Coding for Desktop and Mobile with HTML5 and Java EE 7

leanpub.com/
everythingjs

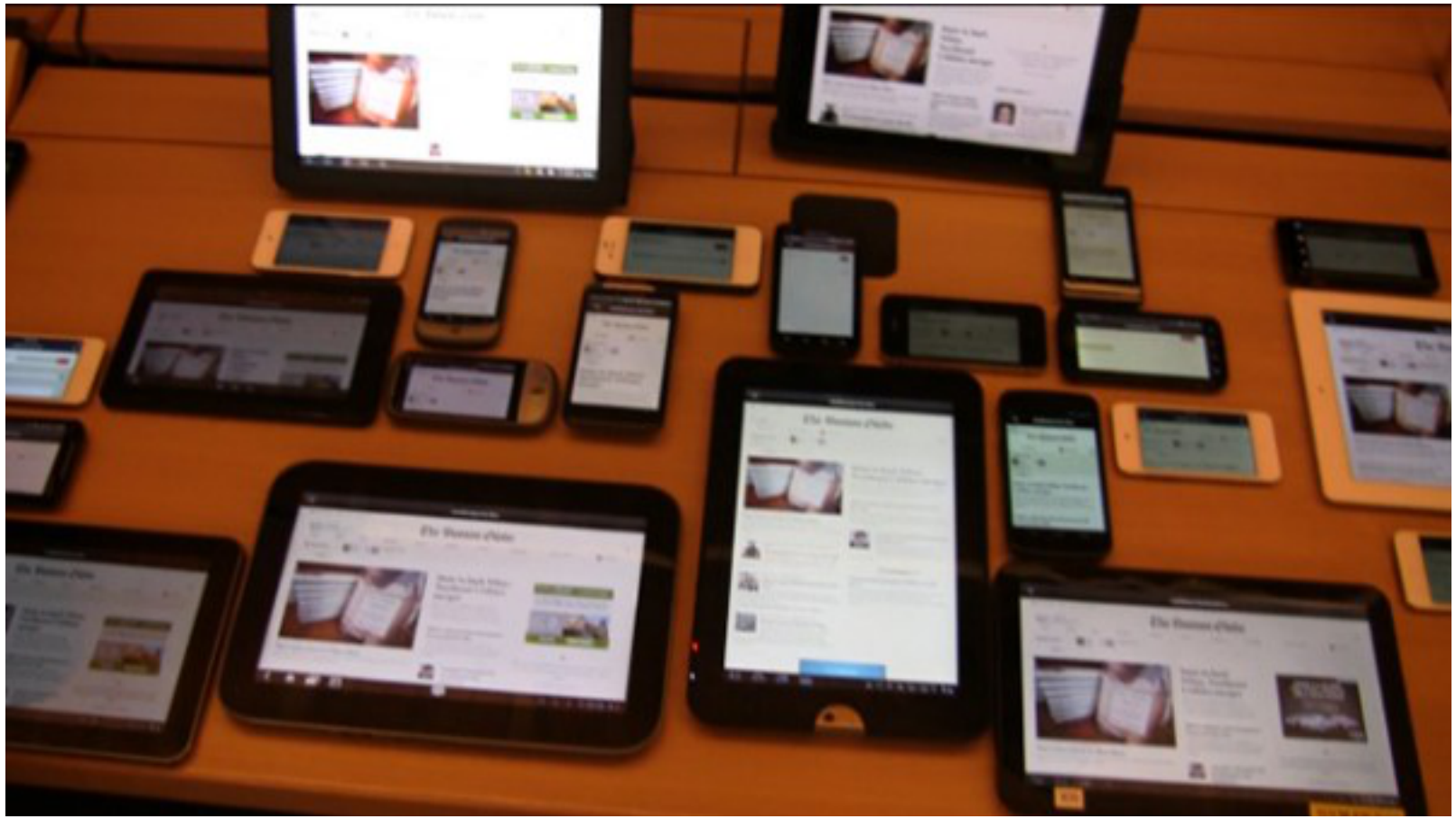
Coding for Desktop and Mobile with HTML5 and Java EE 7

Geertjan Wielenga
- NetBeans
- DukeScript
- VisualVM
- Jfugue Music Notepad
- Java
- JavaScript
@geertjanw

