

June 19, 2018

The Art of the Font

by Ayal Spitz

Accenture Interactive

Agenda

- 01 Introduction
- 02 Motivation & History
- 03 Fonts & Apple
- 04 CoreText
- 05 TextKit
- 06 Dynamic Kit
- 07 Closing Thoughts

Introduction

01

Ayal Spitz

Senior Principal Software Engineer

Accenture Interactive

@aspitz

ayal.spitz@accenture.com



Motivation & History

02

Motivation & History

The Art of the Font by Ayal Spitz

Code ➔ Compile ➔ Run

Motivation & History

The Art of the Font by Ayal Spitz

Code ➔ Compile ➔ Run

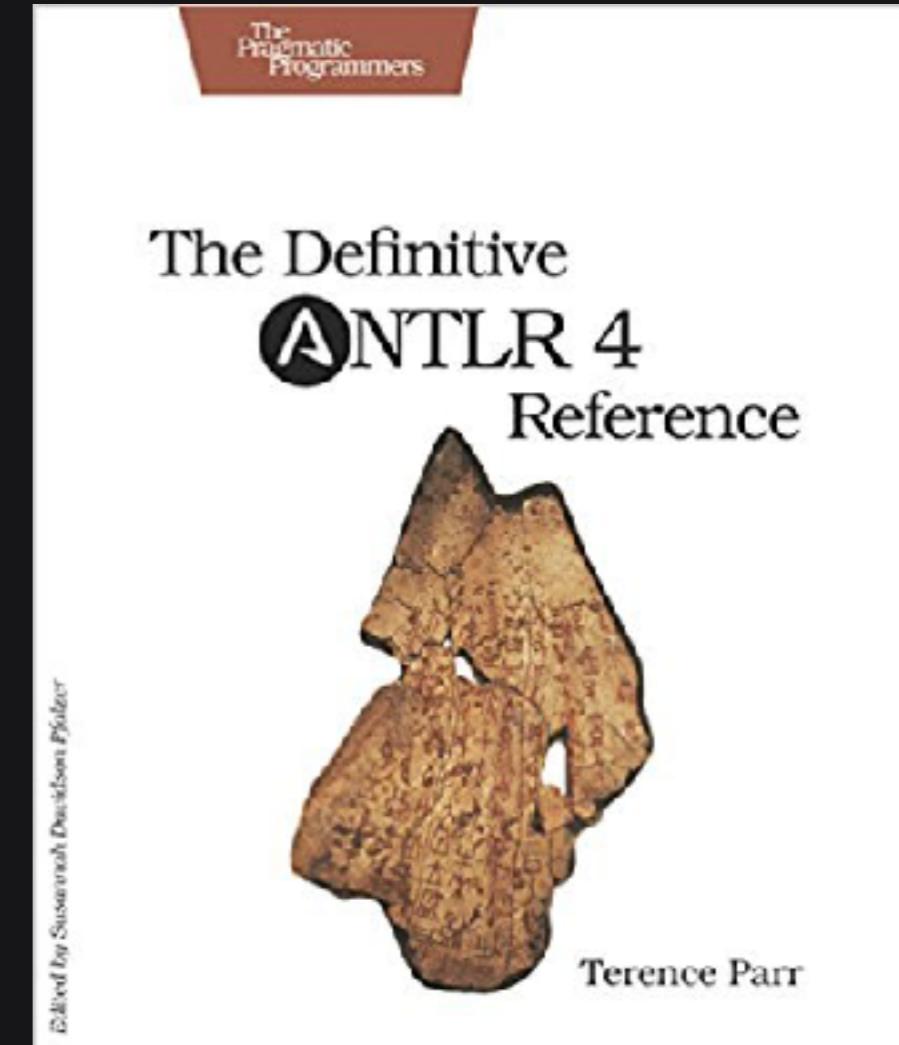
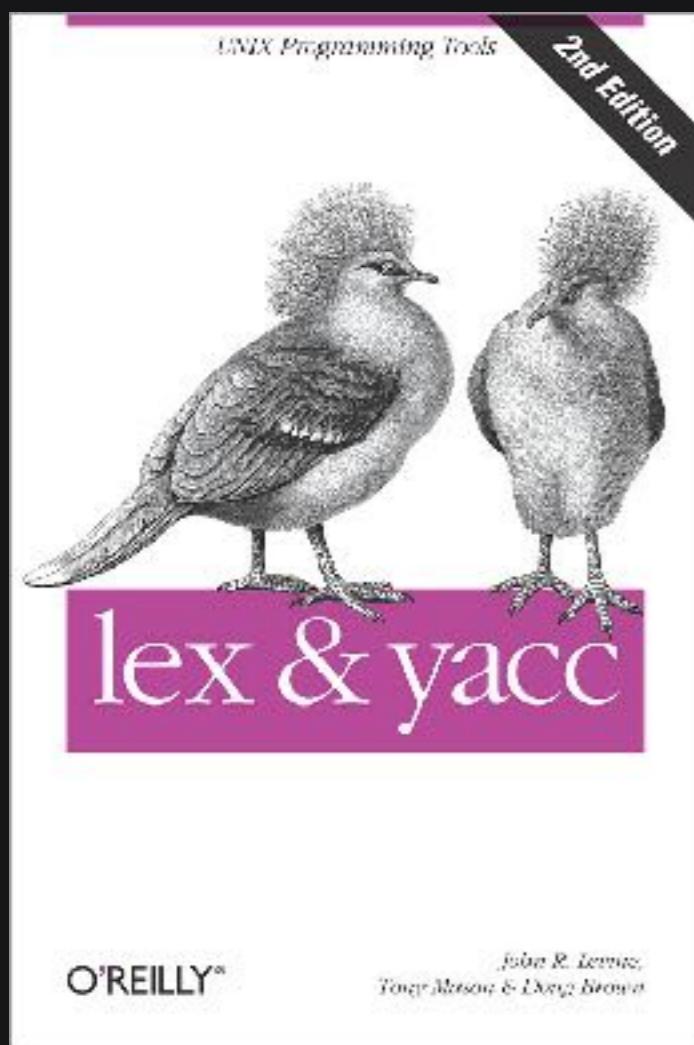
Motivation & History

