKey); // Uncaught SyntaxError: Unexpected string
```

4.5. Object Methods

```
const jonas = {
 firstName: "Jonas",
 lastName: "Schmedtmann",
 birthYeah: 1991,
 job: "teacher",
 friends: ["Michael", "Peter", "Steven"],
 hasDriversLicense: true,
};

console.log(jonas);
```

4.6. Iteration: The for Loop

```
for (let rep = 1; rep <= 10; rep++) {
 console.log(`Lifting weights repetition ${rep}`);
}

// Lifting weights repetition 1

// ...

// Lifting weights repetition 10
```

```
const jonas = [
 "Jonas",
 "Schmedtmann",
 2037 - 1991,
 "teacher",
 ["Michael", "Peter", "Steven"],
 true,
];

const types = [];

for (let i = 0; i < jonas.length; i++) {
 console.log(jonas[i], typeof jonas[i]);

 // Jonas string
 // Schmedtmann string
 // 46 number
 // teacher string
 // (3) ["Michael", "Peter", "Steven"] object
 // true boolean

 // types[i] = typeof jonas[i];

 types.push(typeof jonas[i]);
}

console.log(types); // (6) ["string", "string", "number", "string", "object", "boolean"]

const years = [1991, 2007, 1969, 2020];

const ages = [];

for (let i = 0; i < years.length; i++) {
 ages.push(2037 - years[i]);
}

console.log(ages); // (4) [46, 30, 68, 17]

// continue and break

console.log("--- ONLY STRINGS ---");

for (let i = 0; i < jonas.length; i++) {
 if (typeof jonas[i] !== "string") continue;
 console.log(jonas[i], typeof jonas[i]);
}

console.log("--- BREAK WITH NUMBER ---");

for (let i = 0; i < jonas.length; i++) {
 if (typeof jonas[i] === "number") break;
 console.log(jonas[i], typeof jonas[i]);
}
```

4.7. Looping Arrays, Breaking and Continuing

```
const jonas = [
 "Jonas",
 "Schmedtmann",
 2037 - 1991,
 "teacher",
 ["Michael", "Peter", "Steven"],
];

for (let i = jonas.length - 1; i >= 0; i--) {
 console.log(i, jonas[i]);
}

// 4 (5) ["Michael", "Peter", "Steven"]

// 3 teacher

// 2 46

// 1 Schmedtmann

// 0 Jonas

for (let exercise = 1; exercise < 4; exercise++) {
 console.log(`-------- Starting exercise ${exercise}`);

 for (let rep = 1; rep < 6; rep++) {
 console.log(`Exercise ${exercise}: Lifting weight repetition ${rep}`);  }
}
```

4.8. The while Loop

```
for (let rep = 1; rep <= 10; rep++) {
 console.log(`Lifting weights repetition ${rep}`); }

let rep = 1;

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

let dice = Math.trunc(Math.random() * 6) + 1;

console.log(dice); // 4

while (dice !== 6) {
 console.log(`You rolled a ${dice}`);
 dice = Math.trunc(Math.random() * 6) + 1;
 if (dice === 6) console.log("Loop is about to end..."); }
```

4.9 Functions

```
// function declaration
function logger() {
 console.log("My name is Jonas"); }

// calling / running / invoking function

logger(); // My name is Jonas


function sum(a, b) {
 return a + b;
}

const result = sum(1, 2);

console.log(result); // 3

// function declaration
const logger = function () {
 console.log("My name is Jonas"); };

// calling / running / invoking function

logger(); // My name is Jonas


const sum = function (a, b) {
 return a + b;
};

const result = sum(1, 2);
console.log(result); // 3

// arrow function
const logger = () => {
 console.log("My name is Jonas"); };

// calling / running / invoking function

logger(); // My name is Jonas


const sum = (a, b) => {
 return a + b;
};

const result = sum(1, 2);
console.log(result); // 3

const sum = (a, b) => a + b;
const result = sum(1, 2);
console.log(result); // 3
```

4.10. Function Declarations vs. Expressions

```
// function declaration
function calcAge1(birthYeah) {
 return 2037 - birthYeah;
}

const age1 = calcAge1(1991);

// function expression

const calcAge2 = function (birthYeah) {
 return 2037 - birthYeah;
};

const age2 = calcAge2(1991);

console.log(age1, age2); // 46 46
```

4.11. Callback Functions

```
function greet(name, callback) {
 console.log("Hello, " + name + "!"); // Hello, Alice!
 callback();
}

function sayBye() {
 console.log("Goodbye!"); // Goodbye!
}

greet("Alice", sayBye);

// Hello, Alice!
// Goodbye!
```

4.12. Higher-Order Array Methods

map

```
const years = [10, 20, 30, 40, 50];

// ES6

const modifiedYears = years.map((year) => {
 return year + 1;
});

console.log(modifiedYears); // (5) [11, 21, 31, 41, 51]

// const modifiedYears = years.map((year) => year + 1);
// console.log(modifiedYears); // (5) [11, 21, 31, 41, 51]

const modifiedYears = years.map((year, index) => {
 return `Year ${index + 1}: ${year + 1}`;
});

console.log(modifiedYears); // (5) ["Year 1: 11", "Year 2: 21", "Year 3: 31", "Year 4: 41", "Year 5: 51"]

const modifiedYears = years.map((year, index) => {
 return `Year ${index + 1}: ${year + 1}`;
});

console.log(modifiedYears); // (5) ["Year 1: 11", "Year 2: 21", "Year 3: 31", "Year 4: 41", "Year 5: 51"]
```

forEach

```
const years = [10, 20, 30, 40, 50];

years.forEach((year) => {
 console.log(year + 1);
});
// 11 21 31 41 51

years.forEach((year, index) => {
 console.log(`Year ${index + 1}: ${year + 1}`);
});

// Year 1: 11
// Year 2: 21
```

filter

```
const years = [10, 20, 30, 40, 50];

const modifiedYears = years.filter((year) => {
 return year > 30;
});

console.log(modifiedYears); // (2) [40, 50]
```

reduce

```
const years = [10, 20, 30, 40, 50];

const modifiedYears = years.reduce((acc, cur) => {
 return acc + cur;
}, 0);

console.log(modifiedYears); // 150
```

4.13. OOP in JavaScript

```
const jonas = {
 firstName: "Jonas",
 lastName: "Schmedtmann",
 birthYear: 1991,
 job: "teacher",
 friends: ["Michael", "Peter", "Steven"],
 hasDriversLicense: true,

 // calcAge: function (birthYear) {
 // return 2037 - birthYear;
 // },

 // calcAge: function () {
 // // console.log(this);
 // return 2037 - this.birthYear;
 // },

 calcAge: function () {
 this.age = 2037 - this.birthYear;
 return this.age;
 },

 getSummary: function () {
 return `${this.firstName} is a ${this.calcAge()}-year old ${
 this.job
 }, and he has ${this.hasDriversLicense ? "a" : "no"} driver's license.`;  },
};

console.log(jonas.calcAge()); // 46

console.log(jonas.age); // 46

console.log(jonas.getSummary()); // Jonas is a 46-year old teacher, and he has a driver's license.
```