leanpub.com/
everythingjs

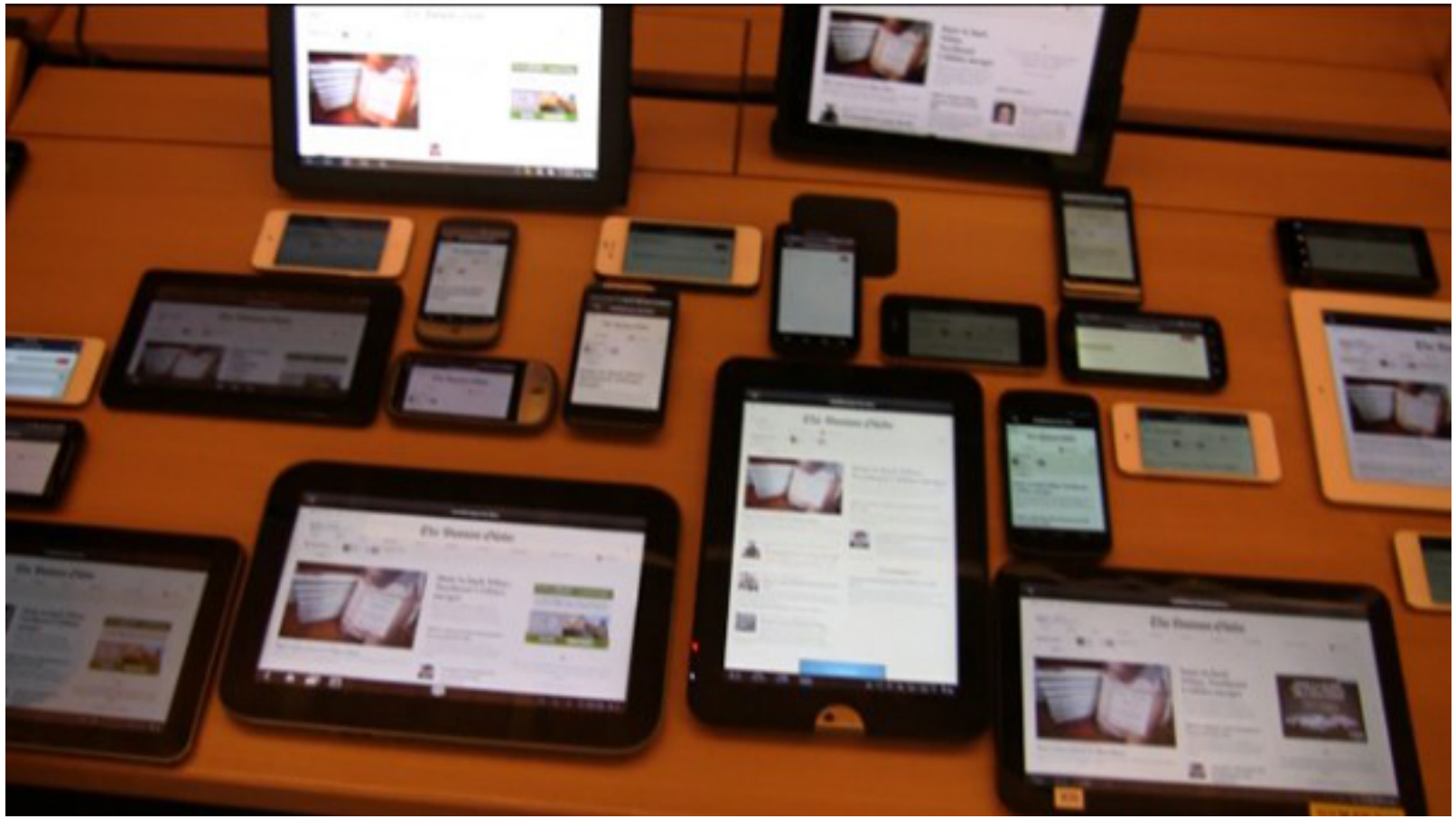
leanpub.com/
everythingjs

Multiple Devices / Responsive Design



leanpub.com/
everythingjs

The Browser Is Everywhere



leanpub.com/
everythingjs

SPA (Single Page Application)

- “A single-page application (SPA), is a web application or web site that fits on a single web page with the goal of providing a more fluid user experience akin to a desktop application.”

- SPA benefits are twofold:
  - less network bandwidth
  - faster navigation

10 Tips for Java Developers Exploring JavaScript in the Enterprise

- **HTML** is an application framework
- **CSS** is all about multimedia
- **JavaScript** is the assembly language of the web
- Many, many, many libraries, frameworks, and tools
- Maybe we should forget about Java altogether?

leanpub.com/
everythingjs

Tip 1: Resist the hype.

leanpub.com/
everythingjs

Tip 1: Resist the hype.

- Monitoring systems
- Management systems
- Large resolution requirements
- All users in single space

leanpub.com/
everythingjs

Tip 1a: Consider supplementing with new devices

- Mobile app to receive notifications
- Web app to display reports for upper management
- And then keep the system exactly as it was since that's how it makes sense

leanpub.com/
everythingjs

Tip 2: Think of HTML5 as an application framework

Tip 2: Think of HTML5 as an application framework

- Originally, HTML was designed as language for semantically describing scientific documents.
- 1999: HTML 4.01 released.
- 2008: First draft of HTML5.
- 2011: HTML5 released, poor support in browsers.
- 2015: All major browsers support it, in one way or another.

Tip 2: Think of HTML5 as an application framework

1. New features should be based on HTML, CSS, JavaScript.
2. Need for external plugins, e.g., Flash, should be reduced.
3. Error handling should be easier than previous versions.
4. Scripting should be replaced by more markup.
5. HTML5 should be device independent.