The Art of the Font by Ayal Spitz



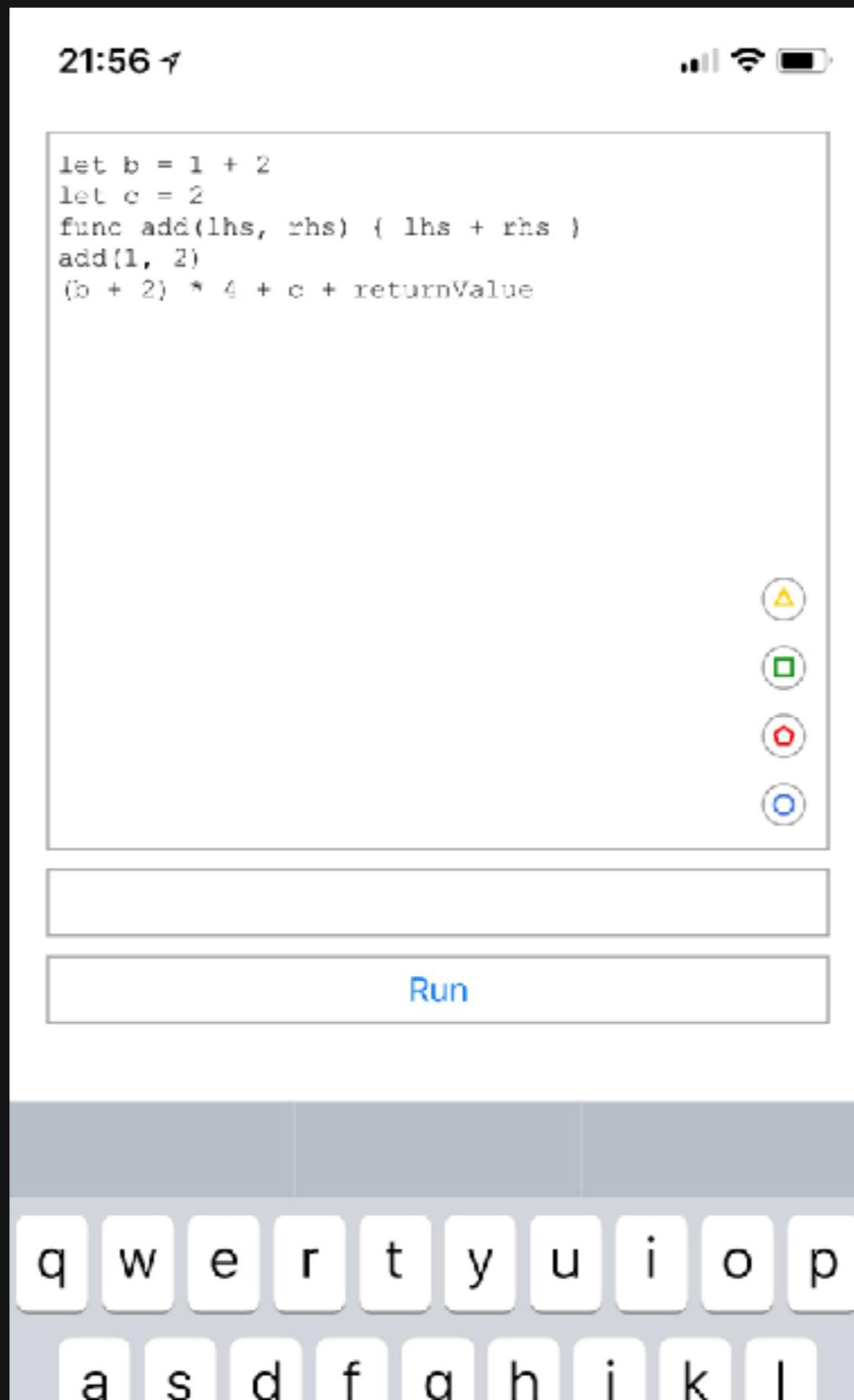
Motivation & History

The Art of the Font by Ayal Spitz



Motivation & History

The Art of the Font by Ayal Spitz



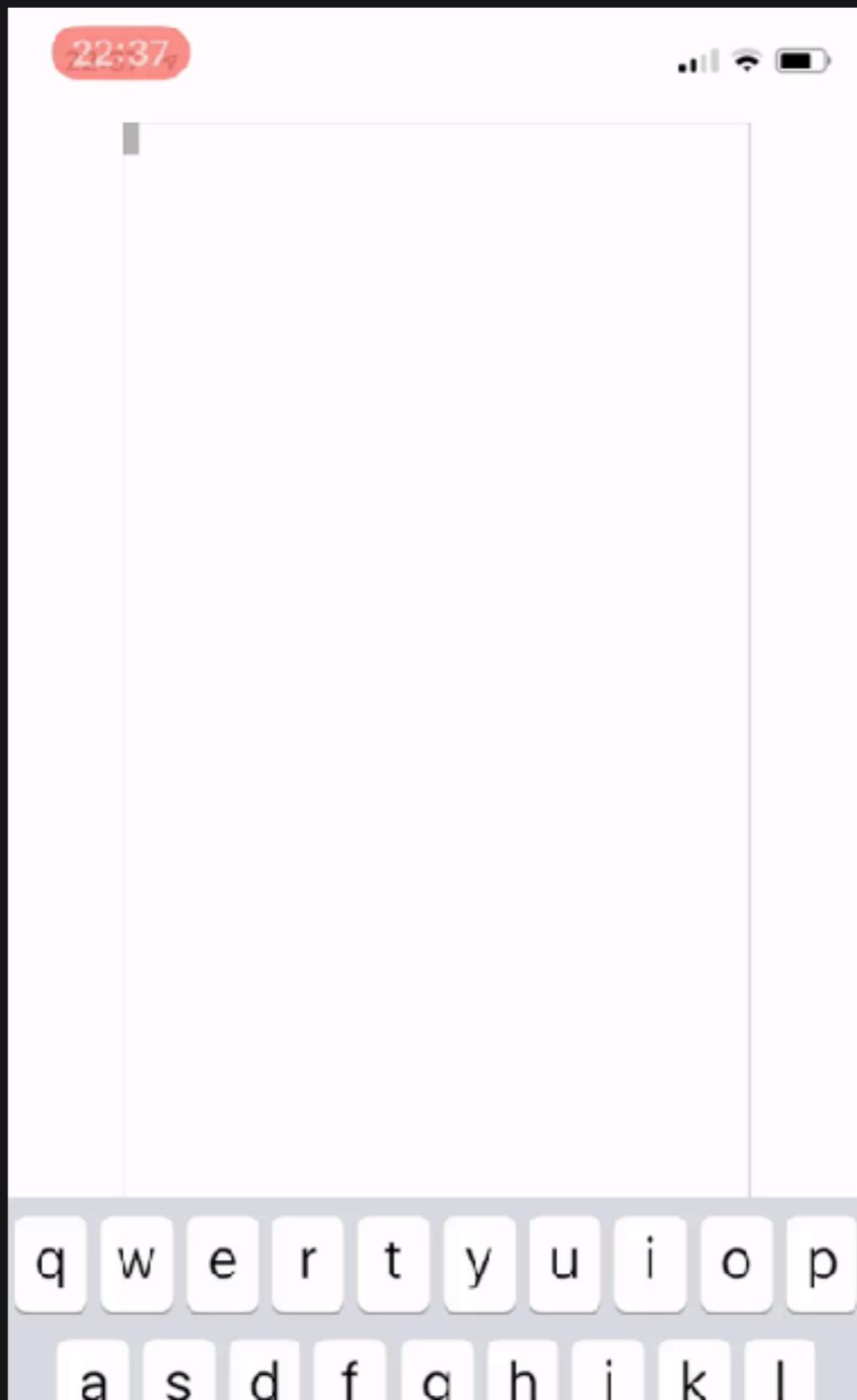
Motivation & History

The Art of the Font by Ayal Spitz



Motivation & History

The Art of the Font by Ayal Spitz



Motivation & History

The Art of the Font by Ayal Spitz

Calligraphy

Reed College



Motivation & History

The Art of the Font by Ayal Spitz

Calligraphy

[ke' ligrafe]**noun**

decorative handwriting or handwritten lettering

origin & etymology

greek

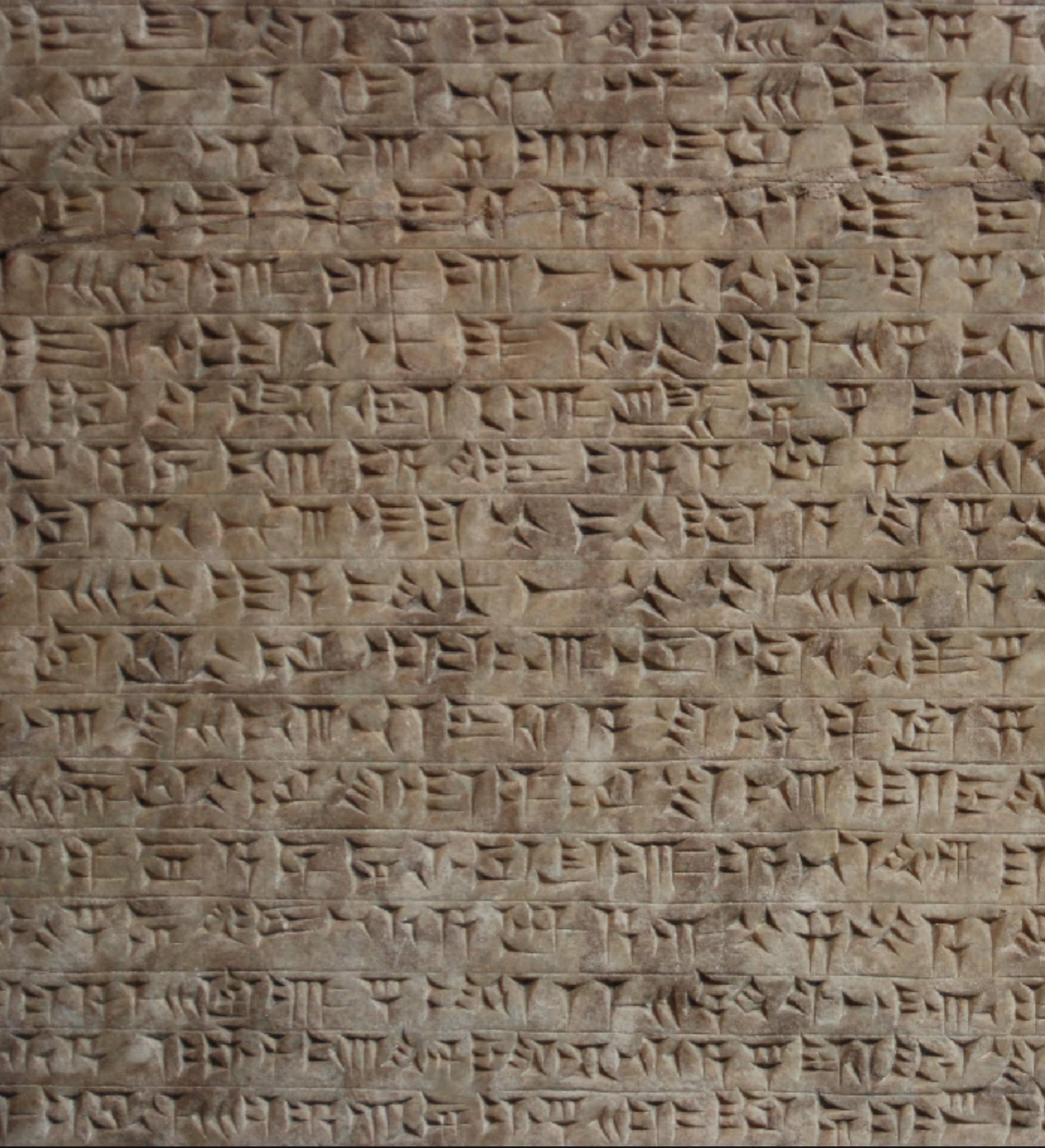
kallos - "beautiful"

graphein - "to write"

