Arabic Unicode and Calligraphy

Nick Doiron - @{mapmeld}
Today I...

WILL

help you avoid embarrassing errors
explain Arabic / RTL support
show pretty calligraphy

WILL NOT

teach you Arabic
Right-to-left

Work into internationalization

Also Hebrew, Dhivehi, related languages
But I can’t read Arabic...

- Most letters connect
- Words have spaces
- Check left / right ends
Most times: Unicode does the work

User types د then ب then ي

They see دبی

Where are the vowels?
صل بابا ونولو إلى دار الصفاده.
سبرقهما اليرى أطفال كثيرون وهم يمرهون.
لولو سعيد الآن لأنه يلتقي أصدقائه.

صباع الخير يا فوفو
صباع الخير يا مني

جئت لألعب معكما.
These letters are so lonely 🙁

<span class='red'>ي</span> &zwj; ﻲ &zwj;
</span>
Making a right-to-left website
Pre-registration is full, but we will do our best to accommodate walk-ups.

Roots of Freedom conference shirts now available

Reserve a shirt

Hacker News

1. ▲ 3.5 Years, 500k Lines of Go (npf.io)
   116 points by NateDad 1 hour ago | flag | hide | 20 comments

2. ▲ Dell’s 32-inch 8K UP3218K Display Now for Sale: Check Your Wallet (anandtech.com)
   40 points by DiabloD3 49 minutes ago | flag | hide | 36 comments

3. ▲ German Space Center has constructed the world’s largest artificial Sun (newatlas.com)
   121 points by kamaal 3 hours ago | flag | hide | 29 comments

4. ▲ Amazon, the world’s most remarkable firm, is just getting started (economist.com)
   19 points by gpresot 57 minutes ago | flag | hide | 7 comments

5. ▲ Chasing the First Arcade Easter Egg (edfries.wordpress.com)
   70 points by anjalik 3 hours ago | flag | hide | 7 comments

6. ▲ Let’s Compile like it’s 1992 (fabiensanglard.net)
The `dir` attribute

```html
<body dir = "rtl">
  <input dir = "auto"/>
</body>
```

Changes *layout direction*

- LTR and RTL input look correct
- Not order of letters / words within tags
dir="rtl" and English placeholders
links and punctuation fixed with Arabic text
dir attribute does not cover everything
CSS

[dir="rtl"] .icon { ... }

Layouts using float: left / right and % widths

Uneven padding, icons, round corners

direction: rtl;
With CSS and RTL applied
Bootstrap 3: https://github.com/morteza/bootstrap-rtl
Numbers in Arabic
Not Arabic numerals
Supporting Nastaliq font
Graphics
Don’t just mirror it
Edge cases...

Why do ‘undo’ arrows point left?

Geolocation

FedEx
Calligraphy
Final Challenge: Geometric Calligraphy
We need text-shaping
Based on a real bug in Chrome / OpenStreetMap

Existing GPL libraries

Circle example will make more sense
Arabic Unicode: Extended Universe
Arabic Unicode: Extended Universe

Quran, Currencies, Calligraphy
Kashida “stretch Arabic”
A sample workflow for handling bi-di text

1. Determine that content contains RTL text
2. Not Arabic:
   a. Pair letters + modifiers in ‘RTL buffer’
   b. Read pairs right-to-left
3. Arabic:
   a. Add letters + modifiers based on position
   b. Add ligatures
if (!arabicRegex.test(inputText)) {
    // Hebrew or Thaana RTL script
    for (var n = 0; n < inputText.length; n++) {
        var c = inputText[n];
        if (((thaanaVowel.test(c) || hebrewSign.test(c)) && rtlBuffer.length) {
            rtlBuffer[rtlBuffer.length - 1] += c;
        } else if (rtlRegex.test(c)) {
            rtlBuffer.push(c);
        } else if (c === ' ' && rtlBuffer.length) {
            // whitespace within RTL text
            rtlBuffer = [rtlBuffer.reverse().join('') + ' '];
        } else {
            // non-RTL character
            ret += rtlBuffer.reverse().join('') + c;
            rtlBuffer = [];
        }
    }
}
let previous_output = output[output.length - 1];
if (previous_output[FORM] === NOT_SUPPORTED) {
    // not Arabic before this one
    output.push([letter, 'isolated']);
} else if (!_connects_with_letter_before(letter)) {
    // this letter doesn't try to connect with previous
    output.push([letter, 'isolated']);
} else if (!_connects_with_letter_after(previous_output[LETTER])) {
    // previous letter doesn't try to connect to me
    output.push([letter, 'isolated']);
} else if (previous_output[FORM] === 'final' & & (!_connects_with_letters_before_and_after(previous_output[LETTER]))) {
    // previous letter was final and cannot be medial to connect to me
    output.push([letter, 'isolated']);
} else if (previous_output[FORM] == 'isolated') {
    // previous letter was alone - we can change it to be initial of my phrase
    // for now this letter is the final of the phrase
    output[output.length - 1][1] = 'initial';
    output.push([letter, 'final']);
} else {
    // previous letter can be changed to medial
    // this one can be final
    output[output.length - 1][1] = 'medial';
    output.push([letter, 'final']);
}
for (let x = 0; x < LIGATURES.length; x++) {
    let ligature = LIGATURES[x];
    for (let y = 0; y < ligature.matches.length; y++) {
        let pattern = ligature.matches[y];
        let textFragment = text;
        let textFragmentOffset = 0;
        while (textFragment.indexOf(pattern) > -1) {
            // determine which ligature form to use
            let a = textFragment.indexOf(pattern);
            let start_form = output[a + textFragmentOffset][FORM];
            let end_form = output[a + textFragmentOffset + pattern.length - 1][FORM];
Now in outie and innie styles!
“بِنِينَ”

.length === 1

Arabic Unicode and Calligraphy

Nick Doiron - @mapmeld
ndoiron@mapmeld.com