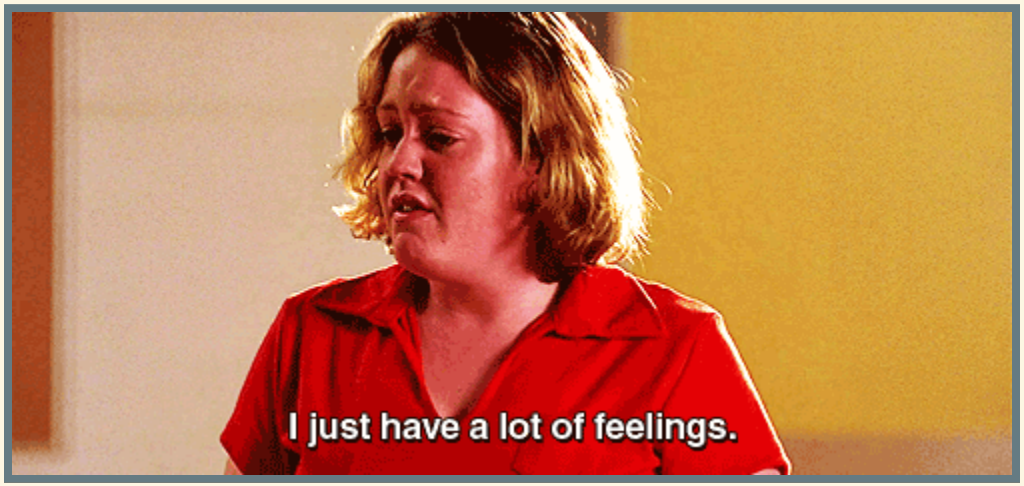
UNICODE!

(AND HOW ES6 CAN HELP)