Motivation & History

The Art of the Font by Ayal Spitz

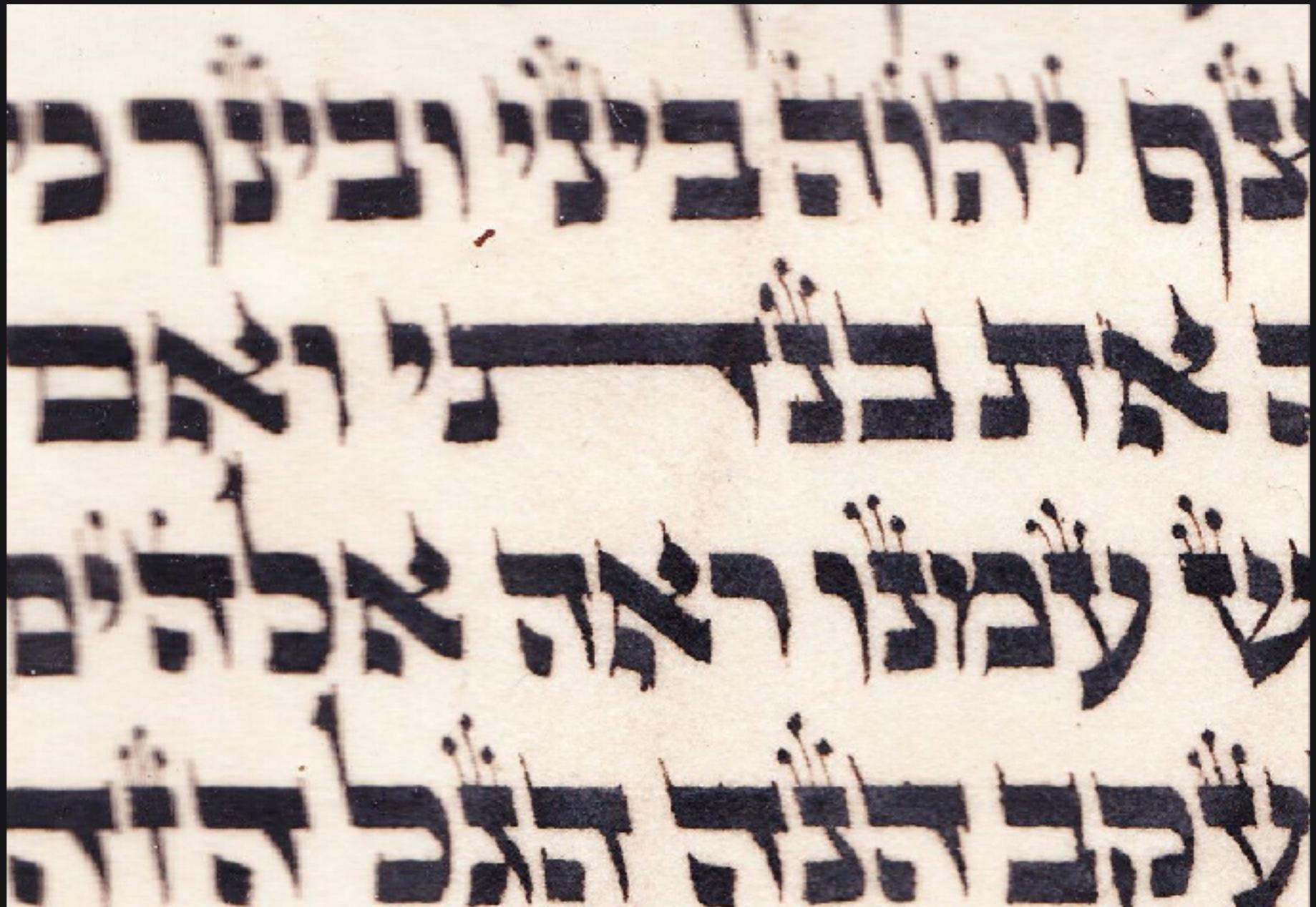
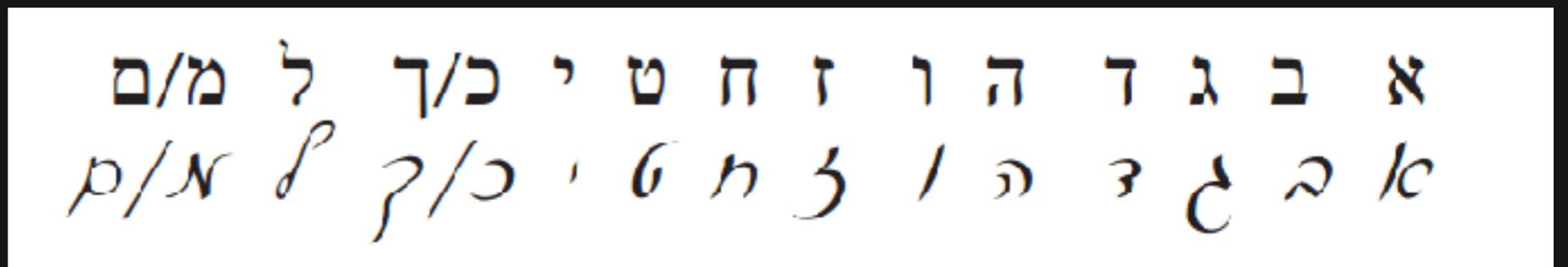
Written Language



