MAHARISHI UNIVERSITY OF MANAGEMENT

*Higher Consciousness and Professional Excellence*

# Object Oriented and Functional Programming in JavaScript

CS303

April 2021

Ralph Bunker, Ph.D.

2021

Maharishi's 14th Year of Invincibility – Global Raam Raj

**CS303**

# Object Oriented and Functional Programming in JavaScript

**April 2021**

## Course Overview

| | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday |
|---|---|---|---|---|---|---|
| | | | | | | |
| **Week 1** | Functions and Arrays Review | Object Properties and References | Object Methods and Constructor Functions | Data Types: Strings and Arrays | Data Types: Map Filter Reduce | Review |
| | | | | | | |
| **Week 2** | Recursion, Rest and Spread | Closures and lexical environments | Scheduled call backs | Call Context, function binding | Javascript and Unicode | Review |
| | | | | | | |
| **Week 3** | Properties | Prototypal inheritance, F.protype | Native prototypes, prototype methods | Classes basic syntax, inheritance | [Static, private, protected] properties and methods, | Review |
| | | | | | | |
| **Week 4** | Encapsulation and Polymorphism | Error Handling | Review | Exam 3 | 15 minute interviews | |

# CS303
# Object Oriented and Functional Programming in JavaScript

## Course Goals

In this course, you will gain proficiency in using JavaScript as a functional programming language. You will learn object-oriented programming principles and the nature of JavaScript's loosely typed programs and dynamic object model.

## Course Objectives

Understand and write JavaScript programs for the following:

- programs composed of functions and data types appropriate for a different application types;
- advanced function concepts such as recursion, closures, higher order functions, callbacks and call forwarding;
- JavaScripts lexical environment and scoping;
- revealing module pattern and encapsulation of implementation details;
- prototypal inheritance model and constructor functions;
- JavaScript's new class syntax;
- professional coding principles and practices for high quality code including coding conventions, linting processors, JS doc, and unit testing.

**How You Will Accomplish the Objectives**
Preview the slides and reading, attend lectures, complete readings and answer quiz questions.

**How You Will Demonstrate You Have Accomplished the Objectives**
Complete homework labs and project and write small applications on exams.

## Course Texts

Kantor, I. JavaScript.info (https://javascript.info) The Modern JavaScript Tutorial, 2021. (main text)

Haverbeke, M. Eloquent JavaScript, 3rd Ed., (https://eloquentjavascript.net/), No Starch Press, San Francisco, 2009. (supplement)

## Evaluation Criteria

| | |
|---|---|
| Labs | 40% |
| Final exam | 50% |
| Class participation | 10% |

| | |
|---|---|
| a+ | 97 - 100 |
| a | 92 - 96 |
| a- | 88 - 91 |
| b+ | 84 - 87 |
| b | 79 - 83 |
| b- | 75 - 78 |
| c+ | 71 - 74 |
| c | 66-70 |
| c- | 62-65 |

## Class Attendance

Attendance at all classes is required, because all elements of class — lectures, questions and answers, discussions, laboratory work — contribute to the learning process. Absences are usually excused only if you are sick in bed or have a family emergency.

If you must miss a class, please let your instructor know ahead of time. Call, send an email, or send a note with a friend. There is no such thing as a "personal day." If you have personal business to take care of, please schedule it for after class or during the days between blocks. At the same time, it may occasionally be necessary for you to miss a class (or part of a class) for some reason other than illness or family emergency. Please speak with the instructor beforehand, who will be open to considering your needs.

The first lesson of each course is the most important. Students are expected to be present from the first lesson onward. Any student not present on the first morning (except for such compelling reasons as illness or family emergency) may be asked to withdraw from the course. Unexcused absences may result in the student receiving a grade of NC (No Credit) for the whole course.

**Object Oriented and Functional Programming in JavaScript**

## **Reading assignments** from https://javascript.info

Always pre-read the assignments the day before the lesson. This will help you follow the lecture. After the lecture you should be able to briefly review any parts of the reading that were unclear and quickly proceed with the lab assignment for the day.