Tip 2: Think of HTML5 as an application framework

HTML5 Element Index

Head	Sections	Grouping	Tables	Forms	Forms 2	Embedded	Text-level	Text-level 2
doctype	body	p	table	form	keygen	img	a	time
html	article	hr	caption	fieldset	output	iframe	em	code
head	nav	pre	thead	legend	progress	embed	strong	kbd
title	aside	blockquote	tbody	label	meter	object	i, b	samp
base	section	ol	tfoot	input				
link	header	ul	tr	button				
meta	footer	li	th	select				
style	h1-h6	dl, dt, dd	td	datalist				
script	main	figure	col	optgroup				
noscript	address	figcaption	colgroup	option				
		div		textarea				

HTML5 form attributes

- placeholder
- autofocus
- autocomplete
- required
- pattern
- list
- multiple
- novalidate
- formnovalidate
- form
- formaction
- formenctype
- formmethod
- formtarget

leanpub.com/everythingjs

Tip 2: Think of HTML5 as an application framework

- Multimedia desktop experience on the web
- Focused on applications, animations, games, movies
- New **HTML media** elements, e.g., <audio>, <video>
- New **semantic markup**, e.g., <article>, <header>
- New **validation** attributes, e.g., “required”, “pattern”
- New **input types**, e.g., “email”, “url”, “color”
- New standard elements, implemented by each browser

Tip 2: Think of HTML5 as an application framework

- Shadow DOM separates content from presentation thereby eliminating naming conflicts and improving code expression. Shadow DOM addresses the DOM tree encapsulation problem.
- This specification describes a method of combining multiple DOM trees into one hierarchy and how these trees interact with each other within a document, thus enabling better composition of the DOM.

  http://www.w3.org/TR/shadow-dom/

Tip 3: Compare CSS vs. JavaScript for responsive design

Tip 3: Compare CSS vs. JavaScript for responsive design

- Originally, CSS was designed as style sheet language for describing the look and feel of a document written in a markup language like HTML.
- 1998: CSS2 released.
- 2011: CSS2.1 released.
- CSS3 under continual development.
- 2015: All major browsers support it, in one way or another.

Tip 3: Compare CSS vs. JavaScript for responsive design

Unlike the previous versions of the CSS specification, CSS3 has broken down the specification into smaller **"modules"**. Each module either describes the newer capabilities or changes to the previous specification. One key point here is that backwards compatibility is preserved.

As a result of breaking down the specification into smaller pieces, each module is in a distinct state of readiness. There are **over 50 CSS modules** that are in various states of readiness. **“Media Query”** is the most important one.

Tip 3: Compare CSS vs. JavaScript for responsive design

- Media Queries
- Allow developers to tailor to different resolutions without having to change or remove content.

```
@media screen and (max-width: 600px) {
    background: #FFF;
}
```

- Responsive design.
- However, responsive design can now also be done via JavaScript.