Motivation & History

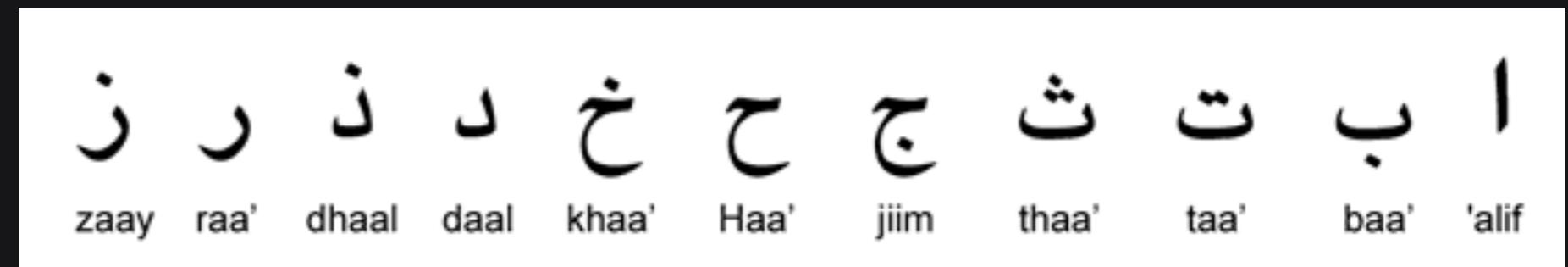
The Art of the Font by Ayal Spitz

Hebrew



Motivation & History

The Art of the Font by Ayal Spitz



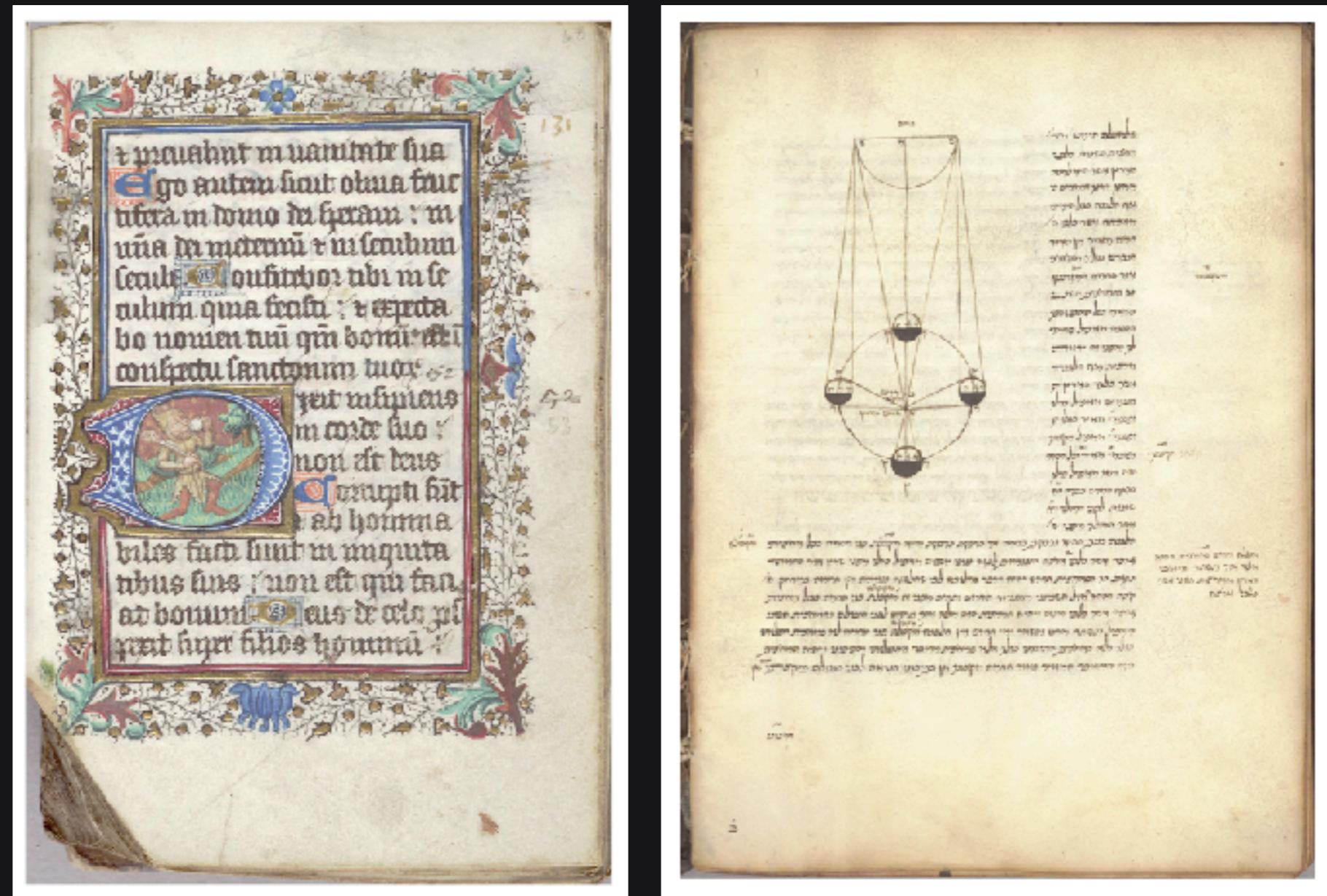
Arabic



Motivation & History

The Art of the Font by Ayal Spitz

Books



Woodblock Printing

世傳富春山居圖為黃子久
畫卷之冠昨年得其所為
山居圖者有董香光鑒跋
時方謂富春圖別為一卷屢



Motivation & History

The Art of the Font by Ayal Spitz

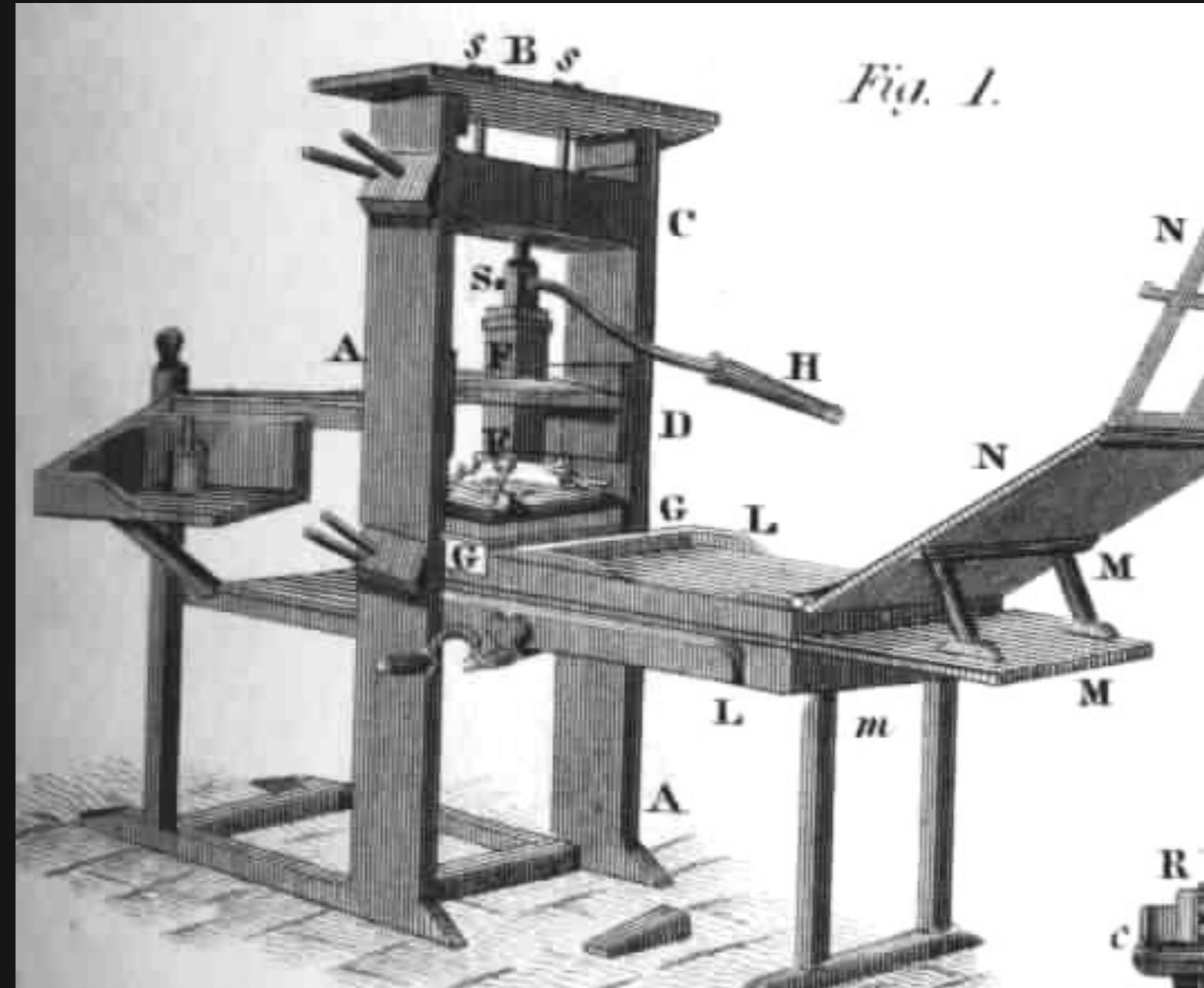
Movable Type



Motivation & History

The Art of the Font by Ayal Spitz

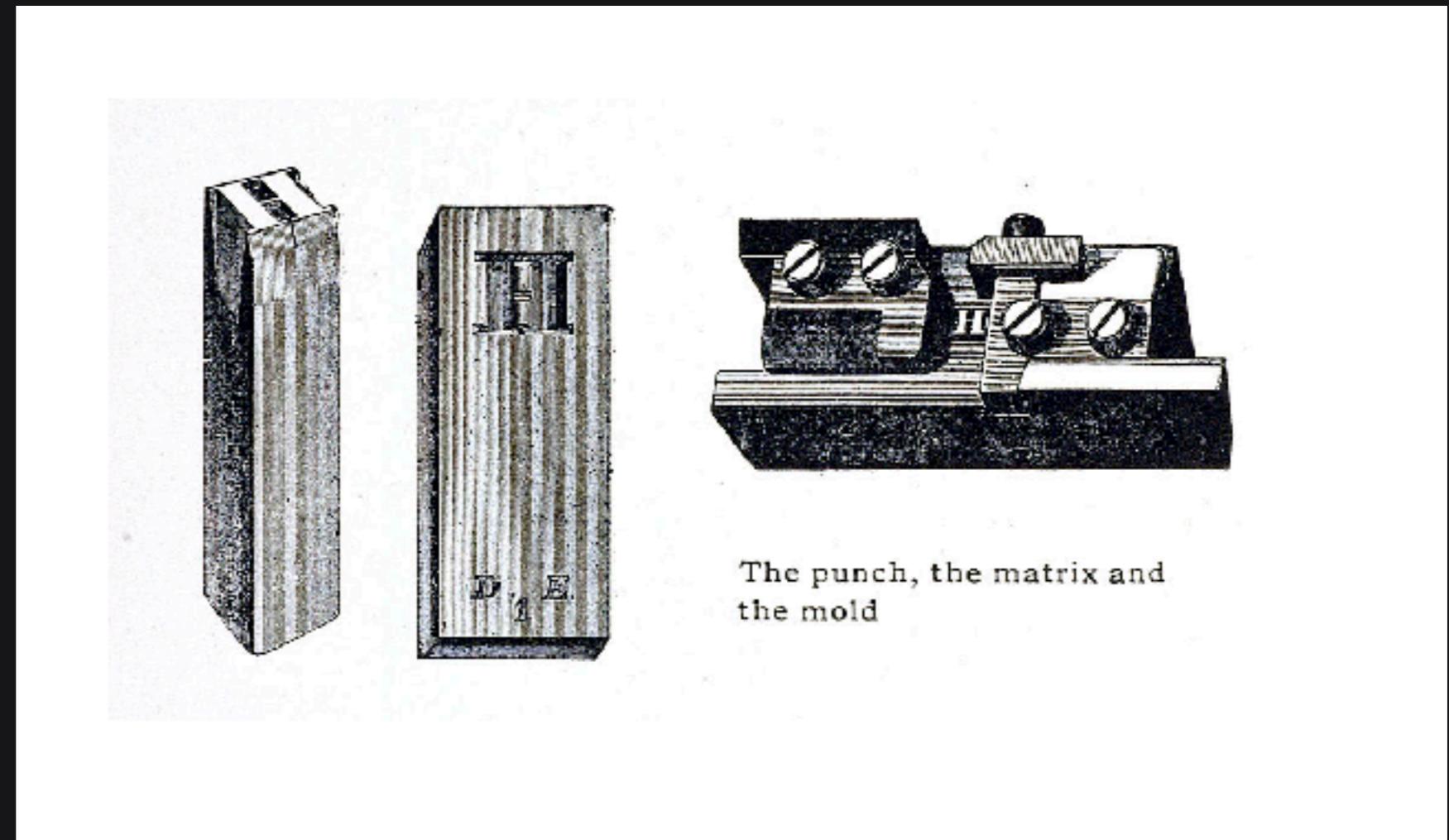
Gutenberg Press



Motivation & History

The Art of the Font by Ayal Spitz

Gutenberg Press

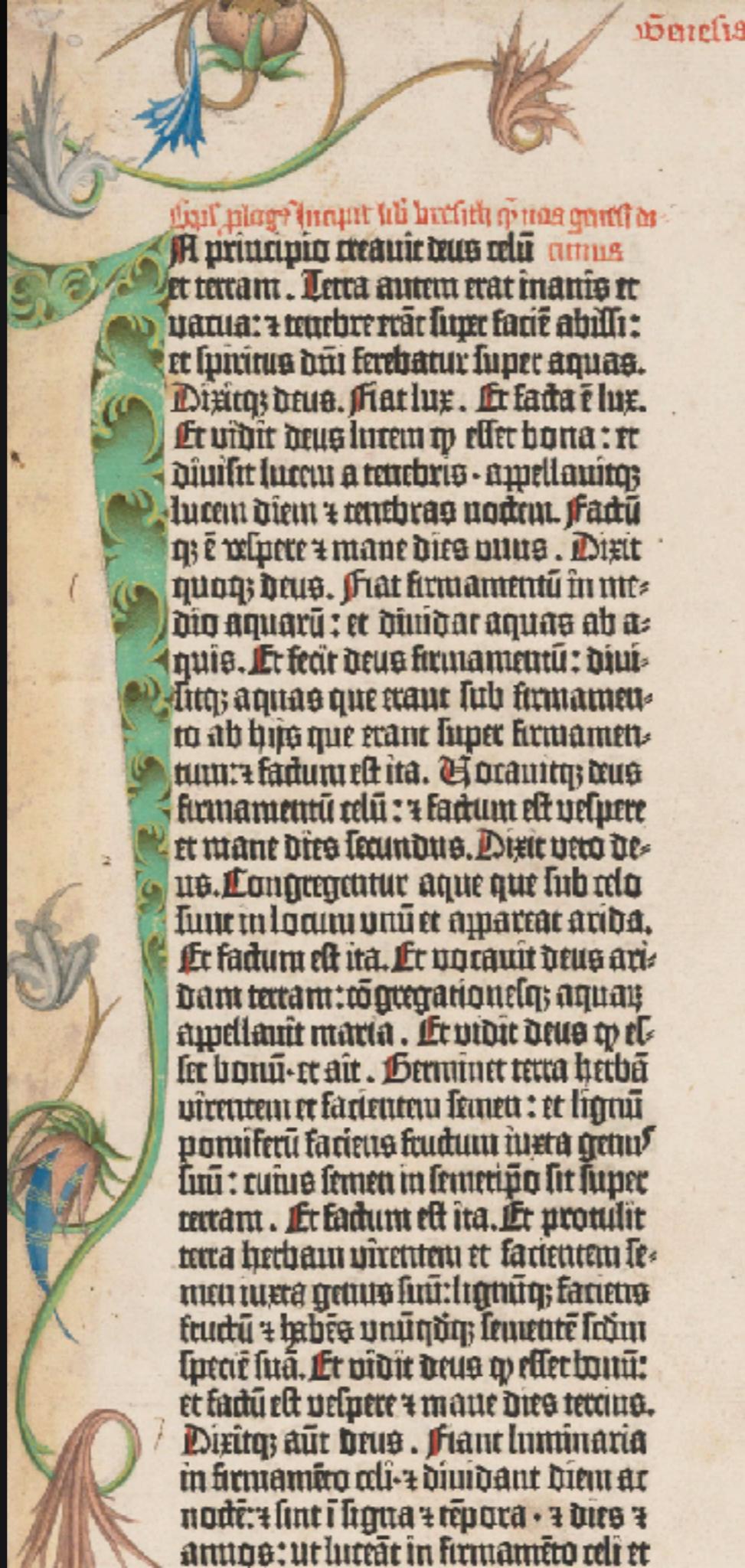


The punch, the matrix and
the mold

Motivation & History

The Art of the Font by Ayal Spitz

Gutenberg Bible



peſſent diem ac nōctē: et dividat lucem
ac tenebras. Et vidit deus q̄ esset bonū:
et factū ē vespere et mane dies quartus.
Dixit etiam deus. Producant aque
reptile animē viventis et volantile sup
terram: sub firmamento celī. Creauitq;
deus rē grandia: et omnē animā vi-
uentē atq; notabilēm quā produxe-