### Week 1

**Monday**
JavaScript Fundamentals Chapter, https://javascript.info/first-steps
Functions: https://javascript.info/function-basics
Function expressions: https://javascript.info/function-expressions
Arrow functions the basics: https://javascript.info/arrow-functions-basics
JavaScript specials: https://javascript.info/javascript-specials
Data types chapter: https://javascript.info/data-types

- Arrays: https://javascript.info/array

**Tuesday**
Objects: The basics chapter https://javascript.info/object-basics
Objects: https://javascript.info/object
Object references and copying: https://javascript.info/object-copy
Garbage collection: https://javascript.info/garbage-collection

**Wednesday:**
Objects: The basics chapter https://javascript.info/object-basics
Object methods, this: https://javascript.info/object-methods
Constructor operator, new: https://javascript.info/constructor-new
Optional chaining: https://javascript.info/optional-chaining

**Thursday:**
Data types: https://javascript.info/data-types

- Methods of primitives: https://javascript.info/primitives-methods
- Numbers: https://javascript.info/number
- Strings: https://javascript.info/string
- Arrays: https://javascript.info/array

**Friday:**
Data types: https://javascript.info/data-types
Map and Set: https://javascript.info/map-set
Object keys, values, entries: https://javascript.info/keys-values-entries
Date and Time: https://javascript.info/date
JSON methods, toJSON: https://javascript.info/json

**Saturday**
Review

**Object Oriented and Functional Programming in JavaScript**

**Week 2**

**Monday:**
Advanced working with functions: https://javascript.info/advanced-functions
- Recursion and stack: https://javascript.info/recursion
- Rest parameters and the spread operator: https://javascript.info/rest-parameters-spread

**Tuesday:**
Advanced working with functions: https://javascript.info/advanced-functions
- Variable scope, closure: https://javascript.info/closure
- The "old" var: https://javascript.info/var
- Global object: https://javascript.info/global-object
- Function object: https://javascript.info/function-object

**Wednesday:**
Advanced working with functions: https://javascript.info/advanced-functions
- Scheduling: setTimeout and setInterval: https://javascript.info/settimeout-setinterval

**Thursday:**
Advanced working with functions: https://javascript.info/advanced-functions
- Decorators and forwarding, call/apply: https://javascript.info/call-apply-decorators
- Function binding: https://javascript.info/bind
- Arrow functions revisited: https://javascript.info/arrow-functions

**Friday:**
Data types: https://javascript.info/data-types
- Unicode: https://javascript.info/string#internals-unicode

**Saturday:**
- Review

**WEEK 3**

**Monday:**
Object properties configuration: https://javascript.info/object-properties
- Property flags and descriptors: https://javascript.info/property-descriptors
- Property getters and setters: https://javascript.info/property-accessors

**Tuesday:**
Review Objects: the basics: https://javascript.info/object-basics
- Constructor, operator "new": https://javascript.info/constructor-new

Protypes, inheritance: https://javascript.info/prototypes
- Protypal inheritance: https://javascript.info/prototype-inheritance
- F.prototype: https://javascript.info/function-prototype

**Wednesday:**

**Object Oriented and Functional Programming in JavaScript**

Protypes, inheritance: https://javascript.info/prototypes

Native prototypes: https://javascript.info/native-prototypes

Prototype methods, objects without __proto__: https://javascript.info/prototype-methods

**Thursday:**

Classes: https://javascript.info/classes

Class basic syntax: https://javascript.info/class

Class inheritance: https://javascript.info/class-inheritance

**Friday:**

Classes: https://javascript.info/classes

Static properties and methods: https://javascript.info/static-properties-methods

Private and protected properties and methods: https://javascript.info/private-protected-properties-methods

**Saturday:**

Review

**WEEK 4**

**Monday:**

Classes: https://javascript.info/classes

Encapsulation: https://javascript.info/private-protected-properties-methods#summary

Polymorphism: https://javascript.info/function-object#the-length-property

**Tuesday:**

Error handling: https://javascript.info/error-handling

Error handling, “try…catch”: https://javascript.info/custom-errors

**Wednesday:**

Review

**Thursday:**

Exam

**Friday:**

One-on-one interviews

# Lesson 1
# Review of JavaScript Functions and Arrays
## The Whole Is Greater than the Sum of the Parts

**MAIN POINTS**

1. Functions are the fundamental components that do all the processing in JavaScript programs. Functions should be short and self-contained. Science of Consciousness: Just like clear self-contained functions promote successful programs, clear self-sufficient awareness promotes successful action in life.

