IN4MATX 285:
Interactive Technology Studio

Practice: Compatibility and Cross-Platform

Today’s goals

By the end of today, you should be able to…

- Explain why compatibility testing is important in web development
- Search for compatibility challenges in implementation
- Describe advantages and disadvantages of developing native and web applications

So I’m setting up for 285, and I get this message…

So I get this message…

Anne Marie Piper Tue, Mar 18, 6:22 AM

to me, Matthew

Hi Daniel,

This looks great!! I want to take your class. :-)

I shared some comments via Slack. Just FYI, I couldn't access the assignment details on my phone but could see them on my laptop.

What’s going on???

Calendar

Mar 30	Mar 31	Apr 1	Apr 2	Apr 3	Apr 4	Apr 5
	Overview: What is This Course? Slides **Programming: HTML** Slides **Live Demo: HTML** 4:00-5:00 | Zoom Demo Code	**Ziqi Office Hours** 5:00-6:00 | Zoom	**Practice: Version Control** Slides **Daniel Office Hours** 4:00-5:00 | Zoom		**Weijun Office Hours** 4:00-5:00 | Zoom	
Apr 6	Apr 7	Apr 8	Apr 9	Apr 10	Apr 11	Apr 12
A1 Due Design Mockup	**Programming: CSS & Design Systems** Slides **Live Demo: CSS & Design Systems** 4:00-5:00 | Zoom Demo Code	**Ziqi Office Hours** 5:00-6:00 | Zoom	**Practice: Frontend vs. Backend Development** Slides **Daniel Office Hours** 4:00-5:00 | Zoom		**Weijun Office Hours** 4:00-5:00 | Zoom	
Apr 13	Apr 14	Apr 15	Apr 16	Apr 17	Apr 18	Apr 19
A1 Due Code Check-in	**Programming: Variables, Loops, & Conditionals in JS** Slides **Live Demo: Variables, Loops, Conditionals** 4:00-5:00 | Zoom	**Ziqi Office Hours** 5:00-6:00 | Zoom	**Practice: Responsive and Adaptive Design** **Daniel Office Hours** 4:00-5:00 | Zoom		**Weijun Office Hours** 4:00-5:00 | Zoom	