rant aque in species suas: et om̄e vo-
lantile secundū genus suū. Et vidit de-
us q̄ esset bonū: benedixitq; ei dicens.
Eccōte et multiplicamini: et replete a-
quas maris: auesq; multiplicetur
super terram. Et factū ē vespere et mane
dies quintus. Dixit quoq; deus. Pro-
ducat terra animā viventem in gette-
re suo: iumenta et reptilia: et bestias ter-
re secundū species suas. Factū ē ita. Et
fecit deus bestias terre iuxta species su-
as: iumenta et om̄ne reptile terre in ge-
nere sunt. Et vidit deus q̄ esset bonū:
et ait. Faciam h̄ominē ad ymaginē et
similitudinē nostrā: et p̄fici p̄scib⁹ maris.
et volatilib⁹ celī: et bestijs unūscop⁹ terre:
om̄iq; reptili qđ mouet i terra. Et crea-
uit deus h̄ominē ad ymaginē et simi-
litudinē suam: ad ymaginem dei crea-
uit illū: masculū et feminā creauit eos.
Benedixitq; illi deus: et ait. Eccōte
et multiplicamini et replete terram: et
subicite eam: et dominamini p̄scibus
maris: et volatilib⁹ celī: et unūscop⁹
animātibus que mouentur sup terrā.
Dixitq; deus. Ecce dedi vobis omnē
herbam afferentem semen sup terram.
et unūsa liqua que habet i seueripis
fructū generis h̄uic: ut sint vobis i sc̄a-

