

# **JavaScript: Objects, BOM, and DOM**

---

**CS 4640**  
**Programming Languages  
for Web Applications**

Objects group variables and functions to create a model representing something you would recognize from the real world

## Object type: Hotel

Event	Happens when
Reserve	reservation is made
Cancel	reservation is cancelled

Events are things or interactions that can happen to the objects

### Properties

Name:	Awesome
Rating:	5
Rooms:	70
Bookings:	56
Pool:	true
Gym:	true

Method	What it does
makeReservation()	increases value of <i>bookings</i> property
cancelReservation()	decreases value of <i>bookings</i> property
checkAvailability()	subtracts value of <i>bookings</i> property from value of <i>rooms</i> property and returns number of rooms available

Properties tell us the characteristics of the objects

Methods represent tasks that are associated with the objects (or things we can do with the objects)

## Car

Accelerate	driver speeds up	changeSpeed()
------------	------------------	---------------

Method	What it does
changeSpeed()	increases or decreases value of <i>currentSpeed</i> property

### Properties

Make:	UVA I
currentSpeed:	30
Color:	yellow
Fuel:	gasoline

# JavaScript Objects

- JavaScript is an **object-based** language
  - It supports for object-oriented programming but not at the same level as other languages (ES6: introduced `class` – still lacks private property)
- Objects are represented as **property-value** pair
  - The property values can be data or functions (methods)
- A property is something that can be modified :
  - **Data properties** : primitive values or references to objects
  - **Method properties** : can be executed
- Objects can be created and their properties can be **changed dynamically**
  - JS is not really typed .. If it doesn't care between a number and a string, why care between two kinds of objects?

# Creating Objects

Create an object and assign variables and functions directly by using `{ }` syntax

```
var hotel = {  
  name: "Awesome",  
  rating: 5,  
  rooms: 70,  
  bookings: 56,  
  pool: true.
```

Global frame

hotel

object

Name	"Awesome"
Rating	5
Rooms	70
Bookings	56
Pool	true
Gym	true
checkAvailability	<pre>function () {   return this.rooms - this.bookings; }</pre>

```
n() {  
s.bookings;
```

# Creating Objects with Constructors

revisit

Global frame

Hotel

hotel1

hotel2

```
function Hotel(name, rating, rooms, bookings, pool, gym) {  
  this.name = name;  
  this.rating = rating;  
  this.rooms = rooms;  
  this.bookings = bookings;  
  this.pool = pool;  
  this.gym = gym;  
  this.checkAvailability = function() {  
    return this.rooms - this.bookings;  
  };  
};
```

object

name	"Awesome"
rating	5
rooms	70
bookings	56
pool	true
gym	true
checkAvailability	function () { return this.rooms - this.bookings; }

object

name	"Duh"
rating	3
rooms	45
bookings	27
pool	false
gym	false
checkAvailability	function () { return this.rooms - this.bookings; }

ing the constructor function and

rooms, bookings, pool, gym) {

unction() {  
.bookings;

e', 5, 70, 56, true, true);  
3, 45, 27, false, false);