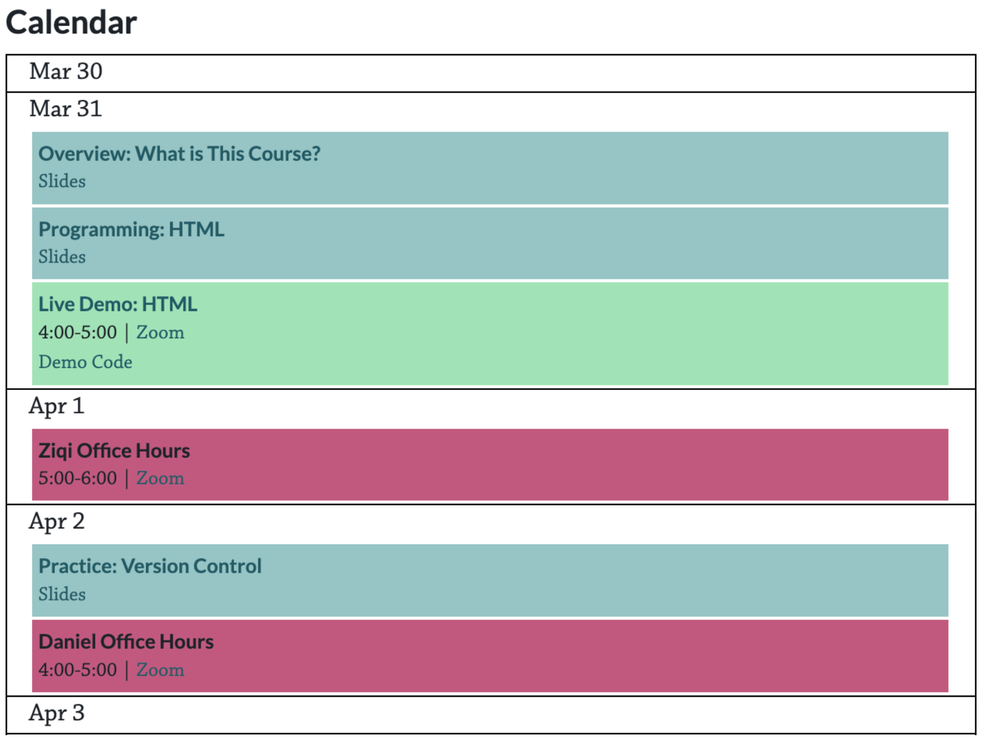
Desktop, in Chrome

Calendar

iPhone, in Safari

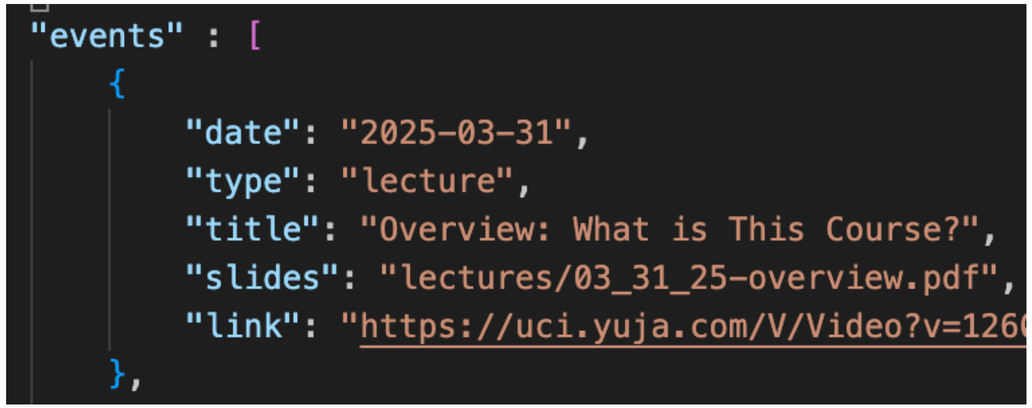
What’s going on???

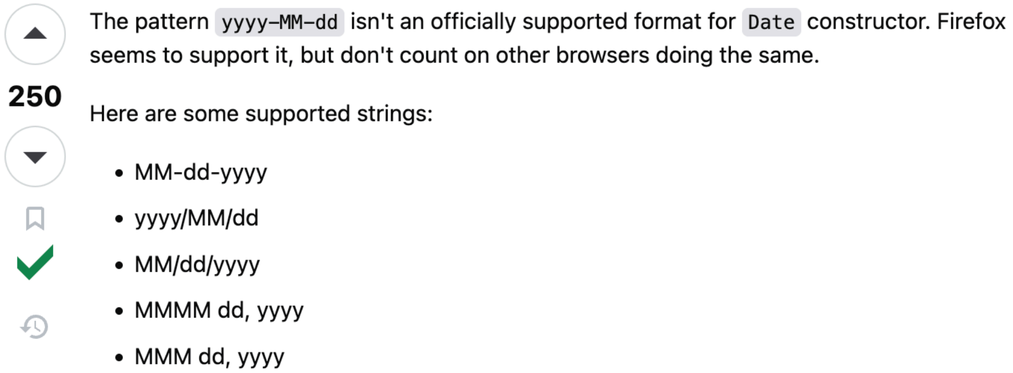
- At first, I thought this was a problem with the responsive design.
- But when I change the screen size, the calendar renders as I expected it to.



What’s going on???

- Turns out, Chrome and Safari handle Dates differently, and my calendar was silently failing.





https://stackoverflow.com/questions/4310953/invalid-date-in-safari

Wait, JavaScript is different in different browsers?!?

Yes, yes it is.