Fonts & Apple

03

Fonts & Apple

The Art of the Font by Ayal Spitz

Apple II

DOS



Fonts & Apple

The Art of the Font by Ayal Spitz

Mac Fonts

Mac OS 1 - 9



Fonts & Apple

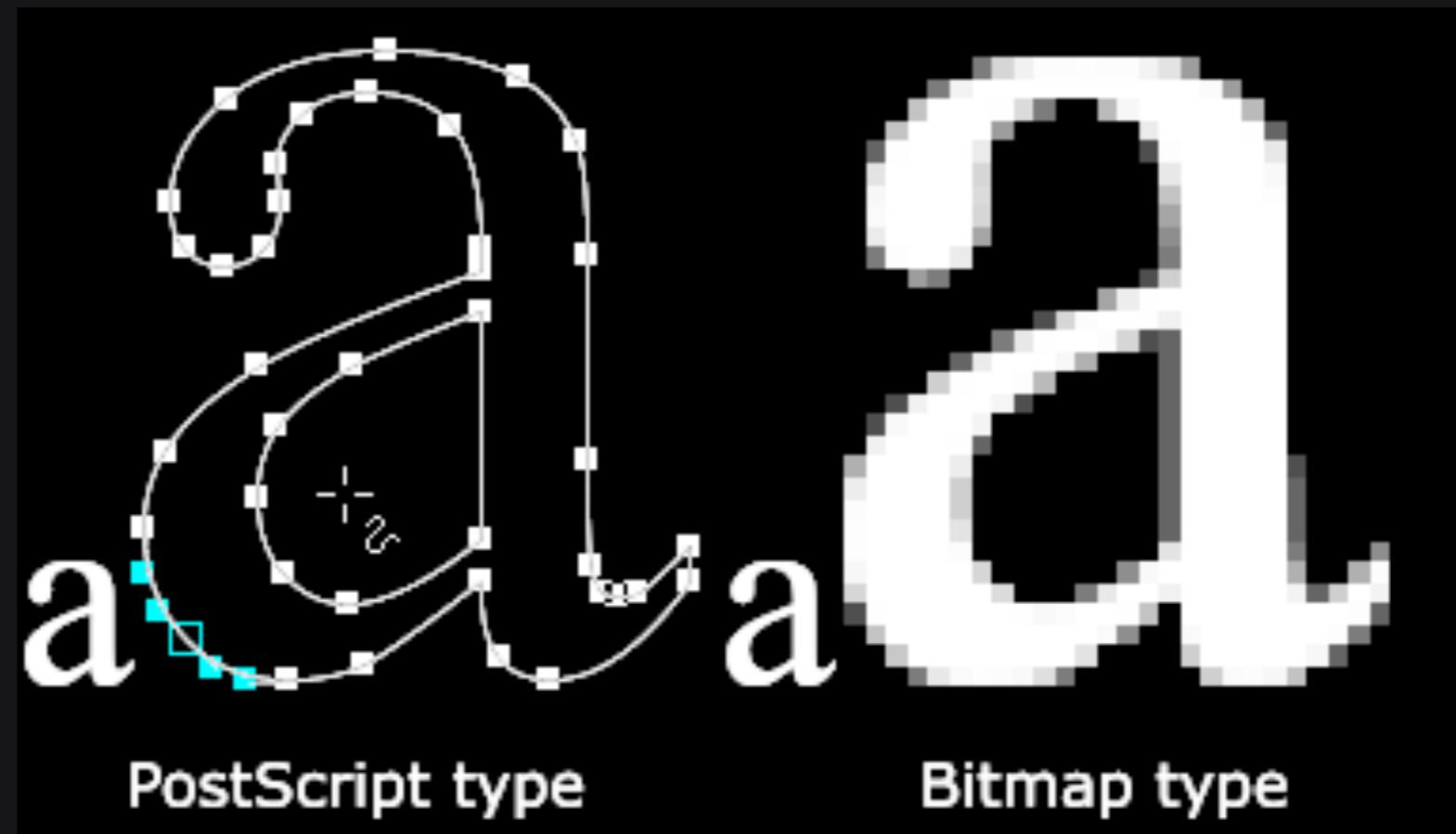
The Art of the Font by Ayal Spitz

Adobe PostScript



Outline Font

Type 1



Fonts & Apple

The Art of the Font by Ayal Spitz

TrueType



Fonts & Apple

The Art of the Font by Ayal Spitz

OpenType



Fonts & Apple

The Art of the Font by Ayal Spitz

Apple Advanced Typography



Fonts & Apple

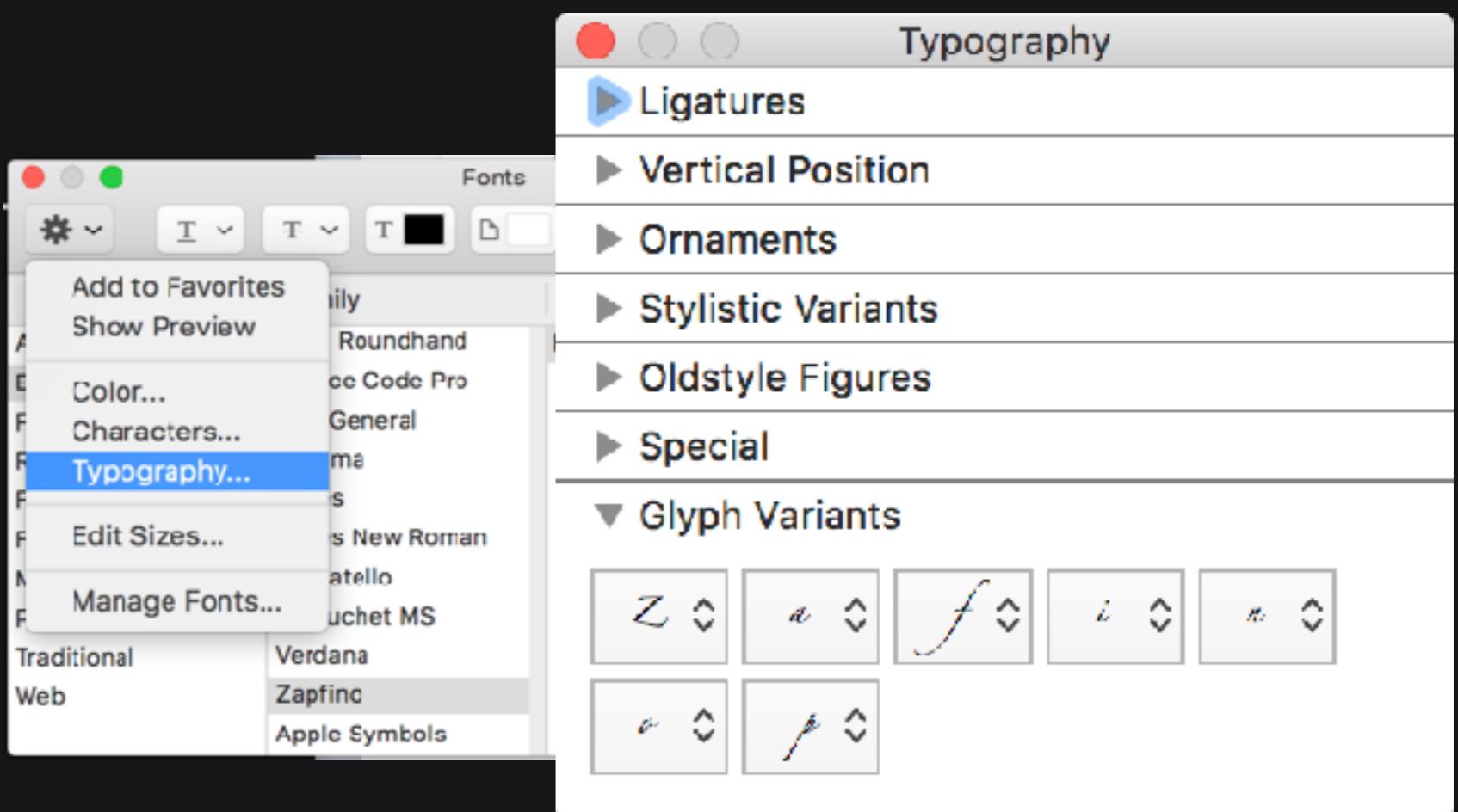
The Art of the Font by Ayal Spitz

Apple Advanced Typography

Fonts & Apple

The Art of the Font by Ayal Spitz

Apple Advanced Typography



CoreText

04

CoreText

The Art of the Font by Ayal Spitz

CoreText

per iOS 7

UILabel

UITextField

UITextView

UIWebView

String Drawing

WebKit

Core Text

Core Graphics

CoreText

The Art of the Font by Ayal Spitz

```
extension Character {
    public func glyph(font: CTFont) -> CGGlyph? {
        let c = unicodeScalars[unicodeScalars.startIndex].utf16
        var chars = Array(c)
        var glyphs = Array<CGGlyph>(repeating: 0, count: 1)
        CTFontGetGlyphsForCharacters(font, &chars, &glyphs, 1)

        return glyphs.count > 0 ? glyphs[0] : nil
    }
}

extension CTFont {
    func cgPath(for char: Character, affineTransform: CGAffineTransform? = nil)
-> CGPath? {
        guard let glyph = char.glyph(font: self) else { return nil }
        if affineTransform != nil {
            var localAffineTransform = affineTransform!
            return CTFontCreatePathForGlyph(self, glyph, &localAffineTransform)
        } else {
            return CTFontCreatePathForGlyph(self, glyph, nil)
        }
    }
}
```

CoreText

The Art of the Font by Ayal Spitz

```
extension UIFont {
    public func cgPath(for char: Character, affineTransform: CGAffineTransform? = nil) -> CGPath? {
        return (self as CTFont).cgPath(for: char, affineTransform: affineTransform)
    }

    public func uiBezierPath(for char: Character) -> UIBezierPath? {
        let flipVertical = CGAffineTransform(a: 1.0, b: 0.0, c: 0.0, d: -1.0, tx: 0.0, ty: lineHeight)
        guard let path = cgPath(for: char, affineTransform: flipVertical) else { return nil }
        return UIBezierPath(CGPath: path)
    }

    public func image(for char: Character, with color: UIColor = .black, at scale: CGFloat = UIScreen.main.scale) -> UIImage? {
        guard let bezierPath = uiBezierPath(for: char) else { return nil }
        let letterSize = "\u{char}".size(withAttributes: [NSAttributedString.Key.font: self])

        UIGraphicsBeginImageContextWithOptions(letterSize, false, UIScreen.main.scale)
        color.setFill()
        bezierPath.apply(CGAffineTransform(translationX: 0, y: descender))
        bezierPath.fill()
        let image = UIGraphicsGetImageFromCurrentImageContext()
        UIGraphicsEndImageContext()
        return image
    }
}
```