2. JavaScript functions are executed if they end with parentheses. They can also be treated as values, parameters, and return values.

3. Arrays are fundamental data structures for ordered collections, where we have a 1st, a 2nd, a 3rd element and so on.

**CONNECTING THE PARTS OF KNOWLEDGE**
**WITH THE WHOLENESS OF KNOWLEDGE**
The Whole Is Greater than the Sum of the Parts

1. Key components of JavaScript programs are functions and arrays.

2. Functions can be values as well as executable instructions.

-----------------------------------------------------------------------------------------------------------

3. **Transcendental consciousness** is the experience of the wholeness that is our own awareness and the field in which all other experiences live.
4. **Impulses within the transcendental field:** Thoughts from this level are naturally integrated and coherent and infused with the wholeness of pure awareness.
5. **Wholeness moving within itself:** In unity consciousness we appreciate the value of wholeness in all the component of daily life.

# Lessons 2 & 3
# Objects: Knowledge is the Wholeness of Knower, Known, and Process of Knowing

(Almost) everything in JavaScript is an object. Objects encapsulate state information in properties and associated behavior in methods. Science of Consciousness: Objects are an example of the common model-view-controller pattern of computer science, which closely corresponds to knower-known-process of knowing in our own consciousness. The state information contained in the properties is a model. The methods define the controller. The property names are the interface or view.

## MAIN POINTS

1. Objects and properties can be created very simply using the object literal syntax. Properties can be added after the object is created simply by assigning a value to a property. Science of Consciousness: This is an example of accomplishing something in a very simple manner. Natural law takes the path of least action.

2. Variables bound to primitive types contain actual values. Variables bound to objects contain references to memory locations that store the objects. Science of Consciousness: Variables are identifiers. They are not the actual values. When we transcend thought during the practice of our TM Technique we experience deeper values of our Self beyond our surface identity.

3. The behavior associated with an object is defined in methods, which are properties that have functions as their value. Methods refer to other object properties using the keyword 'this'. Since functions are first-class they can be passed between objects, in which case 'this' can refer to different objects at different times.

4. Constructor functions are helpful when we need to create many similar objects

## CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

Knowledge Is the Wholeness of Knower, Known, and Process of Knowing

1. Objects contain properties and their values.

2. Large JavaScript programs use objects and their properties to break the program into smaller manageable components. It is the interaction of all the component objects that make the program work.

---------------------------------------------------------------------------------------------

4. **Transcendental consciousness**. State of wholeness.

**4. Impulses within the transcendental field:** Pure consciousness, by its own nature, is aware of itself by it. It thereby becomes the knower of itself through the self-referral process of knowing.

**5. Wholeness moving within itself:** In unity consciousness the value of wholeness is appreciated even in the components of knower, known, and process of knowing.

## Lessons 4 & 5
## Data Types: Knowledge Has Organizing Power

All programs are organized in terms of data (information) and the operations that can be performed on it. Science of Consciousness: Programs are organized around different types of data and operations common to those types. Knowledge Has Organizing Power.

### MAIN POINTS

1. Numbers, strings, and booleans each have special operations such as parseInt, parseFloat, round, includes, and slice.

2. Arrays are used in almost every program. There are special methods for common operations on them including to modify, search, and transform arrays.

### CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

Knowledge Has Organizing Power

1. Numbers, strings, and arrays are important data types that have many common operations unique to their purpose and many methods in the language to support those operations.
2. JavaScript arrays are highly flexible data structures with many built in methods.

-----------------------------------------------------------------------------------------------

3. **Transcendental consciousness.** Is the experience of total knowledge and perfect orderliness.
4. **Impulses within the transcendental field:** Thoughts connected to the field of all the laws of nature will be supported by that level of total knowledge and coherence.
5. **Wholeness moving within itself:** In unity consciousness one appreciates daily perceptions and experiences as being infused with order and purpose.