By Eddie Antonio Santos / @_eddieantonio

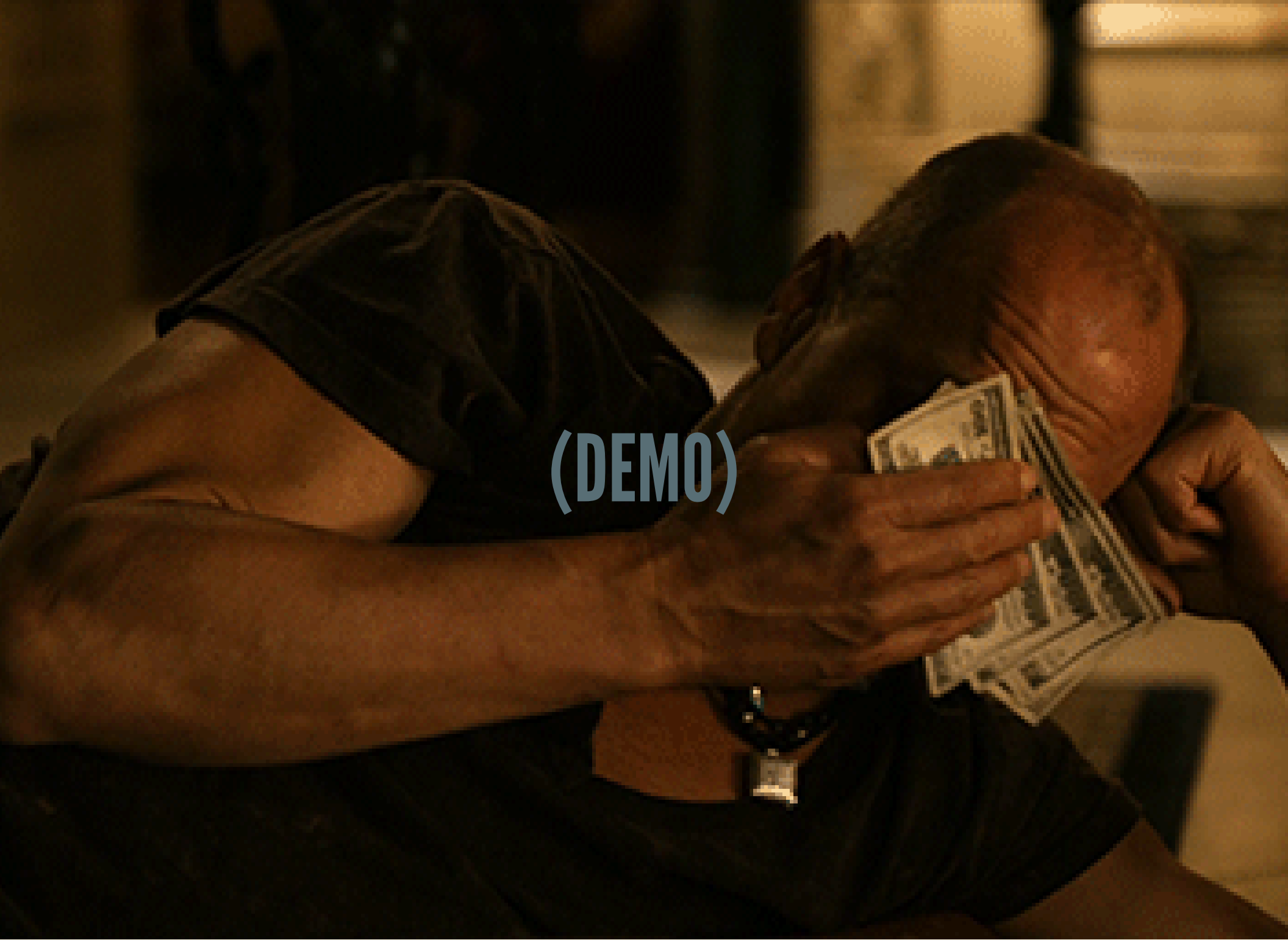
Source: eddieantonio.ca/unicode-es6

(HYPOTHETICAL SCENARIO)


I just have a lot of feelings.


bitter

(Thanks, Kim!)


(DEMO)

WHAT DOES `String.length` ACTUALLY MEASURE?

Let's consult the standard!

*The String type is the set of all ordered sequences of zero or more **16-bit unsigned integer values** ("elements")*

---

*The String type is generally used to represent textual data in a running ECMAScript program, in which case each element in the String is treated as a **UTF-16 code unit** value.*

---

*The length of a String is the **number** of elements (i.e., **16-bit values**) within it.*

ECMAScript 6.0 Standard §6.1.4

WHAT THE IS A UTF-16 CODE UNIT VALUE?

UNICODE!

WHAT IS UNICODE?

- A mapping of numbers (*code points*) to every character. Ever.
- Database of properties for each character (e.g., name, general category).

CODE POINTS

Unique number given to each character

They look like this:

U+*hhhh* or U+*hhhhhh*

Range from U+0000 to U+10FFFF

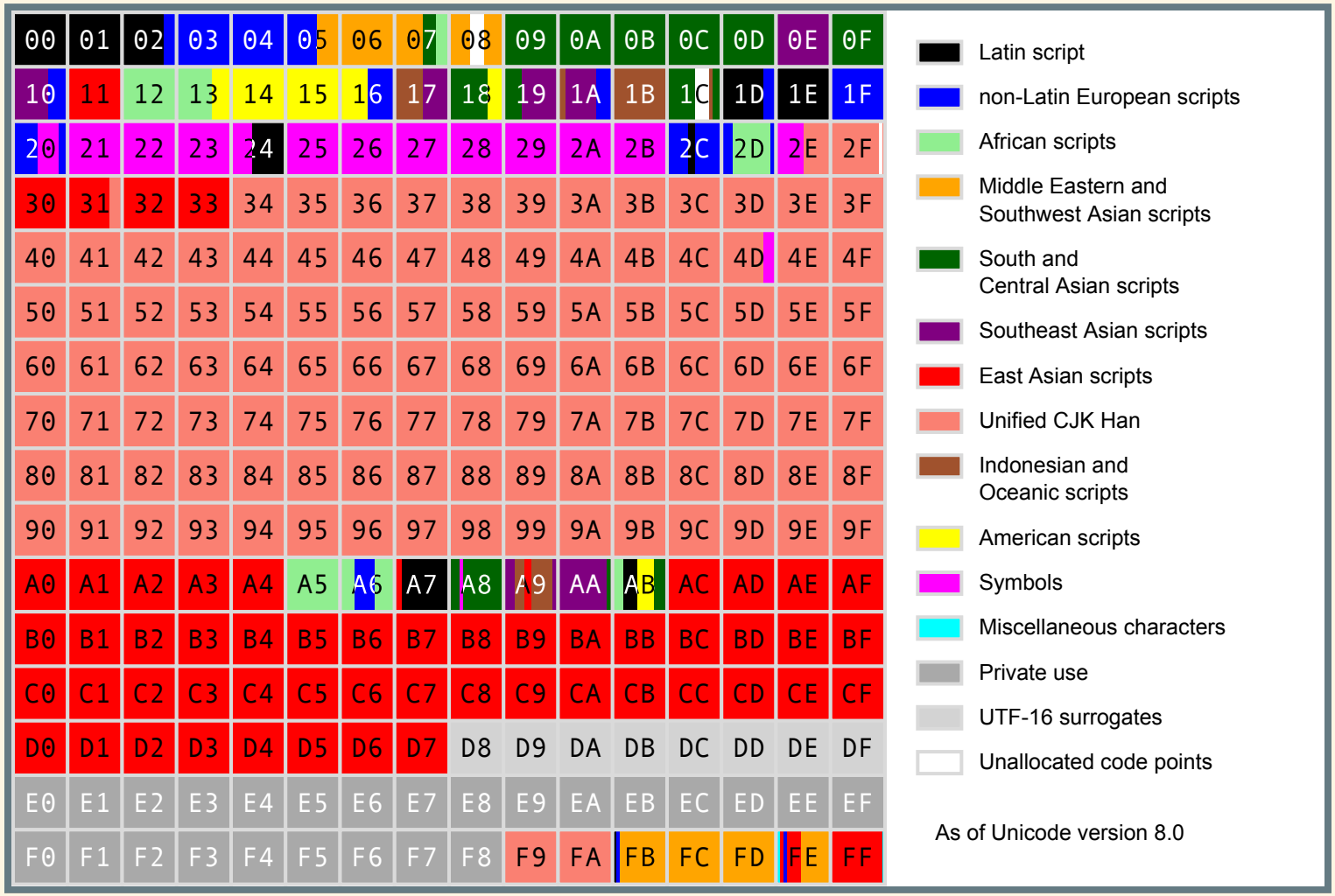
1,114,112 code points available in total

