Lesson 02.04

The Date Object

- The **Date Object** is instantiated using the **new** keyword: **new Date()**.
- The Date Object returns the full date-time from the user's computer.
- The return value is saved as an **instance** (variable) of the Date Object.
- The instance is an object with all the Date Object's methods and properties.

1. Instantiate the Date Object, saving the returned object as **dateTime**:

```
let dateTime = new Date();
console.log(dateTime); // Mon Mar 27 2023 11:08:04 GMT-0400 (Eastern
Daylight Time)
```

The individual time units are accessible via the Date object's "get methods".

2. Get the current hour, minute and second:

```
let hour = dateTime.getHours();
console.log(hour); // 0-23

let minute = dateTime.getMinutes();
console.log(minute); // 0-59

let second = dateTime.getSeconds();
console.log(second); // 0-59
```

3. Express the time in 00:00:00 format:

```
let timeIs = `${hour}:${minute}:${second}`;
console.log(timeIs); // 11:8:4
```

adding leading zeroes to minute and second

If minute and/or second are less than 10, the time says 11:8:4, not 11:08:04. To fix this, add leading zeros to minute and second, as needed. This is done with conditional logic.

4. I minute or second are less than 10, set the variable equal to itself with a '0' concatenated in front:

```
if(minute < 10) {
  minute = '0' + minute;
}
```

```
if(second < 10) {
  second = '0' + second;
}

timeIs = `${hour}:${minute}:${second}`;
console.log(timeIs); // 11:08:04
```

Whenever there is only one line of code inside the curly braces of an "if statement", you can delete the curly braces, and put it all onto the same line.

5. Comment out the above "if statements" and re-write them as one-liners:

```
if(minute < 10) minute = '0' + minute;
if(second < 10) second = '0' + second;

timeIs = `${hour}:${minute}:${second}`;

console.log(timeIs); // 11:08:04
```

converting military time to AM/PM time

The Date Object returns an hour from 0-23, which is in military time:

- Instead of 3:00 pm, it's 1500 hours
- Instead of 10:00 pm it's 2200 hours

To convert to AM/PM time, we need:

- a variable to store the string "AM" or "PM". We can give it a default value of "AM".
- follow the variable declaration with two if-statements, in this exact order:
    - if hour > 11, set the variable to "PM" ("12:01 PM" is one minute after "12 noon")
    - if hour > 12, subtract 12 from hour (1900 hours becomes "7:00 PM");

6. Declare a variable **amOrPm** with an initial value of 'AM', and follow that with the if-statements:

```
  let amOrPm = 'AM';

  if(hour > 11) {
      amOrPm = 'PM';
  }

  if(hour > 12) {
      hour -= 12;
  }

  // OR:
  // if(hour > 11) amOrPm = 'PM';
  // if(hour > 12) hour -= 12;
```

```
  timeIs = `${hour}:${minute}:${second} ${amOrPm}`;
  console.log('time is: ', timeIs);
```

Making a "Timely Greeting"

Let's make a "timely greeting", one which is appropriate for the time of day.

- **if** the hour is less than 12, say "Good morning!"
- **else if** the hour is less than 18 (6:00pm), say "Good afternoon!"
- **else**, say "Good Evening!"

7. Our original hour may be old and / or may have had 12 subtracted from it, so get a fresh instance of the Date Object, as well as a fresh time units. We'll use shortened variable names this time:

```
let dt = new Date();
let h = dt.getHours();
let m = dt.getMinutes();
let s = dt.getSeconds();
```

8. Declare greeting in the global scope as an empty string. It's value will be set by the "if else" logic.

```
let greeting = "";
```

9. If the current hour is less than 12 (noon), set the greeting to "Good Morning!":

```
if(h < 12) {
    greeting = "Good Morning!";
}
```

10. If the current hour is NOT less than 12, but it IS less than 18 (6:00 PM), set the greeting to "Good Afternoon!":

```
if (h < 12) {
  greeting = "Good Morning!";
} else if (h < 18) {
  greeting = "Good Afternoon!";
}
```

11. If both of these conditions are false, then the the current hour must be in the 18-23 range, that is, 6:00 PM or later. So set the greeting to "Good Afternoon!":

```
if (h < 12) {
  greeting = "Good Morning!";
} else if (h < 18) {
  greeting = "Good Afternoon!";
} else {
  greeting = "Good Evening!";
}

console.log(greeting);
```

"Timely Greeting"

Let's display the greeting, along with the time in AM-PM format. This requires us to recalculate the hour, since we have a new variable for that:

12. Declare a new var for am-pm so that we can convert military time to "regular" time:

```
let amPM = "AM";
```

13. If the current hour is greater than 11, set amPM to "PM":

```
if(h > 11) amPm = 'PM';
```

14. If the current hour is greater than 12, subtract 12 from the hour:

```
if(h > 12) h -= 12;
```

15. Add a leading "0" to minute or second, if needed:

```
if(m < 10) m = '0' + m;
if(s< 10) s = '0' + s;
```

16. Concatenate a fresh time display

```
timeIs = `${h}:${m}:${s} ${amPm}`;
```

17. Log the timely greeting along with the time:

```
let timelyGreeting = `${greeting} The time is: ${timeIs}`;
console.log(timelyGreeting);
```

Outputting the "timely greeting" to the web page

18. Get the html element that we have for displaying the timely greeting:

```
let greetingTag = document.getElementById('greeting');
```

19. Set the **textContent** property of the tag object to **timelyGreeting**:

```
greetingTag.textContent = timelyGreeting;
```

Today's Date

Let's concatenate today's date from other Date Object time units.

20. Get today's date, which is an integer from 0-31:

```
let date = dt.getDate();
console.log('date', date);
```

21. Get the month, which is a number from 0-11, where Jan is 0:

```
let month = dt.getMonth();
console.log('month', month);
```

22. Get the month as a string (January, February, etc). This gives us the flexibility to use the month as either a number or a day:

```
let fullMonth = dateTime.toLocaleString('default', {month:'long'})
console.log('fullMonth', fullMonth);
```

23. Get the day of the week as a number, with Sunday being 0

```
let day = dt.getDay();
console.log('day', day);
```

24. Make an array of the days of the week.

```

```

```
let daysOfTheWeek = ["Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"];
```

Recall that an array is a variable which stores comma-separated values inside square brackets. Also recall that we can look up an item in an array by: **array[index]**.

25. Look up the first item in the array (Sunday) by its index (0):

```
let firstDay = daysOfTheWeek[0];
console.log('first day of week:', firstDay); // first day of week: Sunday
```

26. Get the day of the week by looking it up in the array by its index (0-6), which we have as day:

```
let dayOfWeek = daysOfTheWeek[day];
console.log('day of week:', dayOfWeek); // day of week: Monday
```

27. Get the four-digit year:

```
let fullYear = dt.getFullYear();
console.log(fullYear);
```

28. Concatenate today's dates:

```
let todaysDate = `${dayOfWeek}, ${fullMonth} ${date}, ${fullYear}`;
console.log(todaysDate); // Monday, March 27, 2023
```

29. Output today's date to the web page:

```
let todaysDateTag = document.getElementById('todays-date');
todaysDateTag.textContent = todaysDate;
```