Tip 4: Evaluate framework vs. library

Tip 4: Evaluate framework vs. library

leanpub.com/
everythingjs

Tip 4: Evaluate framework vs. library

leanpub.com/
everythingjs

Tip 4: Evaluate framework vs. library

Module Systems

Build Tools

Tip 4: Evaluate framework vs. library

BACKBONE.JS

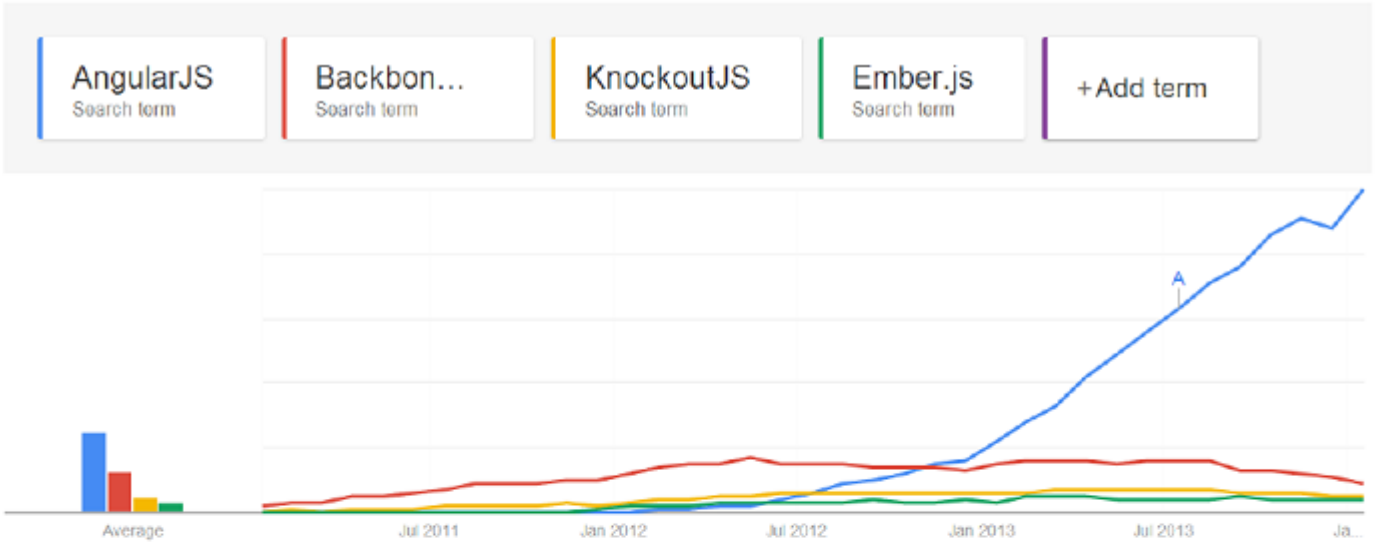
Ext JS

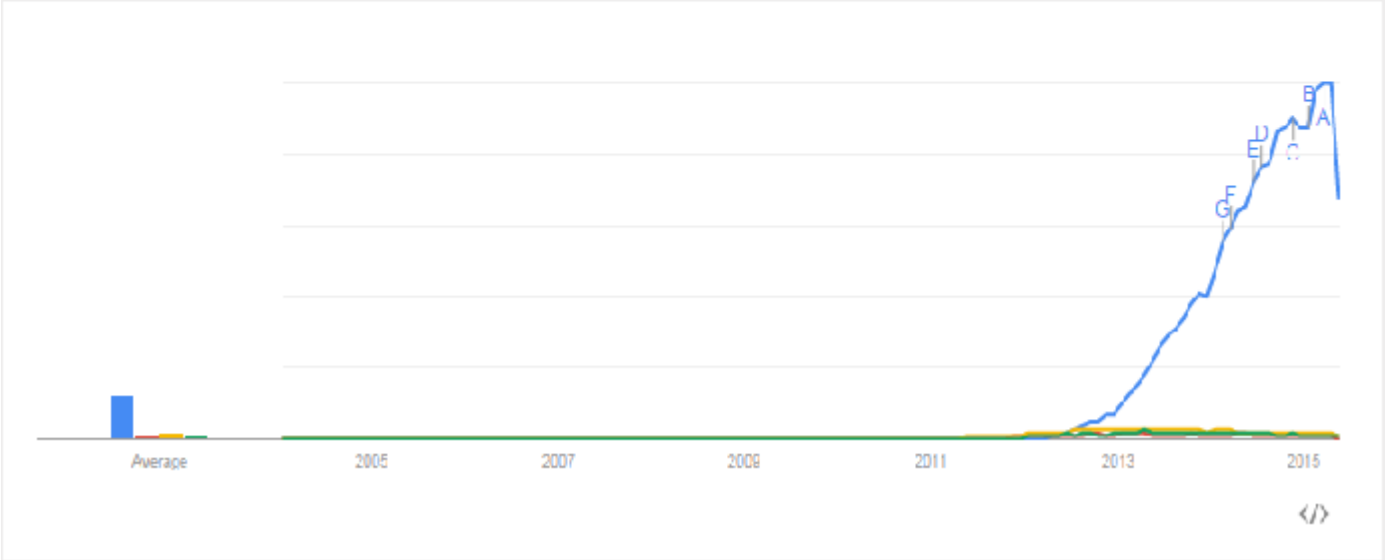
ember

Knockout.

METEOR

leanpub.com/
everythingjs





leanpub.com/
everythingjs

> Think of it this way: Knockout is primarily used to control UI representation in lower complexity applications, whereas Angular is a JavaScript framework that is much better suited for large, complex enterprise applications. It provides not only UI binding, but also best practices for application structure, development and testing.

https://www.devbridge.com/articles/angular-vs-knockout-similarities-and-fundamental-differences/

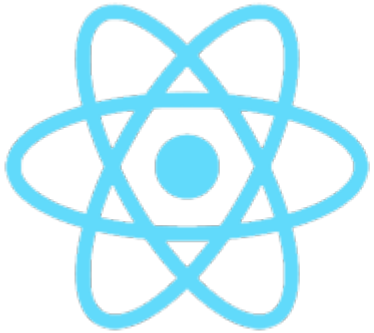
> I think the all-in-one approach of AngularJS makes it a great choice for small projects (less than 10 screens or 10-20 JavaScript files) that are using a SPA model or want to take advantage of several of AngularJS's capabilities. You have a single package to maintain with reasonable tools that can operate at higher scales and there are tons of examples available.

http://blogs.lessthandot.com/index.php/webdev/uidevelopment/angularjs-vs-knockout-final-thoughts-9/

Tip 4: Evaluate framework vs. library

- Backbone.js – Strong focus on MVC.
- Knockout.js – Strong focus on two-way data binding.
- AngularJS – Strong focus on “all in one” complete architecture.

