

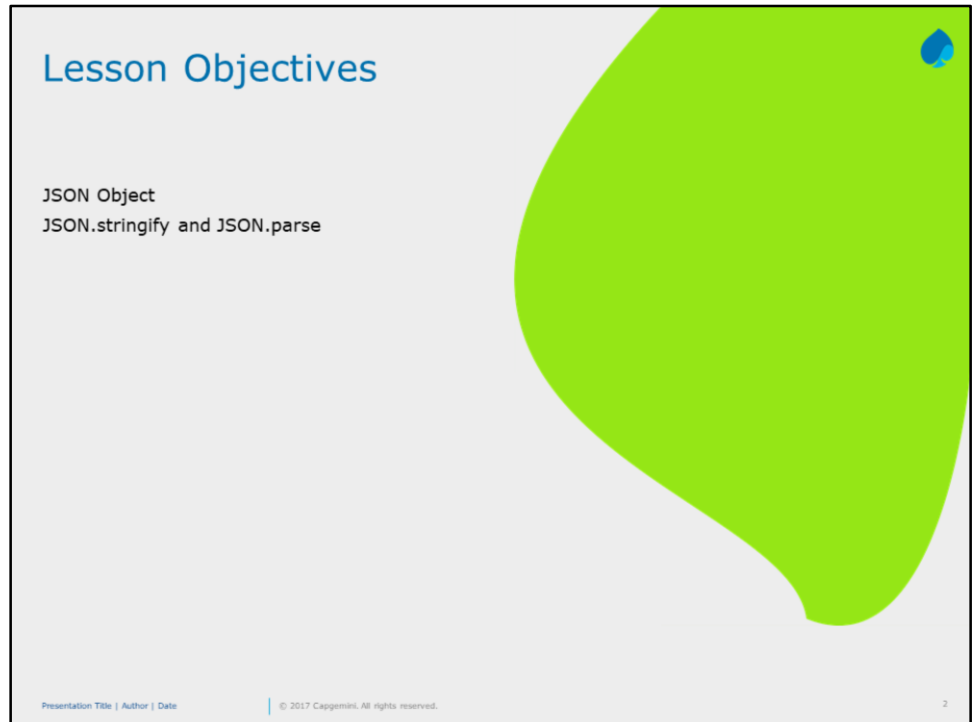
**Instructor Notes:**

Add instructor notes here.



## Instructor Notes:

Add instructor notes here.



Following contents would be covered:

1.1 : What are Web services

1.1.1 Web service components and architecture

1.1.2 How do Web services work

1.2: HTTP and SOAP messages

1.3: Overview of JAX – WS and JAX – RS

**Instructor Notes:**

## 3.1. JSON Introduction

**JSON Introduction**

JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax.

It is commonly used for transmitting data in web applications (e.g., sending some data from the server to the client, so it can be displayed on a web page, or vice versa).

A JSON object can be stored in its own file, which is basically just a text file with an extension of .json, and a MIME type of application/json.

JSON is purely a data format — it contains only properties, no methods.

JSON requires double quotes to be used around strings and property names. Single quotes are not valid.

Ecma International is an industry association founded in 1961 and dedicated to the standardization of Information and Communication Technology (ICT) and Consumer Electronics

**ECMAScript** is the scripting language standardized by Ecma International in the ECMA-262 specification and ISO/IEC 16262. The language is widely used for client-side scripting on the web, in the form of several well-known implementations such as JavaScript, JScript and ActionScript.

**Terminology**

**Namespace** A container which lets developers bundle all functionality under a unique, application-specific name.  
**Class** Defines the object's characteristics. A class is a template definition of an object's properties and methods.  
**Object** An instance of a class.  
**Property** An object characteristic, such as color.  
**Method** An object capability, such as walk. It is a subroutine or function associated with a class.  
**Constructor** A method called at the moment an object is instantiated. It usually has the same name as the class containing it.  
**Inheritance** A class can inherit characteristics from another class.  
**Encapsulation** A method of bundling the data and methods that use the data.  
**Abstraction** The conjunction of an object's complex inheritance, methods, and properties must adequately reflect a reality model.  
**Polymorphism** Poly means "*many*" and morphism means "*forms*". Different classes might define the same method or property.

**Instructor Notes:**

## 3.1. JSON Introduction

**JSON Introduction**

Even a single misplaced comma or colon can cause a JSON file to go wrong, and not work.

We can validate JSON using an application like JSONLint.

JSON can actually take the form of any data type that is valid for inclusion inside JSON, not just arrays or objects. So for example, a single string or number would be a valid JSON object.

Unlike in JavaScript code in which object properties may be unquoted, in JSON, only quoted strings may be used as properties.

**Instructor Notes:**

3.1. JSON Introduction

**JSON Type**

Number : integer, real or floating point

String : double-quoted Unicode with backslashes

Boolean : true and false

Array : ordered sequence of comma-separated values enclosed in square brackets

Object : collection of comma-separated "key":value pairs enclosed in curly braces

null

**Instructor Notes:**

3.2. Working with JSON Object

**JSON Object Notation**

A JSON object is an unordered set of name/value pairs

- A JSON object begins with { (left brace) and ends with } (right brace)
- Each name is followed by : (colon) and the name/value pairs are separated by , (comma) and enclosed with in quotes.

The JSON.parse function deserializes JSON text to produce a JavaScript value.

```
var data = {"Name":"Abcd", "age":55}
```

```
var dataparsed = eval(data);
```

```
console.log(dataparsed.Name);  
console.log(dataparsed.age);
```

**Instructor Notes:**

3.2. Working with JSON Object

**JSON Object Notation**

The JSON.stringify function serializes a JavaScript value to JSON text.

```
function Employee(name, age, salary) {  
    this.Name = name;  
    this.age = age;  
    this.salary = salary;  
}  
  
var employeeObject = new Employee('Abcd',25,5118);  
  
console.log(employeeObject);
```

## Instructor Notes:

Add instructor notes here.

Demo

Demo1  
Demo2



Add the notes here.



## Instructor Notes:

Add instructor notes here.

Lab

Lab 3



Add the notes here.

**Instructor Notes:**

Add instructor notes here.

### Summary

In this lesson we have learned about –

JSON Object  
JSON.stringify and JSON.parse



Summary