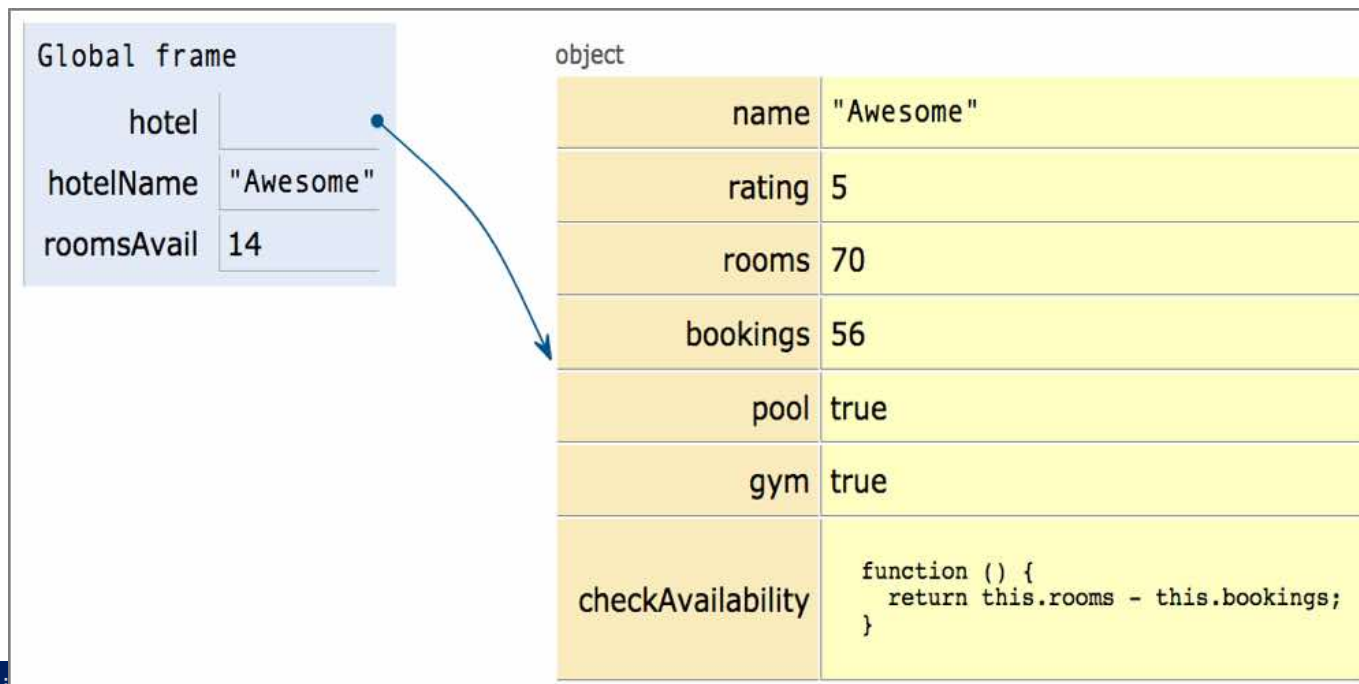
# Accessing Objects

- Access properties or methods of an object using dot notation

```
var hotelName = hotel.name;  
var roomsAvail = hotel.checkAvailability();
```

- Access properties or methods using square brackets

```
var hotelName = hotel['name'];  
var roomsAvail = hotel['checkAvailability']();
```