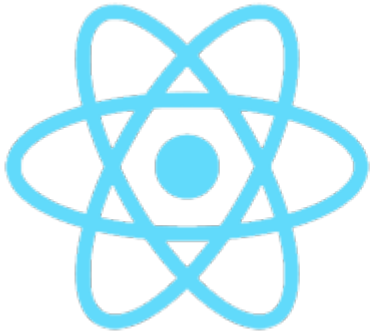
Tip 4: Evaluate framework vs. library



React

leanpub.com/
everythingjs

Tip 4: Evaluate framework vs. library



React

- Om – ClojureScript interface to React
- Mori – Persistent data structures
- Cortex – Centrally manage data
- Mercury – Module system
- Mithril – Similar framework

Tip 4: Evaluate framework vs. library

leanpub.com/
everythingjs

Tip 4: Evaluate framework vs. library

Data grids, menus, sliders, and other widgets.

leanpub.com/
everythingjs

Tip 4: Evaluate framework vs. library

leanpub.com/
everythingjs

Tip 4: Evaluate framework vs. library

YEOMAN

node

leanpub.com/
everythingjs

Tip 5: Evaluate corporate frameworks

Tip 5: Evaluate corporate frameworks

http://blog.backand.com/are-enterprises-migrating-to-angularjs

- Google – DoubleClick
- ING – SpectINGular
- Salesforce – Salesforce Lightning Platform
- IBM – MobileFirst Platform Foundation

Tip 6: Incorporate modularity

leanpub.com/
everythingjs

Tip 6: Incorporate modularity

leanpub.com/
everythingjs

Tip 7: Evaluate abstractions

leanpub.com/
everythingjs

Tip 7: Evaluate abstractions

leanpub.com/
everythingjs

Tip 8: Don't worry about ecosystem volatility

Tip 8: Don't worry about ecosystem volatility

- Lifespan of your app probably equivalent to lifespan of current version of your framework
- 1 ½ to 2 years lifespan
- Is the code hard to read? No worries, it'll be rewritten anyway.
- Don't worry so much about maintainability and backward compatibility.
- Things are changing fast, the ecosystem is already different right now than when this presentation started.

leanpub.com/everythingjs

Tip 9: Reorientate around WONTA

Tip 9: Reorientate around WONTA

leanpub.com/
everythingjs

Write **O**nce, **N**ever **T**ouch **A**gain

leanpub.com/
everythingjs

Write Once, Never Touch Again

leanpub.com/
everythingjs

~~Write Once, Run Anywhere~~

Write **O**nce, **N**ever **T**ouch **A**gain

leanpub.com/
everythingjs

Tip 10: JavaScript is the assembly language of the web

Tip 10: JavaScript is the assembly language of the web

```
; Example of IBM PC assembly language
; Accepts a number in register AX;
; subtracts 32 if it is in the range 97-122;
; otherwise leaves it unchanged.

SUB32  PROC          ; procedure begins here
       CMP  AX,97    ; compare AX to 97
       JL   DONE     ; if less, jump to DONE
       CMP  AX,122   ; compare AX to 122
       JG   DONE     ; if greater, jump to DONE
       SUB  AX,32    ; subtract 32 from AX
DONE:  RET           ; return to main program
SUB32  ENDP          ; procedure ends here
```

FIGURE 17. Assembly language

```
simple_loop:
# parameter 1: %rdi
..B1.1:                         # Preds ..B1.0
..___tag_value_simple_loop.1:   #2.1
        xorl      %eax, %eax    #3.19
        xorl      %edx, %edx    #5.8
        testq     %rdi, %rdi    #5.16
        jle       ..B1.5        # Prob 10% #5.16
                   #LOE rax rdx rbx rbp rdi r12 r13 r14 r15
..B1.3:                         # Preds ..B1.1 ..B1.3
        addq      %rdx, %rax    #6.5
        addq      $1, %rdx      #5.19
        cmpq      %rdi, %rdx    #5.16
        jl        ..B1.3        # Prob 82%   #5.16
..B1.5:                         # Preds ..B1.3 ..B1.1
        ret                     #8.10
        .align    2,0x90
```

leanpub.com/everythingjs

Tip 10: JavaScript is the assembly language of the web

An assembly language is a **low-level programming language** for a computer, or other programmable device, in which there is a very strong (generally one-to-one) correspondence between the language and the architecture's machine code instructions. Each assembly language is **specific to a particular computer architecture**, in contrast to most high-level programming languages, which are generally portable across multiple architectures, but require interpreting or compiling.

Wikipedia

Tip 10: JavaScript is the assembly language of the web

machine architecture: **browser**

machine language: **JavaScript**

“JavaScript is the assembly language of the web.”

– Scott Hanselman, 2011

leanpub.com/
everythingjs

“I said 'JS is the x86 of the web' a couple of years ago but I can't claim it's original.

The point is JS is about as low as we can go. But it also has higher level facilities.”

