Strings

Continuum of pure consciousness

String

- A String in JavaScript is a sequence of characters. String literals can be created in JavaScript by wrapping a sequence of characters inside a set of single quotes (' ') or a double quotes (" ")

```
const age = 9;
const name = "John";
const intro = "Hi I am "+name+" , I am "+age+ " years old";
console.log(intro);
```

- ES6 also added template literals that allows embedded expressions. You can use multi-line string and string interpolation feature with template literals (` `)

```
const intro = `Hi I am ${name}, I am ${age} years old.`
```

Escape Sequence

- There are special characters in JavaScript with special meanings (', ", `, \ and more). When we want to use these characters literally, we may need to escape them by placing a backslash (\) immediately before it.
- There are also some pre-defined escape sequences in JavaScript (\n is for new line).
  - https://www.w3schools.com/js/js_strings.asp

Accessing a Character

- Single character from a string can be access by using array like syntax.

```
const str = "Hello";
console.log(str[2]);
console.log(str[0]);
```

- The `chartAt()` method also does the same thing

```
const str = "Hello";
console.log(str.charAt(2));
console.log(str.charAt(0));
```

String methods

- JavaScript includes several useful string methods

```
const str = "Hello";
console.log(str.toUpperCase()); // HELLO
console.log(str.toLowerCase()); // hello
console.log(str.startsWith("H")); // true
console.log(str.substr(1,3)); //ell
```

- https://www.w3schools.com/jsref/jsref_obj_string.asp

Exercise

- Write a program that keeps on asking for user input and prints it, until user types the word "stop" (without quotes). "Stop" word can be in any case (small, capital or mixed)
- Write a program that takes comma separated text from user and converts it into an array of words and prints in reverse order.
- Write a program to replace all the occurrence of "for" in an input string with 4.

input	output
Thanks for joining us.	Thanks 4 joining us.

Objects

- Object is a complex data structure, that is composed of other data structures.
  - It may be composed of simple data structures like variables of type Boolean, Number or String
  - It may also include other complex data structures like arrays, functions or even other objects.
- Better suited for modeling real world entities, objects.

```
// Spaces and breaks are not important
let student={
    name: "John",
    id: 123,
    course_completed: ['CS301', 'CS303'],
    is_accelerated_track: true,
}
```

- Simplest way to create an object in JavaScript is using the object literal {}, a set of curly braces.

Object properties

- The *name:value* pairs in JavaScript objects are called *propertyes*:

Property	Value
name	“John”
id	123
course_completed	[‘CS301’, ‘CS303’]
is_accelerated_track	true

Accessing and updating object properties

- In JavaScript, you can access object properties in two ways:
  - **`objectName.propertyName`**
  - `objectName[propertyName]`

```
let name = student.name
console.log(name); // John

let first_course = student.course_completed[0];
console.log(first_course); // CS301

student.name = "Jack";
student.is_accelerated_track = false;
console.log(student); //{name: 'Jack,...,is_accelerated_track:false}
```

- See example – *lecture_codes/lesson7/objects.js*

Object methods

- When an object has a function as its property, we call the property as a method of the object (terminology)
- Method are the actions that can be performed on the object

```
let student={
    name: "John",
    id: 123,
    course_completed: ['CS301', 'CS303'],
    is_accelerated_track: true,
    getCourseCount: function(){return this.course_completed.length}
}
```

- In a function definition, `this` refers to the "owner" of the function.
  - When called as an object’s method, the owner of the function is the object itself.

Accessing object methods

- You can access object methods with the following syntax:
  - `objectName.methodName();`
  - E.g. `courseCount = student.getCourseCount();`

- If you access a method without the () parentheses, it will return the function definition:
  - E.g. `functionCourseCount = student.getCourseCount;`

- See example – *lecture_codes/lesson7/objects_with_methods.js*

Assignments

- Readings
  - chapter 12
  - https://www.w3schools.com/js/js_objects.asp
- Chapter 12, programming assignments (all)
- Create a car object with following properties and methods:
  - Properties: `make`, `model`, `year`, `odometer`, `serviceCount`
  - Methods:
    - `service()` which increments the serviceCount value by one every time this method is called.
    - `getOdometerReading()`, this method simply returns the current value of the odometer property.
    - `reset()`, this method resets the value of `odometer` and `serviceCount` both to 0.