History of JavaScript

- *“Developed under the name Mocha, the language was officially called LiveScript when it first shipped in beta releases of Netscape Navigator 2.0 in September 1995, but it later was renamed JavaScript”*

- Java’s popularity was on the rise
  - Marketing ploy
  - Intended to be the “web” language to Java’s “desktop”

https://medium.com/@benastontweet/lesson-1a-the-history-of-javascript-8c1ce3bffb17

History of JavaScript

- Netscape submitted JavaScript to ECMA International for consideration as an industry standard

- Subsequent versions were standardized as “ECMAScript”

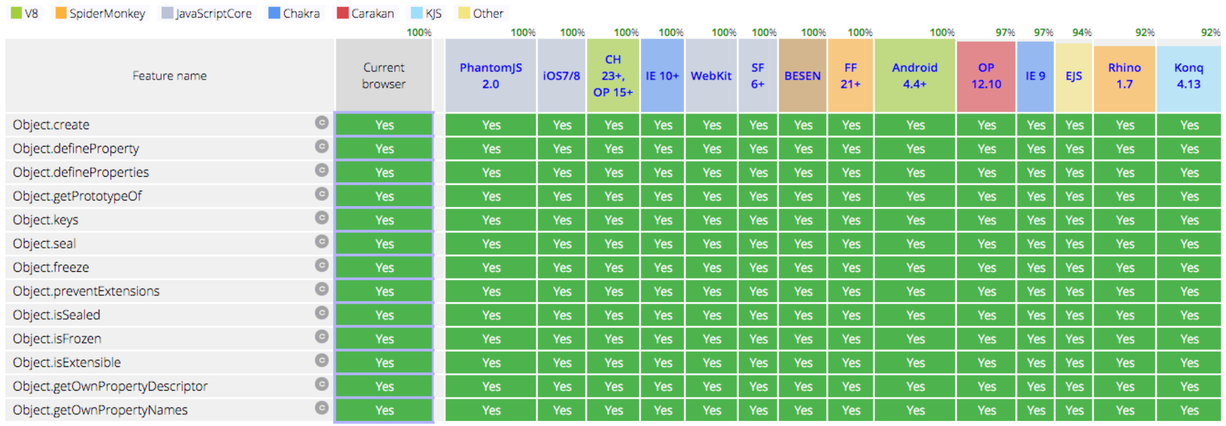
European Computer Manufacturers Association

History of JavaScript

- Alternatives started springing up in the late 1990s and early 2000's
  - Microsoft introduced JScript engine
  - Macromedia Flash was popular for facilitating the dynamic web
- Both were vaguely JavaScript-like, but standards differed

History of JavaScript

- Standards later converged
  - Firefox came out in 2005
  - Adobe bought Flash
  - JScript followed the standards
- But browser's implementations of the language still vary