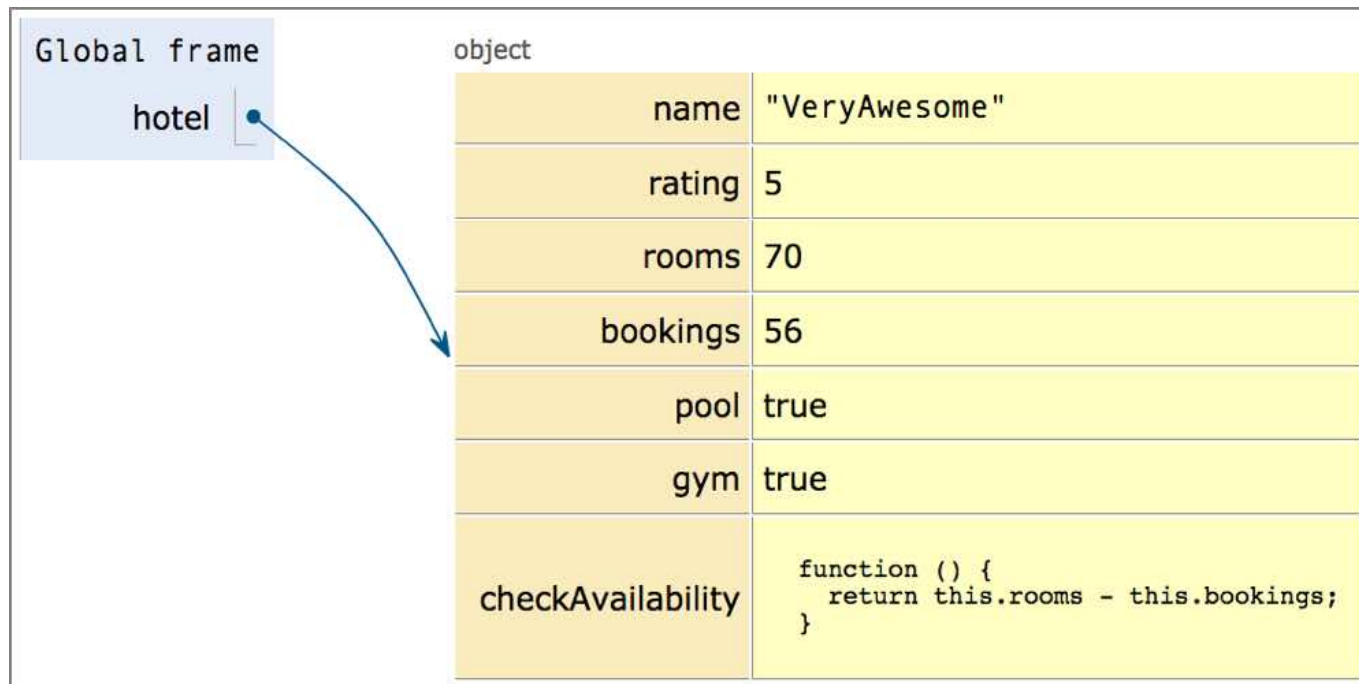
# Updating Properties

- Update properties using dot notation

```
hotel.name = 'VeryAwesome';
```

- Update properties using square brackets

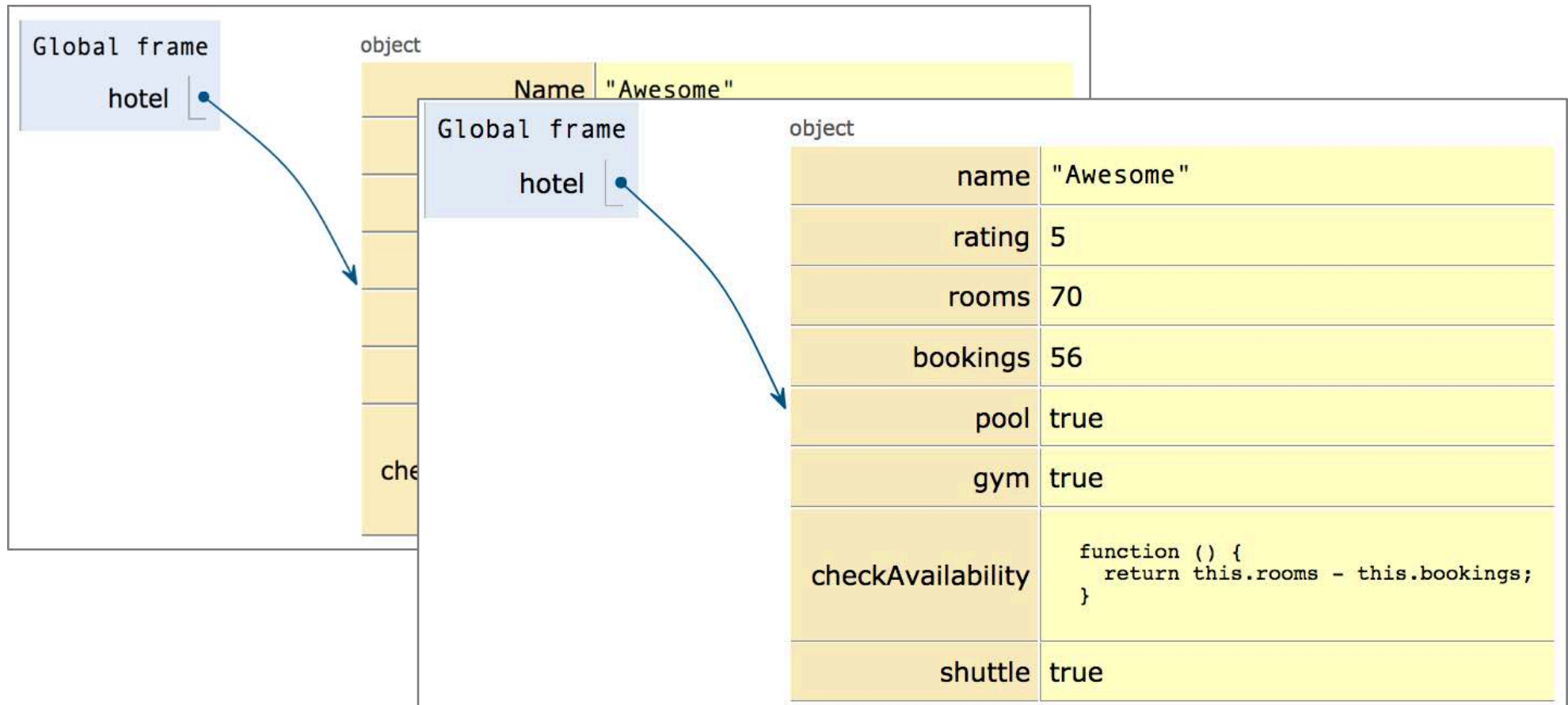
```
hotel['name'] = 'VeryAwesome';
```



# Adding Properties

- Add a property using a dot notation

```
hotel.shuttle = true;
```

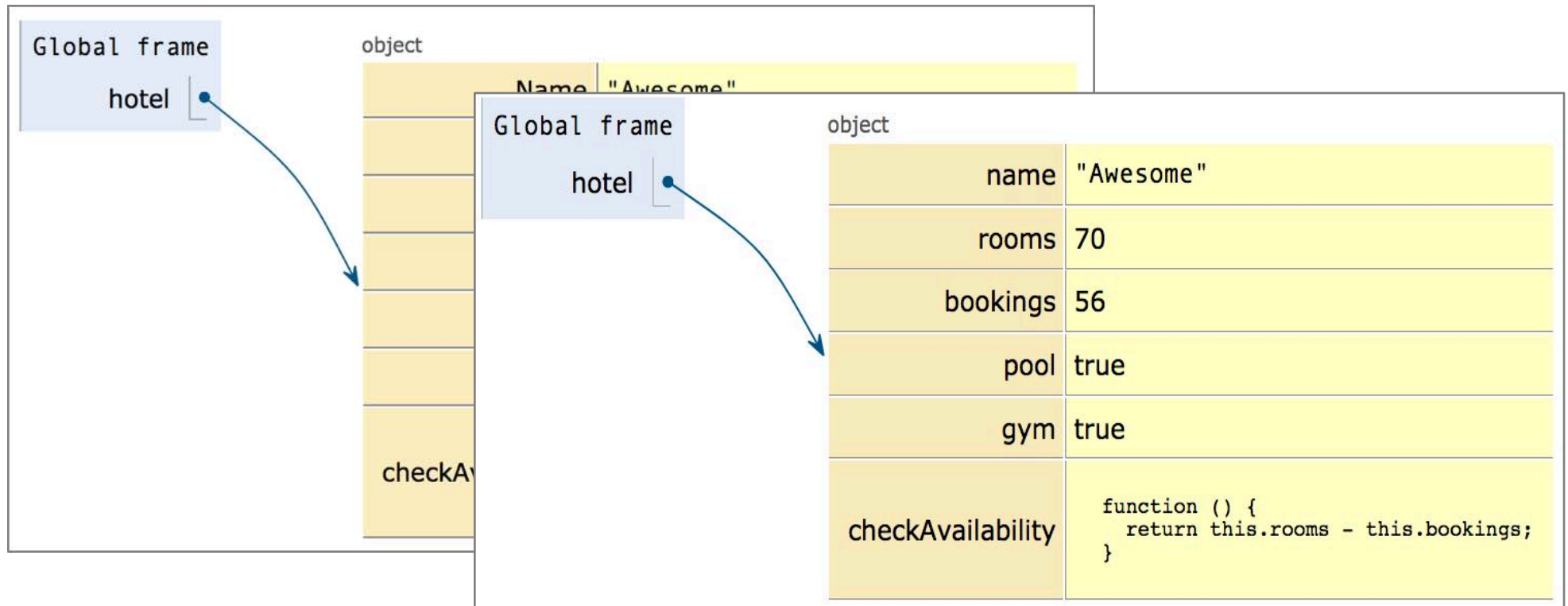




# Deleting Properties

- Delete a property using the **delete** keyword

**delete** hotel.rating;



# Web Browsers and Objects

revisit

**Object type: Window**

**Properties**  
Location: <http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html>

CS 4640 - WebPL

← → ↻ 🏠 [www.cs.virginia.edu/~up3f/cs4640/syllabus.html](http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html) 90% ... ☆

CS 4640: Web PL Syllabus Schedule Quizzes Policies

*Note: This is an initial course description and may be updated without prior notice.*