(See Unicode Chapter 3)

A TOUR OF THE UNICODE CHARACTER SPACE!

Code points are divided into 17 *planes*

THE BASIC MULTILINGUAL PLANE

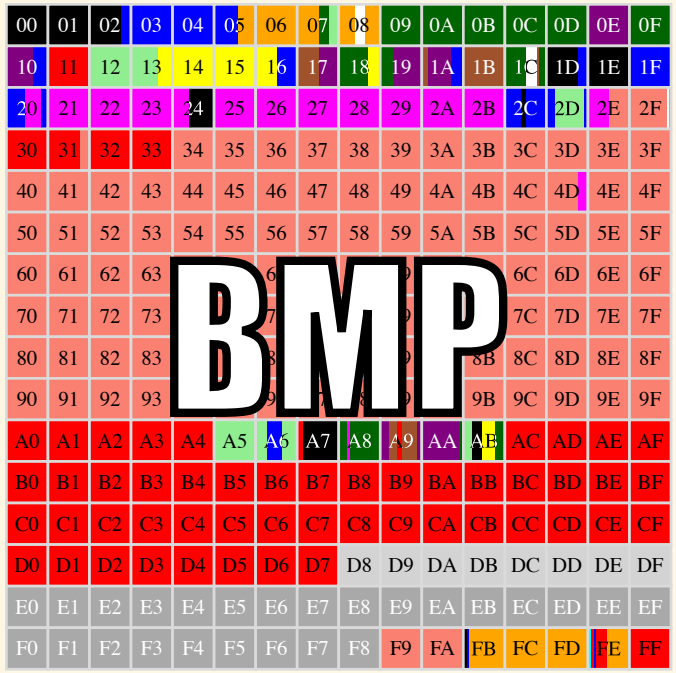


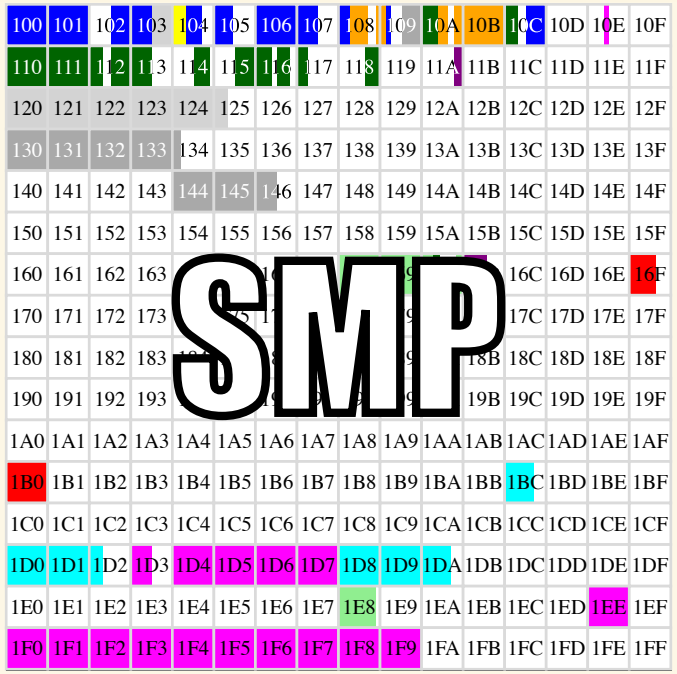
THE BASIC MULTILINGUAL PLANE

(PLANE 0)