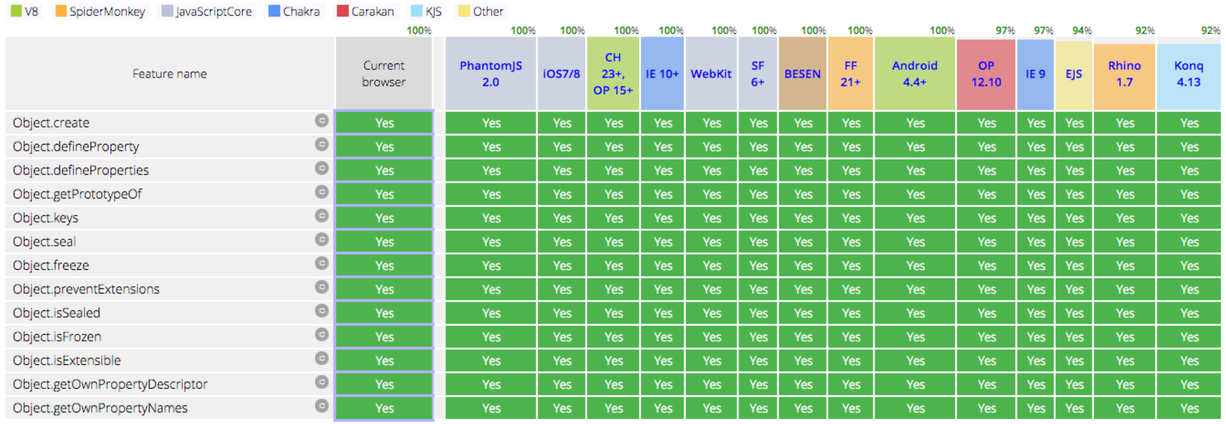


V8 SpiderMonkey JavaScriptCore Chakra Carakan KJS Other

Feature name	Current browser	PhantomJS 2.0	iOS7/8	CH 23+, OP 15+	IE 10+	WebKit	SF 6+	BESEN	FF 21+	Android 4.4+	OP 12.10	IE 9	EJS	Rhino 1.7	Konq 4.13
	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	97%	97%	94%	92%	92%
Object.create	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.defineProperty	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.defineProperties	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.getPrototypeOf	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.keys	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.seal	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.freeze	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.preventExtensions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.isSealed	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.isFrozen	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.isExtensible	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.getOwnPropertyDescriptor	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.getOwnPropertyNames	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

History of JavaScript

- JavaScript Engines
  - SpiderMonkey (Firefox)
  - V8 (Chrome)
  - JavaScriptCore (Safari)
  - Chakra (IE & Edge, no longer)
- Each of these engines follows the standard (ECMAScript), but implement JavaScript slightly differently



V8 SpiderMonkey JavaScriptCore Chakra Carakan KJS Other

Feature name	Current browser	PhantomJS 2.0	iOS7/8	CH 23+, OP 15+	IE 10+	WebKit	SF 6+	BESEN	FF 21+	Android 4.4+	OP 12.10	IE 9	EJS	Rhino 1.7	Konq 4.13
	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	97%	97%	94%	92%	92%
Object.create	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.defineProperty	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.defineProperties	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.getPrototypeOf	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.keys	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.seal	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.freeze	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.preventExtensions	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.isSealed	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.isFrozen	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.isExtensible	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.getOwnPropertyDescriptor	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Object.getOwnPropertyNames	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Versions of JavaScript

- Engines/Browsers continually play catch-up,
  so many tools support slightly older versions of the standard

Feature name	Current browser	Desktop browsers: IE 11	FF 115 ESR	FF 128 ESR	FF 133	FF 134	FF 135	CH 130	CH 131	CH 132	Edge 130	Edge 131	SF 17.6	SF 18	SF TP	WK	OP 111	OP 112
	98%	11%	98%	98%	98%	98%	98%	98%	98%	98%	98%	98%	100%	100%	100%	99%	98%	98%
Optimisation																		
proper tail calls (tail call optimisation)[6]	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	2/2	2/2	2/2	2/2	0/2	0/2
Syntax																		
default function parameters	7/7	0/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7
rest parameters	5/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5
spread syntax for iterable objects	15/15	0/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15
object literal extensions	6/6	0/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6
for..of loops	9/9	0/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9	9/9
octal and binary literals	4/4	0/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4
template literals	7/7	0/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7	7/7
RegExp "y" and "u" flags	6/6	0/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6

https://compat-table.github.io/compat-table/es6/

So, how do we account for different versions?

Libraries

- Other developers often write libraries to help address inconsistencies in implementations
  - For example: Better libraries for dealing with date and time
  - Then, you can use the library, rather than the language directly

https://moment.github.io/luxon/#/

Polyfills

- Polyfills ensure a user's browser has the latest libraries
  - Downloads "fill" versions of added functions, re-written using existing functions
- Sometimes called a "shim" or a "fallback"

Accounting for different versions

- But, a major lesson is compatibility requires testing.
  - Testing across browsers
  - Testing across devices

Slight transition: Cross-Platform development

What is cross-platform development?