## Spring 2019 — Syllabus

*Do users look at web apps the way they are? Or do users look at web apps the way they think?*

- Contact
- Class hours
- Overview
- Learning objectives
- Prerequisite
- Reading
- Learning activities
- Assessment
- Discussion board
- Research
- Academic integrity
- SDAC and other special circumstances
- Religious accommodations
- Statement on violence

### Contact information

**Instructor:** Upsorn Praphamontripong  
**Office:** Rice Hall 206  
**Office hours:** TBD (until they are settled, whenever my door is open!)

**TAs:** TBD

### Class hours

Tuesday, Thursday 2:00PM - 3:15PM, Clark Hall

### Overview

The way web software is built has been rapidly changing. With the rise of e-commerce, gathering to direct customer sales (e-commerce) is a new business model. Many new technologies and frameworks have emerged. What programming languages should you use? What programming styles you should use? What programming languages are in great demand? How will you keep up with web development? How will companies prioritize the foundation of web development?

This course will help you see how fundamental concepts of web development can be applied to develop reliable and usable web software regardless of the technologies or languages or frameworks. Although we put our emphasis on the concepts, you will develop dynamic web software with several commonly used programming languages and technologies.

You will work on user interface design, the front end development, back end development, and web-based information retrieval and processing. Over the semester, you will work on each programming assignment with a partner. Each assignment will be a step in creating a web application. You will be able to choose what to build, with the assignment constraining the features that must be used. By the end of the semester, you will have built a dynamic web application. With a large portion of the process that involves team work, interpersonal skills and conflict management.