CoreText

The Art of the Font by Ayal Spitz

4:23



**Four score and seven years ago our fa
thers brought forth on this continent
, a new nation**

4:25



**Four score and seven years ago our fa
thers brought forth on this continent
, a new nation**

TextKit

05

TextKit

The Art of the Font by Ayal Spitz

TextKit

iOS 7

UILabel

UITextField

UITextView

UIWebView

TextKit

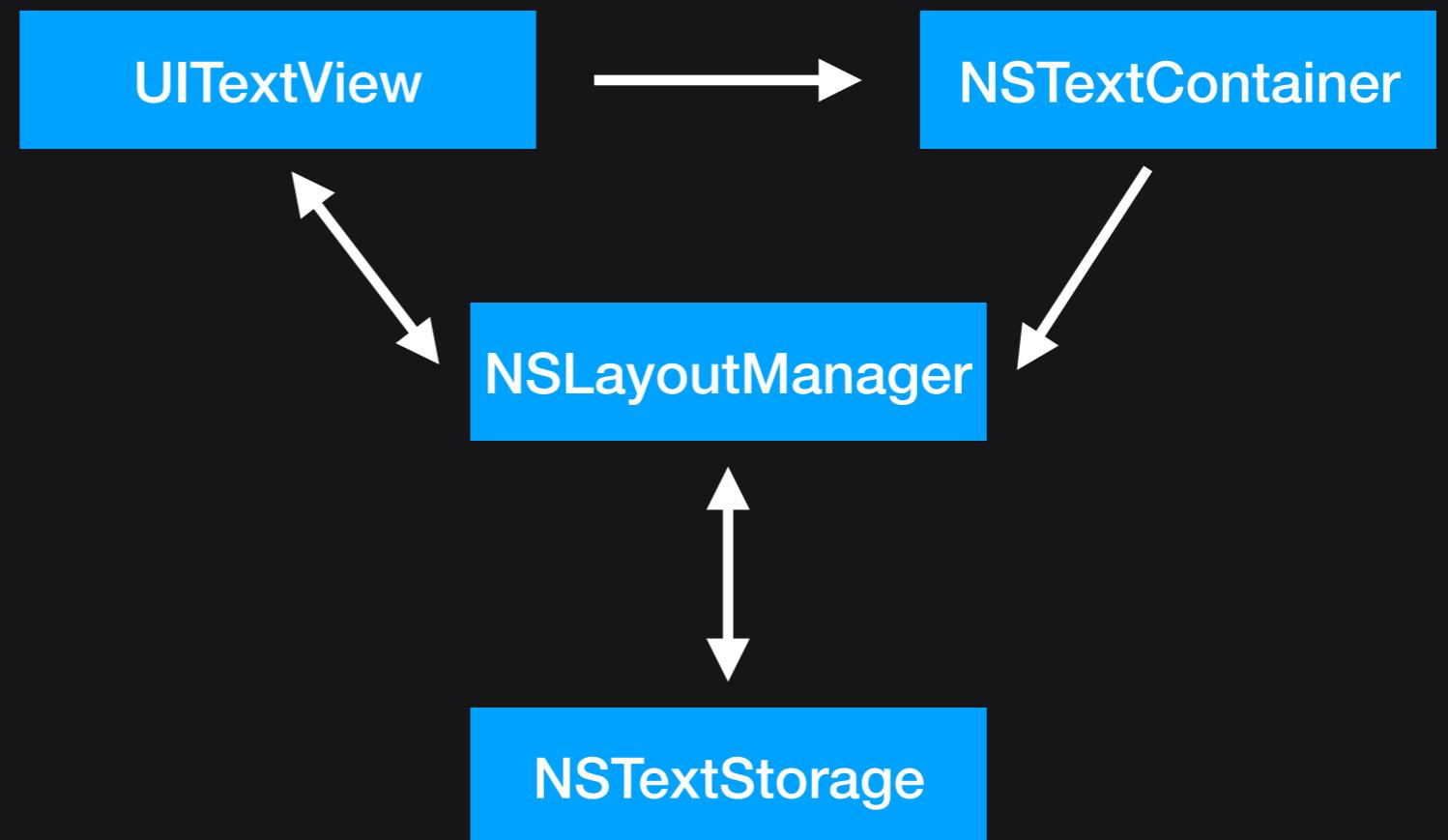
WebKit

Core Text

Core Graphics

TextKit

iOS 7



TextKit

The Art of the Font by Ayal Spitz

```
import UIKit

class CustomTextStorage: NSTextStorage {
    let backingStore = NSMutableAttributedString()

    override var string: String { return backingStore.string }

    override func attributes(at location: Int, effectiveRange range: NSRangePointer?) -> [NSAttributedStringKey : Any] {
        return backingStore.attributes(at: location, effectiveRange: range)
    }

    override func replaceCharacters(in range: NSRange, with str: String) {
        beginEditing()
        backingStore.replaceCharacters(in: range, with: str)
        edited([.editedAttributes, .editedCharacters], range: range,
               changeInLength: str.count - range.length)
        endEditing()
    }

    override func setAttributes(_ attrs: [NSAttributedStringKey : Any]?, range: NSRange) {
        beginEditing()
        backingStore.setAttributes(attrs, range: range)
        edited(.editedAttributes, range: range, changeInLength: 0)
        endEditing()
    }

    override func processEditing() {
        super.processEditing()
    }
}
```

```
import UIKit

class CustomTextContainer: NSTextContainer {
    override var isSimpleRectangularTextContainer: Bool { return false}

    override func lineFragmentRect(forProposedRect proposedRect: CGRect,
        at characterIndex: Int, writingDirection baseWritingDirection: NSWritingDirection,
        remaining remainingRect: UnsafeMutablePointer<CGRect>?) -> CGRect {

        var result = super.lineFragmentRect(forProposedRect:proposedRect,
            at:characterIndex, writingDirection:baseWritingDirection, remaining:remainingRect)

        let r = self.size.height / 2.0
        let y = r - result.origin.y
        let theta = asin(y/r)
        let x = r * cos(theta)
        let offset = self.size.width / 2.0 - r
        result.origin.x = r-x+offset
        result.size.width = 2*x

        return result
    }
}
```

TextKit

The Art of the Font by Ayal Spitz

```
import UIKit

class ViewController: UIViewController {
    var textView: UITextView!
    let textStorage = CustomTextStorage()

    override func viewDidLoad() {
        super.viewDidLoad()

        let textContainerSize = CGSize(width: view.bounds.width, height: view.bounds.width)
        let textContainer = CustomTextContainer(size: textContainerSize)
        textContainer.widthTracksTextView = true

        let layoutManager = NSLayoutManager()
        layoutManager.addTextContainer(textContainer)

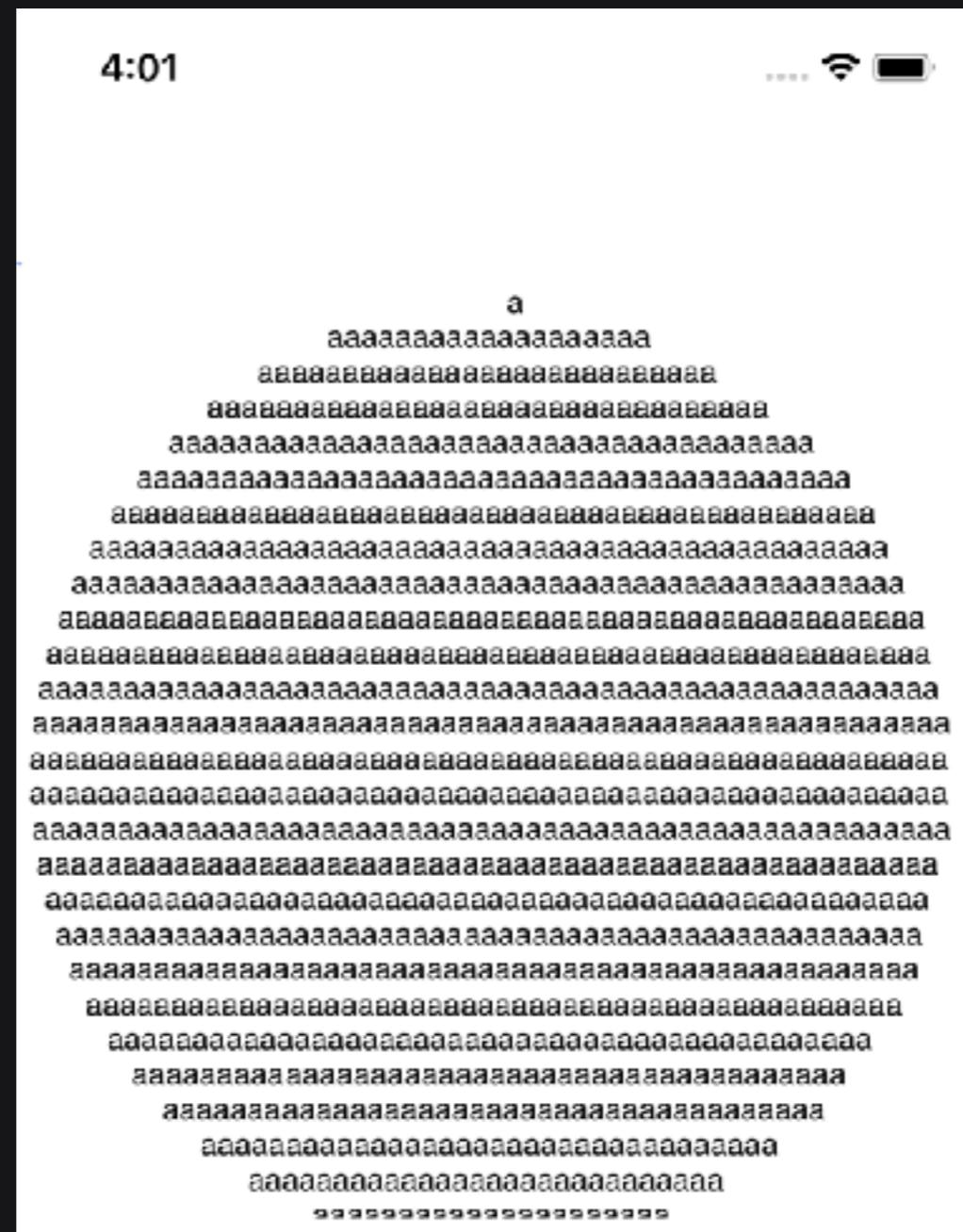
        textStorage.addLayoutManager(layoutManager)

        let frame = CGRect(origin: CGPoint(x: 0.0, y: 100.0), size: textContainerSize)
        textView = UITextView(frame: frame, textContainer: textContainer)

        view.addSubview(textView)
    }
}
```

