📅 Extracting Date and Time Information in JavaScript

JavaScript provides the **Date object**, which allows us to work with **dates and times**. The methods described in the text help extract specific **pieces of date and time**.

1⃣ Creating a Date Object

Before extracting specific date components, we first need to create a **Date object**:

```
var d = new Date();  // Creates a new Date object representing the current date and time
```

2⃣ Extracting Specific Date Components

◆ Getting the Month (`getMonth()`)

- Returns the **month number**, but JavaScript starts counting from **0**.
- **January = 0**, **February = 1**, ..., **December = 11**.
- Example:
- var currentMonth = d.getMonth();
- console.log(currentMonth); // Output: 0 (if it's January)

💡 **Tip:** To display the actual month name, use an **array**:

```
var monthNames = ["January", "February", "March", "April", "May", "June",
                  "July", "August", "September", "October", "November", "December"];

var monthName = monthNames[d.getMonth()];
console.log(monthName);  // Output: "January" (if it's January)
```

◆ Getting the Day of the Month (`getDate()`)

- Returns the **day of the month** (from `1` to `31`).
- Example:
- var dayOfMonth = d.getDate();
- console.log(dayOfMonth); // Output: 16 (if today is the 16th)

◆ Getting the Year (`getFullYear()`)

- Returns the **4-digit year**.
- Example:
- var currYr = d.getFullYear();
- console.log(currYr); // Output: 2025

3️⃣ Extracting Time Components

◆ Getting the Hours (`getHours()`)

- Returns the **hour** in **24-hour format** (from `0` to `23`).
- Example:
- var currentHrs = d.getHours();
- console.log(currentHrs); // Output: 14 (if it's 2:00 PM)

💡 **Tip:** If you want **12-hour format**, use this:

```
var hours12Format = currentHrs % 12 || 12;
console.log(hours12Format); // Converts 14 to 2 (PM)
```

◆ Getting the Minutes (`getMinutes()`)

- Returns the **minutes** (from `0` to `59`).
- Example:
- var currMins = d.getMinutes();
- console.log(currMins); // Output: 45 (if the time is 14:45)

◆ Getting the Seconds (`getSeconds()`)

- Returns the **seconds** (from `0` to `59`).
- Example:
- var currSecs = d.getSeconds();
- console.log(currSecs); // Output: 30 (if the time is 14:45:30)

◆ Getting the Milliseconds (`getMilliseconds()`)

- Returns **milliseconds** (from `0` to `999`).
- Example:
- var currMills = d.getMilliseconds();
- console.log(currMills); // Output: 523

4️⃣ Getting the Timestamp (`getTime()`)

- Returns the **number of milliseconds** that have passed since **January 1, 1970 (Unix Epoch Time)**.
- Example:
- var millsSince = d.getTime();

- console.log(millsSince); // Output: 1708202345231 (depends on current time)

💡 Why Use getTime()?

- Useful for **calculating time differences**.
- Example: **How many days since January 1, 2000?**
- var pastDate = new Date("2000-01-01");
- var today = new Date();
-
- var diffMills = today.getTime() - pastDate.getTime();
- var diffDays = diffMills / (1000 * 60 * 60 * 24);
-
- console.log(Math.floor(diffDays) + " days since January 1, 2000");

📌 Summary of Date Methods

Method	Description	Example Output
getMonth()	Returns the **month number** (0-11)	0 (January)
getDate()	Returns the **day of the month** (1-31)	16
getFullYear()	Returns the **4-digit year**	2025
getHours()	Returns the **hour (24-hour format)** (0-23)	14 (2 PM)
getMinutes()	Returns the **minutes** (0-59)	45
getSeconds()	Returns the **seconds** (0-59)	30
getMilliseconds()	Returns the **milliseconds** (0-999)	523
getTime()	Returns **milliseconds since Jan 1, 1970**	1708202345231

🎯 Conclusion

- The **Date object** allows us to extract **month, day, year, hours, minutes, seconds, and milliseconds**.
- **Months start from 0 (January) to 11 (December).**
- **Hours are in 24-hour format (use % 12 for 12-hour format).**
- **Milliseconds since 1970-01-01 help in calculations.**
- **Use arrays to get proper month and day names.**