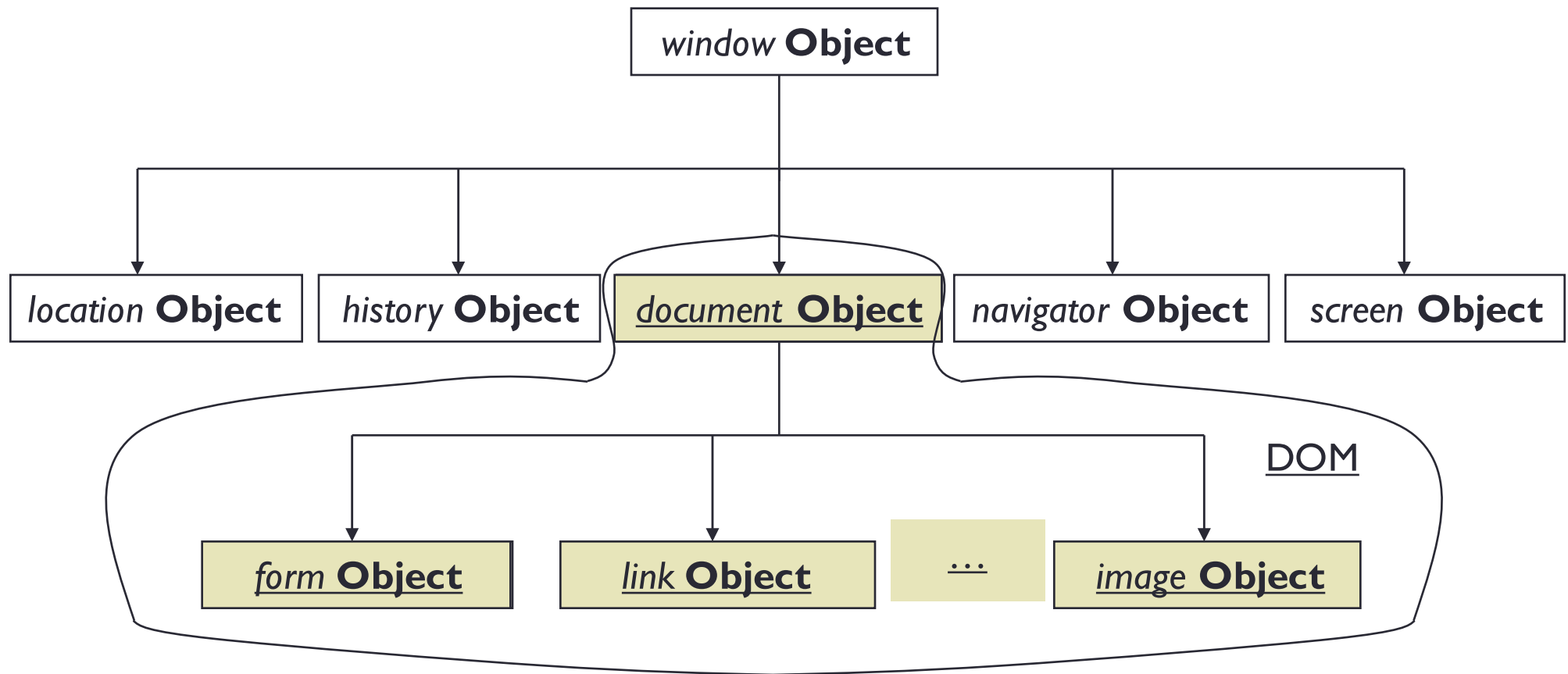
**Object type: Document**

**Properties**  
URL: <http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html>  
lastModified: 01/14/2019 10:19:23  
Title: CS4640 - WebPL

# BOM: Browser Object Model

revisit

- **BOM** – collection of objects that the browser makes available to us for use with JavaScript



# DOM: Document Object Model

CS 4640: Web PL

*Note: This is an initial course description and may be updated without prior notice.*

## Spring 2019 — Syllabus

*Do users look at web apps the way they are? Or do users look at web apps the way they think?*

**Contact information**

**Instructor:** Upsorn Praphamontripont  
Office: Rice Hall 206  
Office hours: TBD (until they are settled)

**TAs:** TBD

**Class hours**

Tuesday, Thursday 2:00PM - 3:15PM

**Overview**

The way web software is built has been gathering to direct customer sales (e-commerce). Many new technologies and frameworks styles you should use? What programming in great demand? How will you keep up companies prioritize the foundation of v

This course will help you see how fundamental regardless of the technologies or languages software with several commonly used p

You will work on user interface design, the front end development, back end development, and web-based information retrieval and processing. Over the semester, you will work on each programming assignment with a partner. Each assignment will be a step in creating a web application. You will be able to choose what to build, with the assignment constraining the features that must be used. By the end of the semester, you will have built a dynamic web application. With a large portion of the process that involves team work, interpersonal skills and conflict management.

Object type: Document	
<b>Properties</b>	
URL:	<a href="http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html">http://www.cs.virginia.edu/~up3f/cs4640/syllabus.html</a>
lastModified:	01/14/2019 10:19:23
Title:	CS4640 - WebPL
<b>Event</b>	<b>Happens when</b>
Load	page and content have finished loading
Click	user clicks the mouse over the page
Keypress	user presses down on a key
<b>Method</b>	<b>What it does</b>
write()	adds content to the document
getElementById()	accesses an element of a given <i>id</i> attribute

# How A Browser See A Web Page

CS4640 - WebPL

www.cs.virginia.edu/~up3f/cs4640/syllabus.html

CS 4640: Web PL

Syllabus Schedule Quizzes Policies

*Note: This is an initial course description and may be updated without prior notice.*

## Spring 2019 — Syllabus

Contact  
Class hours  
Overview  
Learning objectives  
Prerequisite  
Reading  
Learning activities  
Assessment  
Discussion board  
Research  
Academic integrity  
SDAC and other special circumstances  
Religious accommodations  
Statement on violence

### Contact information

**Instructor:** Upsorn Praphamontripong  
Office: Rice Hall 206  
Office hours: TBD (until they are settled, whenever my door is open!)

