App Development

How the Internet Works in 5 Minutes

<https://www.youtube.com/watch?v=7_LPdttKXPc>

What is JavaScript?

<https://www.youtube.com/watch?v=nItSSTwBvSU>

Native Apps, Web Apps, Hybrid Apps

<https://www.mobiloud.com/blog/native-web-or-hybrid-apps/>

Web App

Application accessed via a web browser over a network such as the internet.

Web Site vs Web App

Web Site => informational i.e. Wikipedia

Web App => functional i.e. Facebook

Mobile App

Computer program designed to run on a mobile device.

Mobile apps better than web apps?

* Notifications
* Sharing
* Time
* Ad Revenue

API – application programming interface

The messenger (a software) that takes requests, and tells a system what to do, then returns the response. Since every program, has its own unique code, APIs’ do the translating between everything. Hence, creating connectivity and interactivity between everything. Think of ordering at a restaurant, the waiter is the API, receiving the request, tells the kitchen, returns with response.

SDK – software development kit

A set of tools used for developing applications for specific operating systems.

CMS – content management system

A program that helps create and manage content…i.e. create a blog without coding knowledge

i.e. WordPress

Native App

App written in languages that the platform accepts, created with their main or appropriate language. i.e. Swift for IOS, Java for Android

Why most apps are native

* Fast and Responsive
* Wider Functionality; interacts with other programs on the same platform easier
* UI matches with the platform

Why not?

* More codebases => one for IOS and one for Android
* Costs more $$$ and time

Main disadvantage…develop apps separately for each platform…only if you stick with the native SDKs provided by Apple and Google. This is when REACT NATIVE comes in.

Hybrid App

A cross between a native app and a web app.

Installed like a native app but is a web app on the inside.

Have an idea for an app, but don’t know if people will like it or not. Goal = MVP

Web app may be the minimal option but won’t allow you to test whether people will download or not. Hybrid app is the solution.

Advantages of a hybrid app

* One codebase to manage
* Save time and money
* Easier to scale to another platform
* Still have access to device features

Disadvantages of a hybrid app

* Performance – WebView, mimics native apps
* Cross-platform is tough
* UX will suffer
* Unlike a native app, with Appstore presence, push notifications, home screen icon, offline use

Choosing an app

<https://infinum.co/the-capsized-eight/android-development-is-30-percent-more-expensive-than-ios#disqus_thread>

* UX is important, best to do two native apps
* Time to market and cost – hours required and hourly cost
* Cost of hiring app developers
* Device features
* Performance
* Gestures
* Data processing needs
* Hiring developers…difficult because of high-demand, developers specialize
* Time required to build a native app – average of 18 weeks to build a standard native mobile app, 10 weeks for the back-end and 8 for the front end

” A 50%-good solution solves more problems and survives longer than a 99% solution that nobody has because it’s in your lab where you’re endlessly polishing the damn thing.” – Joel Spolsky, CEO and co-founder of Stack Overflow

Other Terminology

What is Node.js?

<https://www.youtube.com/watch?v=RF5_MPSNAtU>

Node.js is a JavaScript framework for writing custom servers.

What is a user interface?

<https://www.webopedia.com/TERM/U/user_interface.html>

User to computer program interaction.

An interface is a set of commands or menus through which a user communicates with a program.

What is ReactJS?

ReactJS is an open-source JavaScript library for building user interfaces.

What is a command line interpreter?

<https://www.lifewire.com/what-is-a-command-line-interpreter-2625827>

A command line interpreter is any program that allows the entering of [commands](https://www.lifewire.com/what-is-a-command-2625828) and then executes those commands to the [operating system](https://www.lifewire.com/operating-systems-2625912). It's literally an interpreter of commands.

What is Node Package Manager?

<https://docs.npmjs.com/getting-started/what-is-npm>

Windows Command Line Tutorial

<https://www.youtube.com/watch?v=MBBWVgE0ewk>

What is compiling in programming?

<https://www.youtube.com/watch?v=8zmaYXNiwZk>

If you wrote a program and directly gave it to the pc, it would not understand; the pc speaks in binary, not language. The act of compiling, is when the language is converted into binary so that the pc can understand and follow commands.

What is a transpiler/transcompiler (source to source compiler)?

A compiler that takes the [source code](https://en.wikipedia.org/wiki/Source_code) of a program written in one [programming language](https://en.wikipedia.org/wiki/Programming_language) as its input and produces the equivalent source code in another programming language.

What are Markup Languages?

<https://www.lifewire.com/what-are-markup-languages-3468655>

A markup language is a language that annotates text so that the computer can manipulate that text.