– Brendan Eich (inventor of JavaScript)

```
; Example of IBM PC assembly language
; Accepts a number in register AX;
; subtracts 32 if it is in the range 97-122;
; otherwise leaves it unchanged.

SUB32  PROC         ; procedure begins here
       CMP  AX,97   ; compare AX to 97
       JL   DONE    ; if less, jump to DONE
       CMP  AX,122  ; compare AX to 122
       JG   DONE    ; if greater, jump to DONE
       SUB  AX,32   ; subtract 32 from AX
DONE:  RET          ; return to main program
SUB32  ENDP         ; procedure ends here
```

FIGURE 17. Assembly language

```
simple_loop:
# parameter 1: %rdi
..B1.1:                         # Preds ..B1.0
..___tag_value_simple_loop.1:   #2.1
        xorl      %eax, %eax    #3.19
        xorl      %edx, %edx    #5.8
        testq     %rdi, %rdi    #5.16
        jle       ..B1.5        # Prob 10% #5.16
                  #LOE rax rdx rbx rbp rdi r12 r13 r14 r15
..B1.3:                         # Preds ..B1.1 ..B1.3
        addq      %rdx, %rax    #6.5
        addq      $1, %rdx      #5.19
        cmpq      %rdi, %rdx    #5.16
        jl        ..B1.3        # Prob 82%   #5.16
..B1.5:                         # Preds ..B1.3 ..B1.1
        ret                     #8.10
        .align    2,0x90
```

to act like they are the only company in the world?Edit <button id="po-z12hsf3b5s2xej3nw23vdlsazlrls5eyk04" name="eswidget" href="javascript:void(0);" class="esw eswd a-f-i-sb-e" g:entity="buzz:z12hsf3b5s2xej3nw23vdlsazlrls5eyk04" g:type="plusone" aria-pressed="false" title="Click to +1 this post"></button> - Comment - Share +16 by Laurent Bugnion and 15 others 2 shares - Behnam Yousefi and Felix M. Martinez 26 older comments from Larry Ruckman, Adrian Clark, Rafał Legiędź, Mendelt Siebenga, Scott Hanselman, mehrdad tat and John Hostile <img src="https://lh5.googleusercontent.com/-1JLP6eQk7mk/AAAAAAAAAAI/AAAAAAAAAAA/fbqGdArgmik/photo.jpg?sz=32" alt="Ernest Stormann's profile photo" class="a-f-i-q a-b-f-i-q a-f-i-W-Zb-z"> Ernest Stormann - Competition makes everyone better, but Darwin never said you wouldn't be tired.   <button id="po-z12hsf3b5s2xej3nw23vdlsazlrls5eyk04#1309992166066000" name="eswidget" href="javascript:void(0);" class="esw eswd a-f-W-sb-e" g:entity="comment:z12hsf3b5s2xej3nw23vdlsazlrls5eyk04#1309992166066000" g:type="plusone" title="Click to +1 this comment"></button> Add a comment... <h3 class="a-za">Scott Hanselman</h3><img src="//lh5.googleusercontent.com/-73Eb7tLaAEM/AAAAAAAAAAI/AAAAAAAAAG4/-J_WZn_VXQs/photo.jpg?sz=48" alt="Scott Hanselman's profile photo" class="a-f-i-q a-b-f-i-q"> Scott Hanselman -   - Mobile - Public I hope the g+ iPhone app is decent because this Safari nonsense, while technically impressive, is a jittery mess. HTML5 my ass. <button id="po-z12tdltjdrreslamw23vdlsazlrls5eyk04" name="eswidget" href="javascript:void(0);" class="esw eswd a-f-i-sb-e" g:entity="buzz:z12tdltjdrreslamw23vdlsazlrls5eyk04" g:type="plusone" aria-pressed="false" title="Click to +1 this post"></button> - Comment - Share +14 by Robert Scoble, John Sheehan, Laurent Bugnion and 11 others 1 share - Gianluca Gravina 11 older comments from Jonathan Sampson, Daniel Treadwell, M. Edward (Ed) Borasky, Michael Neale, Scott Hanselman and Bob

JavaScript is machine language for the browser.

leanpub.com/
everythingjs

Maybe we should forget about Java altogether?

leanpub.com/
everythingjs

Transpilers

Source-to-source compiler

From Wikipedia, the free encyclopedia

A **source-to-source compiler**, **transcompiler**, or **transpiler** is a type of compiler that takes the source code of a programming language as its input and outputs the source code into another programming language. A source-to-source compiler translates between programming languages that operate at approximately the same level of abstraction, while a traditional compiler translates from a higher level programming language to a lower level programming language. For example, a source-to-source compiler may perform a translation of a program from Pascal to C.

https://github.com/jashkenas/coffeescript/wiki/List-of-languages-that-compile-to-JS

Benefits of Transpilers

- Use a “better” (different, more familiar) language
- Make use of different language features
- Integrate with JavaScript landscape
- Same language on frontend and backend