**TAs:** TBD

### Class hours

Tuesday, Thursday 2:00PM - 3:15PM, Clark Hall 108

### Overview

The way web software is built has been rapidly changing. As use of the world wide web has shifted from information presentation to information gathering to direct customer sales (e-commerce) to enterprise applications, the amount and complexity of software has steadily been increasing. Many new technologies and frameworks have emerged everyday. Have you ever wondered what technologies, frameworks, or architectural styles you should use? What programming languages you should be familiar with? After you graduate, what languages will be popular or will be in great demand? How will you keep up with web development technologies? According to the UVA CS Industrial Advisory Board meeting, companies prioritize the foundation of web development over any specific technologies or frameworks.

This course will help you see how fundamental concepts of web development can be applied to develop reliable and usable web software regardless of the technologies or languages or frameworks. Although we put our emphasis on the concepts, you will develop dynamic web software with several commonly used programming languages and technologies.

You will work on user interface design, the front end development, back end development, and web-based information retrieval and processing. Over the semester, you will work on each programming assignment with a partner. Each assignment will be a step in creating a web application. You will be able to choose what to build, with the assignment constraining the features that must be used. By the end of the semester, you will have built a dynamic web application. With a large portion of the process that involves team work, interpersonal skills and conflict management.

The browser receives an HTML page

It creates a model of the page and stores it in memory

container  
bottom ...

It shows the page on screen using a rendering engine

fact  
nation



# Using BOM Objects (Some Properties)

Property	Description
<code>window.screenX</code>	X-coordinate of pointer, relative to top left corner of screen (in pixels)
<code>window.screenY</code>	Y-coordinate of pointer, relative to top left corner of screen (in pixels)
<code>window.location</code>	Current URL of window object
<code>window.document</code>	Reference to document object
<code>window.history</code>	Reference to history object for browser window or tab, which contains details of the pages that have been viewed in that window or tab
<code>window.history.length</code>	Number of items in history object
<code>window.screen</code>	Reference to screen object
<code>window.screen.width</code>	Accesses width property of screen object
<code>window.screen.height</code>	Accesses height property of screen object

# Using BOM Objects (Some Methods)

Method	Description
<code>window.alert()</code>	Create modal dialog box with message (user must click OK button to close it)
<code>window.open(url)</code>	Open new browser window with the specified URL
<code>window.print()</code>	Tell browser that user wants to print contents of current page (act like user has clicked a print option)
<code>window.history.back()</code>	Move backward through history
<code>window.history.forward()</code>	Move forward through history
<code>window.history.go(step)</code>	Move to specific page from session history (step specifies the number of pages, forward or backward)
<code>history.pushState(state, title, url)</code>	Create a new entry (or add a URL) at the top of the browser history
<code>history.replaceState(state, title, url)</code>	Modify the current entry (current URL at the top) of the browser history