HTML – most common, all web pages are written in a flavor of HTML– defines the way images, multimedia and text are displayed in web browsers – father of CSS (W3C separated style and content because of the direction of HTML)

XML – a language for writing markup language – i.e. working with genealogy => tags of <father> <mother> etc.

XHTML – old HTML

What is Babel?

<https://www.quora.com/What-exactly-is-BabelJs-Why-does-it-understand-JSX-React-components?share=1>

Babel is an ECMAScript 6 (New JavaScript) to ECMAScript 5 (Old JavaScript) transpiler.

In short - it allows you to use language features “from the future”. It does so by transpiling your code to the currently supported version of JavaScript.



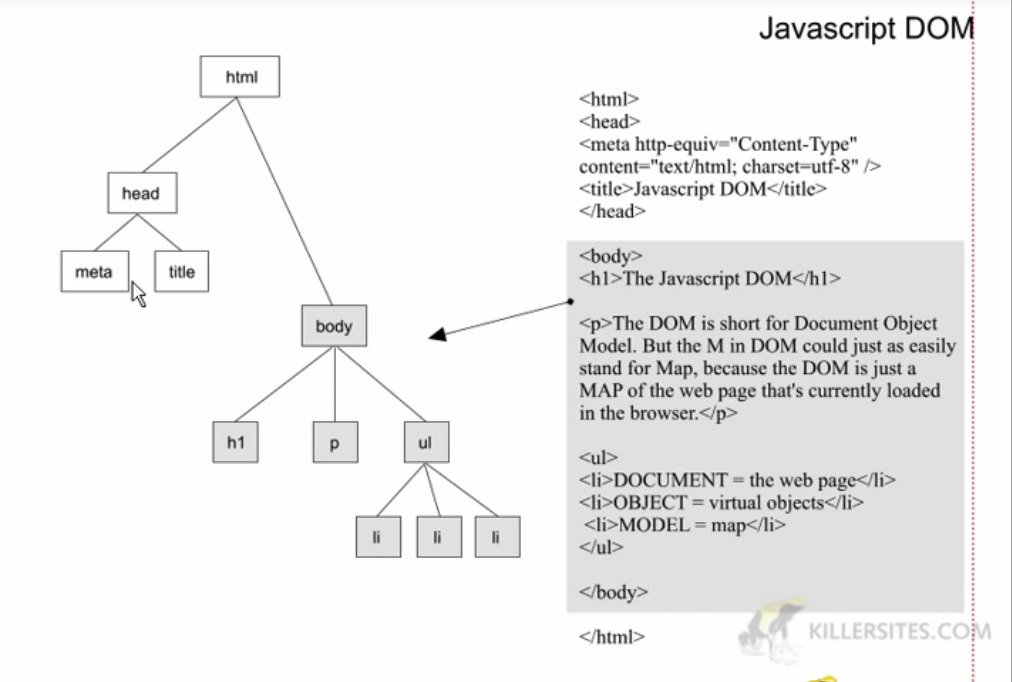
You need Babel because browser vendors are slow to adopt new language features.

What is DOM?

<https://www.youtube.com/watch?v=6DFGZnCUYgM>

<https://www.youtube.com/watch?v=-0ZcldkGlt8>

A map or model of the webpage, takes HTML tags and spreads it out on a tree.



React Native

React Native by Example

<https://www.youtube.com/watch?v=6ZnfsJ6mM5c>

React Native Crash Course

<https://www.youtube.com/watch?v=mkualZPRZCs>

Learning React Native – Where to Start

<https://hackernoon.com/learning-react-native-where-to-start-49df64cf14a2>

Quick Guide for React Native

<http://www.reactnativeexpress.com/>

React Basics

<https://www.youtube.com/watch?v=MhkGQAoc7bc>

<https://www.youtube.com/watch?annotation_id=annotation_2546538071&feature=iv&index=2&list=PLoYCgNOIyGABj2GQSlDRjgvXtqfDxKm5b&src_vid=MhkGQAoc7bc&v=fd2Cayhez58>

<https://www.youtube.com/watch?v=vu_rIMPROoQ&list=PLoYCgNOIyGABj2GQSlDRjgvXtqfDxKm5b&index=3>

<https://www.youtube.com/watch?v=qh3dYM6Keuw&index=4&list=PLoYCgNOIyGABj2GQSlDRjgvXtqfDxKm5b>

React Native is…

A framework used to build native cross-platform mobile apps with JavaScript and the React library.

Advantages of React Native

* Cross Platform
* Much less expensive
* Easier to code
* Save time
* Open source

Setting up React Native for Android

<https://www.youtube.com/watch?v=uhuoTcbquic>

The video above saves lives.

What are JavaScript modules?