- Characters from practically all widely-used modern-day scripts
- Code points are notated as U+*hhhh*
- Code points range from U+0000 to U+FFFF

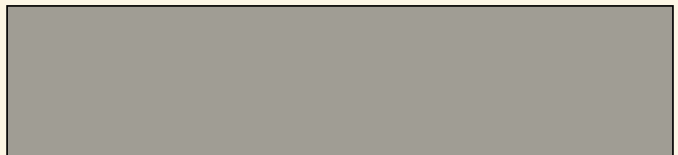
ALL UNICODE CODE POINTS

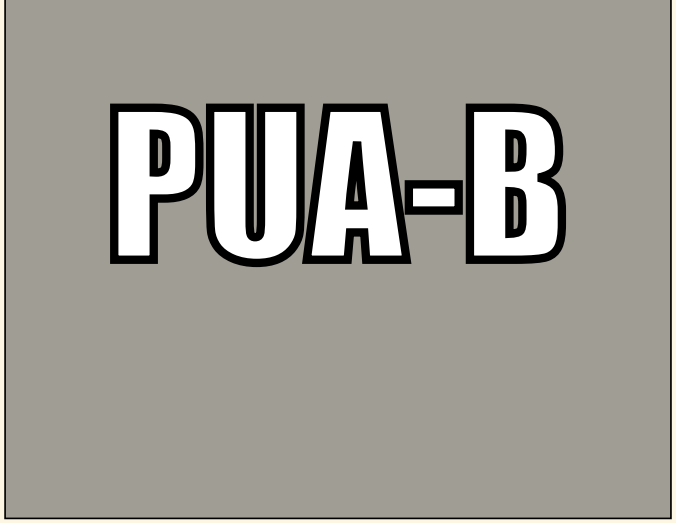

BMP


SMP

THE ASTRAL PLANES

- Everything else (Planes 1-16)
- Characters from ancient scripts, alternative scripts, pictograms, and rare and archaic CJK(V) ideograms (Chinese-style characters). Also, (most) Emoji.
- Two entire planes devoted to *private use characters*
- Code points are notated as U+*hhhhhh*
- Code points range from U+010000 to U+10FFFF





WHAT IS *NOT* UNICODE?

- ~~a character encoding~~ It's several character encodings!
- Code points ≠ Bytes

CODE UNIT

Smallest unit of storage required to store or transmit a single character in an encoding scheme

WAYS OF TRANSMITTING CODE POINTS

- UTF-8
- UTF-16
- UTF-32/UCS-4


UTF-16 NEEDS TWO CODE UNITS TO REPRESENT
ONE ASTRAL CODE POINT

BACK TO OUR PROBLEM...

We want to count **code points** and not **code units**

ENTER:

`String.prototype[@@iterator]`

(`String.prototype.codePointAt()` exists too)

When the @@iterator method is called it returns an Iterator object (25.1.1.2) that ***iterates over the code points*** *of a String value, returning each code point as a String value.*

COMPARE

```
let a = [];
for (let c of s) {
  a.push(c);
}
```

VS.

```
var i, a = [],
for (i = 0; i < s.length; i++) {
  a.push(s[i]);
}
```

LET'S FIX OUR CODE!

Trick: Use Array#from

(it just does this:)

```
function (s) {
  let a = [];
  for (let c of s) {
    a.push(c);
  }
  return a;
}
```

CHANGE OF PLANS

bitter

(Thanks, Kim!)

(DEMO)

THREE DIFFERENT WAYS OF WRITING

- pho = **o + ○ + ○**
- pho = **Matthew + ○**
- pho = **in**

NORMALIZATION FORMS!

Useful for comparing *different representations* of the same abstract character sequence

- **NFD** Canonical decomposition
- **NFC** Canonical decomposition, followed by Canonical Composition
- **NFKD** Compatibility Decomposition
- **NFKC** Compatibility Decomposition, followed by Canonical Composition

(See UAX # 15)

CANONICAL (DE)COMPOSITION

- NFD : ⇒ o᷈ o᷈
- NFC : on ⇒ in

ENTER:

`String.prototype.normalize()`

When in doubt, use

NFC

LET'S FIX OUR APP!

COMPATIBILITY

Check the compatibility table!

ASK ME QUESTIONS

RESOURCES

STANDARDS

- Unicode 8.0
- ECMAScript 6

UNICODE

- Mathias Bynens: JavaScript ♥ Unicode
- Tom Scott: Accidental Emoji Expert
- Tom Scott: Characters, Symbols and the Unicode Miracle
- Tom Scott: Why You Can Tweet More In Japanese: What Counts As A Character?

OTHER

- UTF-8 Everywhere
- JavaScript has a Unicode problem