# Lesson 6
# Destructuring, Dates, JSON: The Nature of Life Is to Grow

JavaScript is a rapidly evolving language. This lesson covers several new data type features that make the language more efficient and powerful. *Science of Consciousness:* The nature of life is to grow and evolve to greater accomplishment and fulfillment.

## MAIN POINTS

1. *Destructuring assignment* is a new (ES6) and now widely used convenience syntax that allows us to "unpack" arrays or objects into a bunch of variables. *Science of Consciousness:* This feature allows developers to more quickly accomplish common coding tasks. Do less and accomplish more.

2. JavaScript has added a new Date object has many convenient methods for working with subtle aspects of dates and time. *Science of Consciousness:* Dates are important in many common programming tasks and involve complex logic. This is an important development for the language. The nature of life is to grow and evolve.

3. JSON is a widely used data exchange format used in every modern programming language. It is based on JavaScript object representations. The main uses are to convert JavaScripts to strings for transmission over a network and to parse JSON strings sent from servers into JavaScript objects. *Science of Consciousness:* JSON is a mechanism for improving communication between remote entities—clients and servers. Coherent and orderly consciousness is a mechanism for improving communication between people.

## CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

The Nature of Life Is to Grow

1. Destructuring assignments, the Date object, and JSON are new features and data representations in JavaScript.
2. Destructuring provides syntactic convenience and programming efficiency, Date provides capabilities, and JSON enables remote transfer of data.

---------------------------------------------------------------------------------------------------

3. **Transcendental consciousness.** Is the simplest state of awareness and home of all the laws of nature.
4. **Impulses within the transcendental field:** Thoughts at this level will be maximally life supporting.
5. **Wholeness moving within itself:** In unity consciousness one experiences the goal of all growth and evolution in daily life.

**Object Oriented and Functional Programming in JavaScript**

# Lessons 7 & 8
# Recursion: Self Referral Awareness

## MAIN POINTS

1. Recursion is when a function calls itself. Recursion is useful when a task can be simplified into an easy action plus a simpler variant of the same task. Science of Consciousness: Repetition, looping, is the basis of computing. All repetition in computing can be implemented through recursion. Recursion is an example of self referral. Self referral awareness is the basis of manifest existence.

2. Rest parameters and the spread operator are convenience syntax for, respectively, collecting function parameters into an array or assigning collection elements across a set of variables.

## CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

The Nature of Life Is to Grow

1. Recursive functions call themselves on successively smaller parts of a program.
2. In theory all of computing can be represented by recursion. In practice, recursive functions work best for problems involving recursive data structures such as trees and linked lists.

-----------------------------------------------------------------------------------------------

3. **Transcendental consciousness.** Is the experience of the Self awake to itself.
4. **Impulses within the transcendental field:** Awareness of Awareness is the most basic process of knowledge and produces a three in one structure of knower, known, and process of knowing.
5. **Wholeness moving within itself:** In unity one perceives the basic reality of Self referral awareness as the source of all elements, matter, and existence.

# Lessons 9, 10, 11
# Closures: Established in Being, Perform Action

Wholeness: A fundamental aspect of function-oriented programming in JavaScript is the use of closures to store state information associated with a function when the function is passed to other objects. Science of Consciousness: Closures provide a protective wrapper for state information associated with a function. An analogy in consciousness is the supportive wrapper that transcendental consciousness provides to our own consciousness.

## MAIN POINTS

1. Closures are created whenever an inner function with free variables is returned or assigned as a callback. Closures provide encapsulation of methods and data. Encapsulation promotes self-sufficiency, stability, and re-usability. **Science of Consciousness: C**losures provide a supportive wrapper for actions that will occur in another context. Transcendental consciousness provides a supportive wrapper for our actions that will occur outside of meditation.

2. Functions are objects and can have their own properties. Function properties can store state information associated with a function like a closure, however, function properties are accessible to external objects unlike closure variables.

## CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

Established in Being, Perform Action

1. Closures are a feature of functional programming languages that allow state information to be encapsulated with functions when they are passed among objects.
2. The scope of variables in modern JavaScript is defined by the lexical environment.