<https://medium.freecodecamp.org/javascript-modules-a-beginner-s-guide-783f7d7a5fcc>

Good authors divide their books into chapters and sections; good programmers divide their programs into modules.

Like a book chapter, modules are just clusters of words (or code, as the case may be).

Good modules, however, are highly self-contained with distinct functionality, allowing them to be shuffled, removed, or added as necessary, without disrupting the system as a whole.

Maintainability, Name spacing, Reusability

ES6 Cheat Sheet

<https://es2017.io/?product_permalink=kOCPh&product_id=kOCPh&sale_id=noshare>

var, let, const

<https://www.youtube.com/watch?v=sjyJBL5fkp8>

<http://www.reactnativeexpress.com/block_scoped_declarations>

var – scoped to functions (don’t use this)

let – scoped to a block

const – scoped to a block (use for constants only)

Arrow functions

<https://www.youtube.com/watch?v=6sQDTgOqh-I>

<https://www.youtube.com/watch?v=mrYMzpbFz18>

<http://www.reactnativeexpress.com/fat_arrow_functions>

A shorter arrow function syntax

What is rendering (web development context)

<https://stackoverflow.com/questions/16518951/rendering-in-context-of-web-development>

Rendering is the process of gathering data, loading and applying it onto associated templates, then outputting it to the user (HTML as the template, embedded into the browser).

Exporting in JavaScript

<https://developer.mozilla.org/en-US/docs/web/javascript/reference/statements/export>

The export statement is used when creating JavaScript modules to export functions, objects, or primitive values from the module so they can be used by other programs with the [import](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import) statement.

Importing modules in JavaScript

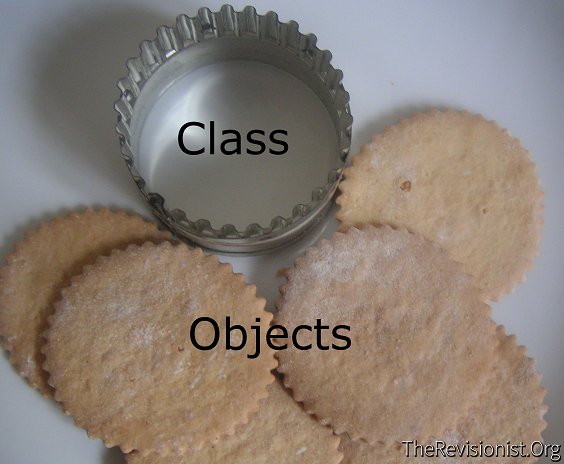
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import>

The **import** statement is used to import bindings which are exported by another module.

Video that summarizes ES6 import/export

<https://www.youtube.com/watch?v=Jqn_wjkSZwo>

Classes vs objects



What is hoisting?

<https://developer.mozilla.org/en-US/docs/Glossary/Hoisting>

Conceptually, for example, a strict definition of hoisting suggests that variable and function declarations are physically moved to the top of your code, but this is not in fact what happens. Instead, the variable and function declarations are put into memory during the compile phase but stay exactly where you typed them in your coding.

Ways of defining classes

Note\* class declarations and expressions are not hoisted.

1. Class declarations

Use the *class* keyword with the name of the class (what you’re used to).

1. Class expressions

Can be named or unnamed.

Unnamed

let Rectangle = *class* { … }

Named

let Rectangle = *class* Rectangle2 { … }

What is *AppRegistry*?

<https://facebook.github.io/react-native/docs/appregistry.html>

*AppRegistry* is the JS entry point to running all React Native apps. App root components should register themselves with *AppRegistry.registerComponent*, then the native system can load the bundle for the app and then actually run the app when it's ready by invoking *AppRegistry.runApplication*.

What is *extends?*

The extends keyword is used in [class declarations](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/class) or [class expressions](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/class) to create a class which is a child of another class.

class ChildClass extends ParentClass { … }

ES6 const

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/const>

A variable whose value cannot be reassigned.

const aNumber = 8;

What is JSX (JavaScript XML)?

a syntax for embedding XML within JavaScript/ a syntax extension to JavaScript/ an added set of JavaScript language rules.

Elements in React

<https://reactjs.org/docs/rendering-elements.html>

Elements are the smallest building blocks of React Apps.

They describe what you want to see on the screen.

They are what components are made of.

They are immutable; an element is like a single frame in a movie: it represents the UI at a certain point in time.

React – Components and Props

<https://reactjs.org/docs/components-and-props.html>

Conceptually, components are like JavaScript functions. They accept arbitrary inputs (called “props”/properties/) and return React elements describing what should appear on the screen.

React Native Props

<https://facebook.github.io/react-native/docs/props.html>

Most components can be customized when they are created, with different parameters. These creation parameters are called props.