Transpilers

- CoffeeScript
- TypeScript
- Dart
- GWT, Errai, Vaadin
- DukeScript

GWT, Errai, Vaadin

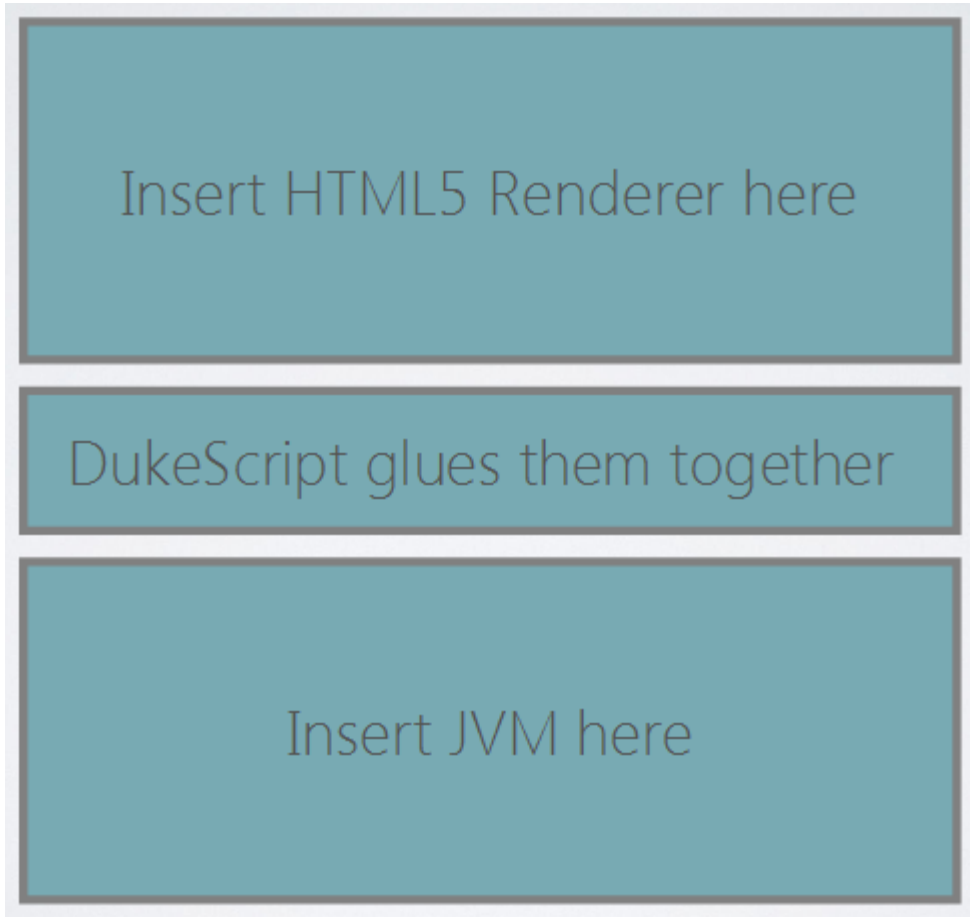
- Google Web Toolkit
- Compiler from Java to JavaScript
- Tons of other features
  - Build applications
  - Facilities for doing REST
  - UI components
- Errai: communications framework on GWT
- Vaadin: component framework on GWT