---------------------------------------------------------------------------------------------

3. **Transcendental consciousness** is the home of all the laws of nature.
4. **Impulses within the transcendental field:** Thoughts encapsulated by this deep level of consciousness will result in actions in accord with all the laws of nature.
5. **Wholeness moving within itself:** In unity consciousness all thoughts and perceptions are encapsulated by blissful wrapper of pure consciousness.

**Object Oriented and Functional Programming in JavaScript**

# Lessons 12 & 13
# Prototype Inheritance: Archetypal Patterns of Intelligence

Wholeness: Inheritance is a fundamental feature of object-oriented programming. Common code is kept in a base component. Specialized components 'inherit' the common code from the more general base component. Science of Consciousness: An archetype is a fundamental pattern or law of nature that gives rise to many variations and realizations at more expressed levels of nature. Deeper levels of awareness make us more connected with these fundamental patterns.

## MAIN POINTS

1. Prototypal inheritance allows object to inherit properties from a 'prototype' parent object. The main purpose of inheritance is to promote code reuse and avoid duplication. Science of Consciousness: Reuse of code for common tasks is efficient and avoids errors that can arise from inconsistent updates of duplicated code. Natural law takes the path of least action. Do less and accomplish more.

2. Programmers cannot directly access the special [[Prototype]] property. All functions have a regular 'prototype' property. When they are called as constructors with 'new' that property will be set as the value of [[Prototype]]. [[Prototype]] can also be set with the __proto__ property, but that is now deprecated in favor of Object.create. Science of Consciousness: JavaScript's prototype is like "archetype", which is an original object that is a basis for other objects. Deeper levels of thought are connected to archetypal patterns of intelligence or 'laws of nature'.

## CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

Archetypal Patterns of Intelligence

1. JavaScript objects often share common methods through prototype chains.
2. Modern JavaScript sets up prototype chains using the prototype property of constructor functions and the Object.create method.

-----------------------------------------------------------------------------------------------

3. **Transcendental consciousness** Is the experience of pure consciousness, the level of awareness that is the basis of all existence and all patterns of intelligence.
4. **Impulses within the transcendental field:** Thoughts arising from this level have direct access to the deepest patterns of intelligence of nature.
5. **Wholeness moving within itself:** In unity consciousness all levels of existence are perceived as expressions of these archetypal patterns of intelligence.

# Lessons 14 & 15
# Classes: Evolution through Layers of Abstraction

Wholeness: JavaScript classes are a new syntax that simplifies and enforces proper use of function constructors and prototypal inheritance for creating objects and using good object-oriented design principles such as encapsulation. Science of Consciousness: Classes are an evolution of JavaScript as a language to support the requirement for good object-oriented design principles needed to develop large applications found in modern web applications. The nature of life is to grow and evolve through layers of abstraction.

## MAIN POINTS

1. JavaScript classes are mainly a helpful syntax over JavaScript constructor functions. Science of Consciousness: JavaScript classes are an instance of abstracting away implementation details and issues of constructor functions with this improved syntax. Life is found in layers and more abstract layers of consciousness are simpler and more effective.

2. JavaScript classes add keywords extend and super that are abstractions over the details of inheritance with function constructors and prototype chains.

3. Encapsulation is a cornerstone principle of good object-oriented design. Encapsulation hides internal component implementation details and provides a well delimited external interface for external components. Science of Consciousness: Experience of the simplest state of awareness eliminates stress and allows us to only have thoughts that are relevant to a given situation. This is like providing a well delimited external interface appropriate to the external environment.

## CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

Evolution through Layers of Abstraction

1. JavaScript classes are a helpful syntax that abstracts out details of constructor functions and prototypal inheritance for creating objects.
2. The extends and super keywords cause objects and properties to be set in the function prototype and [[Prototype]] properties.

------------------------------------------------------------------------------------------------

3. **Transcendental consciousness** Is the simplest state of awareness. It abstracts away everything and is also the basis of everything.
4. **Impulses within the transcendental field:** Impulses at this level are the finest layer of existence and represent the first abstraction of knower, known, and process of knowing when consciousness is aware of itself.
5. **Wholeness moving within itself:** In unity consciousness one appreciates all layers of existence as expressions and abstractions over pure consciousness, the unified field of existence.