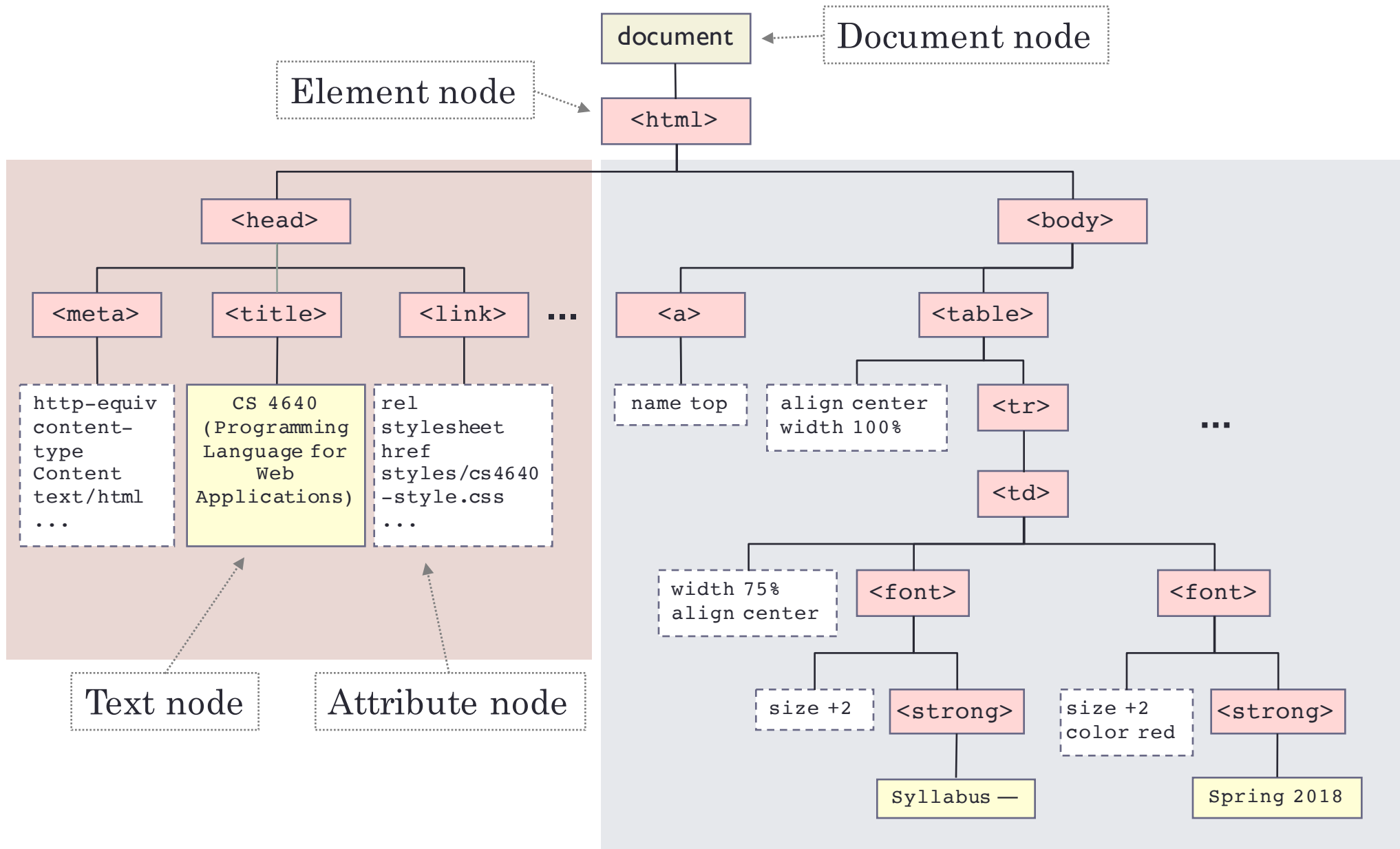
# Using DOM Objects

---

- Not part of HTML or JS
- Separate rules implemented by all major browser markers
- Two primary purposes:
  - Making a **model** of the HTML page
    - Specifies how browsers should create a model of an HTML page
  - **Accessing** and **changing** the HTML page
    - Specifies how JS can access and update the contents of a web page



# DOM: Four Types of Nodes



# Using DOM Objects

## (Some Properties and Methods)

Property	Description
<code>document.title</code>	Title of current document
<code>document.lastModified</code>	Date on which document was last modified
<code>document.URL</code>	String containing URL of current document
<code>document.domain</code>	Domain of current document

Method	Description
<code>document.write( )</code>	Write text to document
<code>document.getElementById(id)</code>	Return element whose id attribute matches the specified id
<code>document.querySelectorAll(selector)</code>	Return list of elements that match the specified CSS selector
<code>document.createElement(element)</code>	Create new element
<code>document.createTextNode(text)</code>	Create new text node