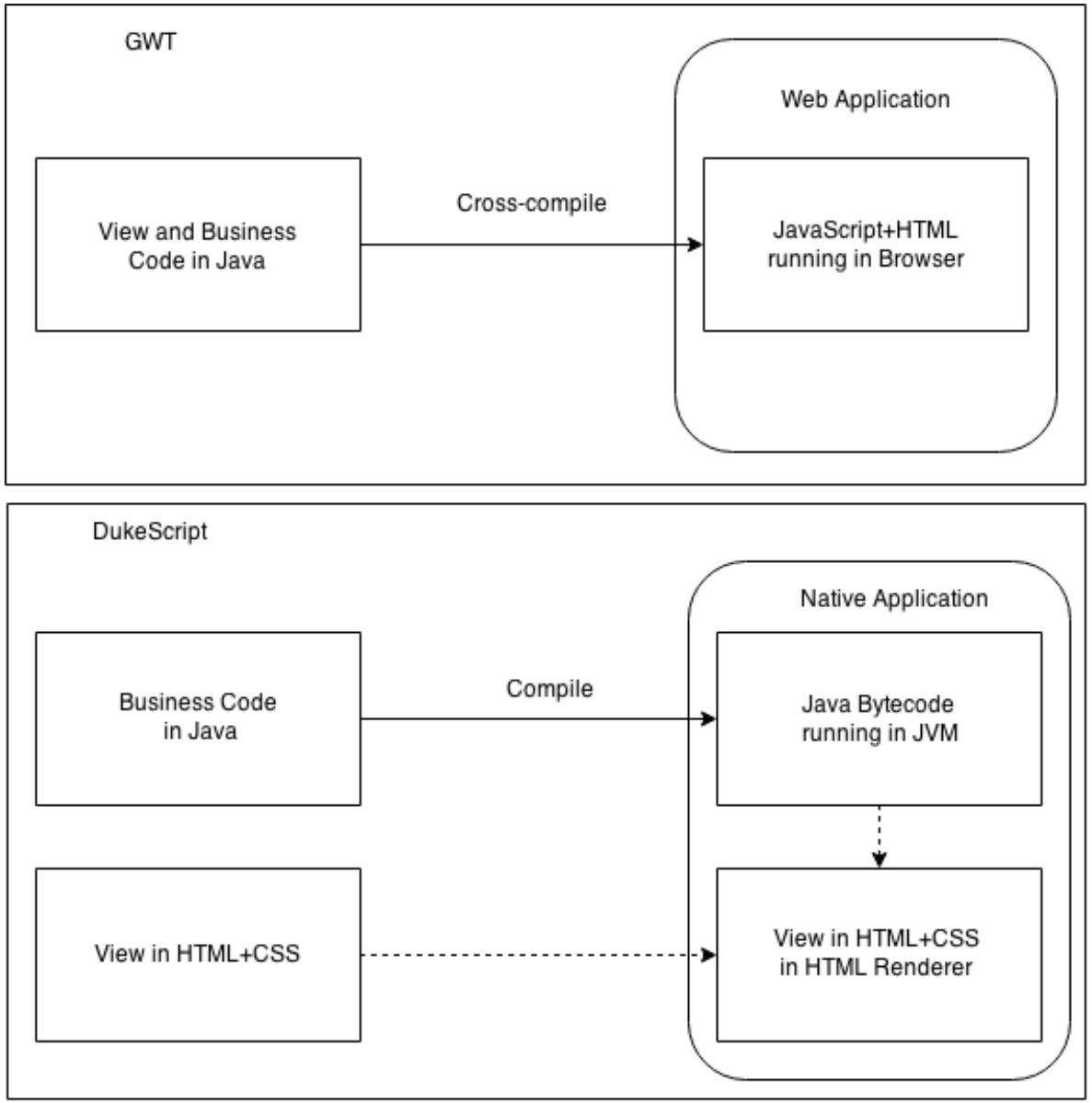
DukeScript

- Framework for creating cross-platform **mobile, desktop and web applications**.
- **Plain Java applications** that internally use HTML5 technologies and JavaScript for rendering.
- Write clean Java code and leverage the latest developments in modern UI technology.

leanpub.com/everythingjs

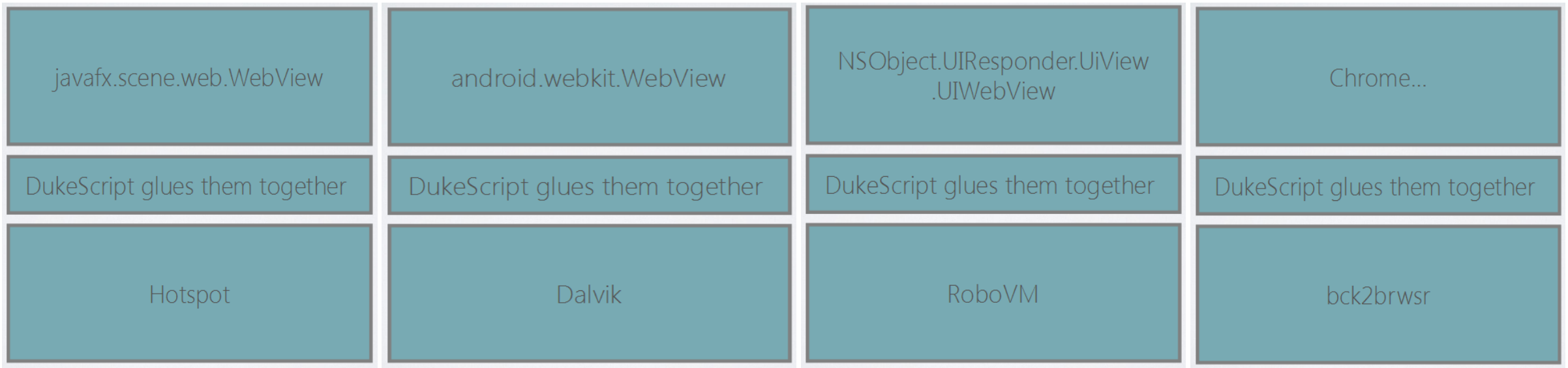
DukeScript





leanpub.com/everythingjs

DukeScript



leanpub.com/
everythingjs

Conclusions

HTML and CSS are much more powerful than before.

JavaScript is the assembly language
for modern web applications.

The browser is the target platform.

Either write JavaScript
or write Java and transpile to JavaScript
or stick to traditional server-side approach.

Choose wisely.

leanpub.com/
everythingjs

10 Tips for Java Developers Exploring JavaScript in the Enterprise

1. Resist the hype.
2. Think of HTML5 as an application framework.
3. Compare CSS vs. JavaScript for responsive design.
4. Evaluate framework vs. library.
5. Evaluate corporate frameworks.
6. Incorporate modularity.
7. Evaluate abstractions.
8. Don't worry about ecosystem volatility.
9. Reorientate around WONTA.
10. Look at JavaScript as an assembly language.

leanpub.com/
everythingjs