- Write one set of code, run it on multiple platforms
  - Operating systems (Mac, Windows)
  - Devices (Android, iPhone)
- Kind of like responsive design, but extended to operating systems

https://en.wikipedia.org/wiki/Cross-platform_software

Cross-platform development

- Putting aside compatibility issues, a major advantage of the web is that it’s cross-platform
  - You can trust that everyone and every device will have access to a browser
- But, not everything can be a website

Question: why might a business want a mobile app over a mobile website?

Mobile app versus mobile website

- Need to access specific hardware capabilities
- Example: you want to build an Augmented Reality app
  - You *maybe* could do this in a browser, but it will be much harder than working with libraries intended for AR
- Lots of libraries aren't implemented in browsers, or implementations are limited
  - Notifications, Camera access, Apple Pay/Google Wallet, etc.

There are a variety of ways to build mobile apps

Native apps

- An app designed to work on a specific piece of hardware
- Usually built with tools created by the hardware or platform manufacturer
  - Android Studio for Android, in Java
  - Xcode for iOS, in Swift or Objective-C

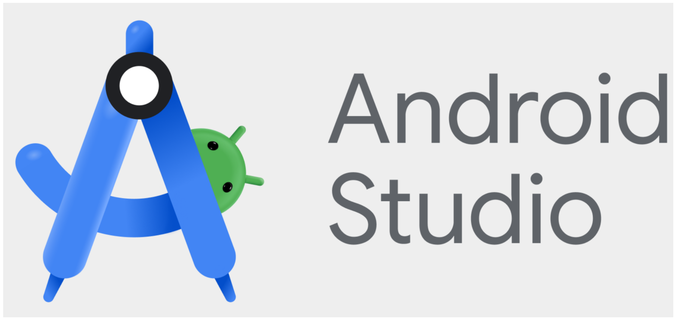
Native apps

- As we think of them today, native apps started with the first iPhone

- Released a development platform alongside the hardware

Native apps

- Platform-specific codebases
  - Android is in Java,
    iOS is in Objective-C or Swift
  - Both use different libraries
    to communicate with the hardware
- Usually require starting to code
  from scratch



What if we already made a website for our app? Or have some other existing codebase?

What if we want to share code across phone platforms?

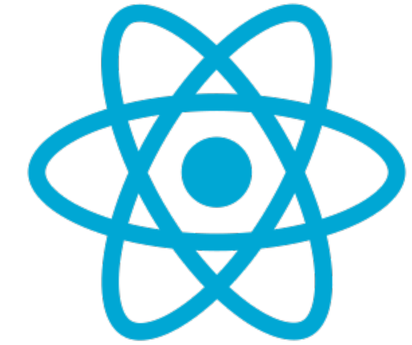
Solution: hybrid apps

Hybrid apps

- Use a common code base to deploy native-like apps on a wide range of platforms
- Typical strategy: run a webpage written in HTML/CSS/JavaScript, on the phone's internal browser
- Load that browser in a lightweight native app
- Ideally, expose some native APIs to the browser

Hybrid app frameworks

- These frameworks use web technologies (HTML, CSS, JavaScript)
  rather than platform-specific technologies



Pros and cons of each option

Strengths of hybrid apps

- Can share a codebase between web and mobile
- Can save time and effort (sometimes)
- Easily design for various form factors
- Access to some device capabilities

Weaknesses of hybrid apps

- Performance issues
- Inconsistency with the platform itself
- Limited access to device capabilities

Strengths of native apps

- Consistent experience with platform
- Leverages full device capabilities
- Uses native UI elements

Weaknesses of native apps

- Need to support separate development for each platform
- Cost of app development and maintenance
- Need to learn/manage multiple programming languages
- Need to manage multiple sets of tools

Today’s goals

By the end of today, you should be able to…

- Explain why compatibility testing is important in web development
- Search for compatibility challenges in implementation
- Describe advantages and disadvantages of developing native and web applications

IN4MATX 285:
Interactive Technology Studio

Practice: Compatibility and Cross-Platform