TextKit

The Art of the Font by Ayal Spitz

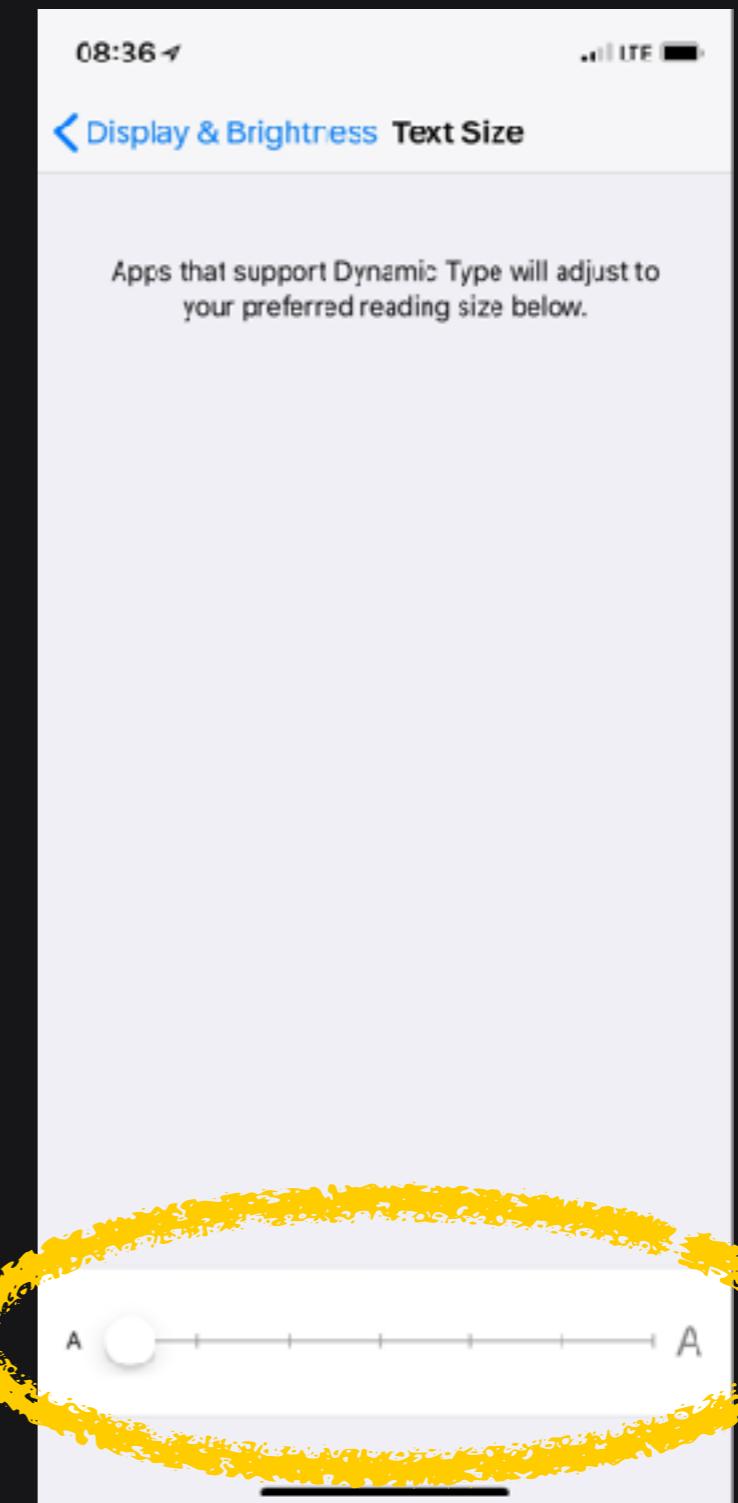


Dynamic Type

06

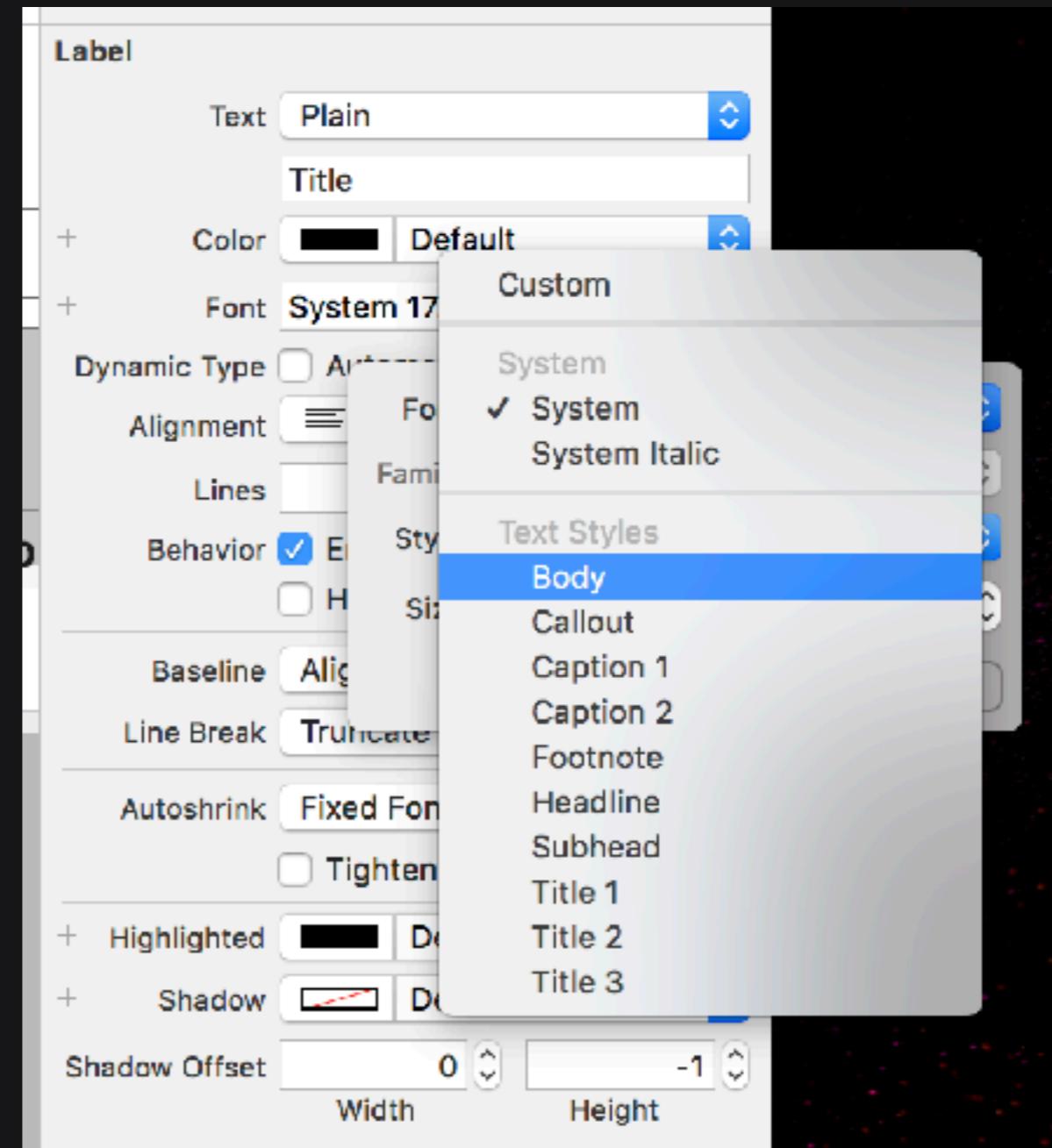
Dynamic Type

- Introduced in iOS 7
- Hidden away in **Settings** →
Display & Brightness is the **Text Size** preference (as of iOS 11)



Dynamic Type

Interface Builder



Dynamic Type

The Art of the Font by Ayal Spitz

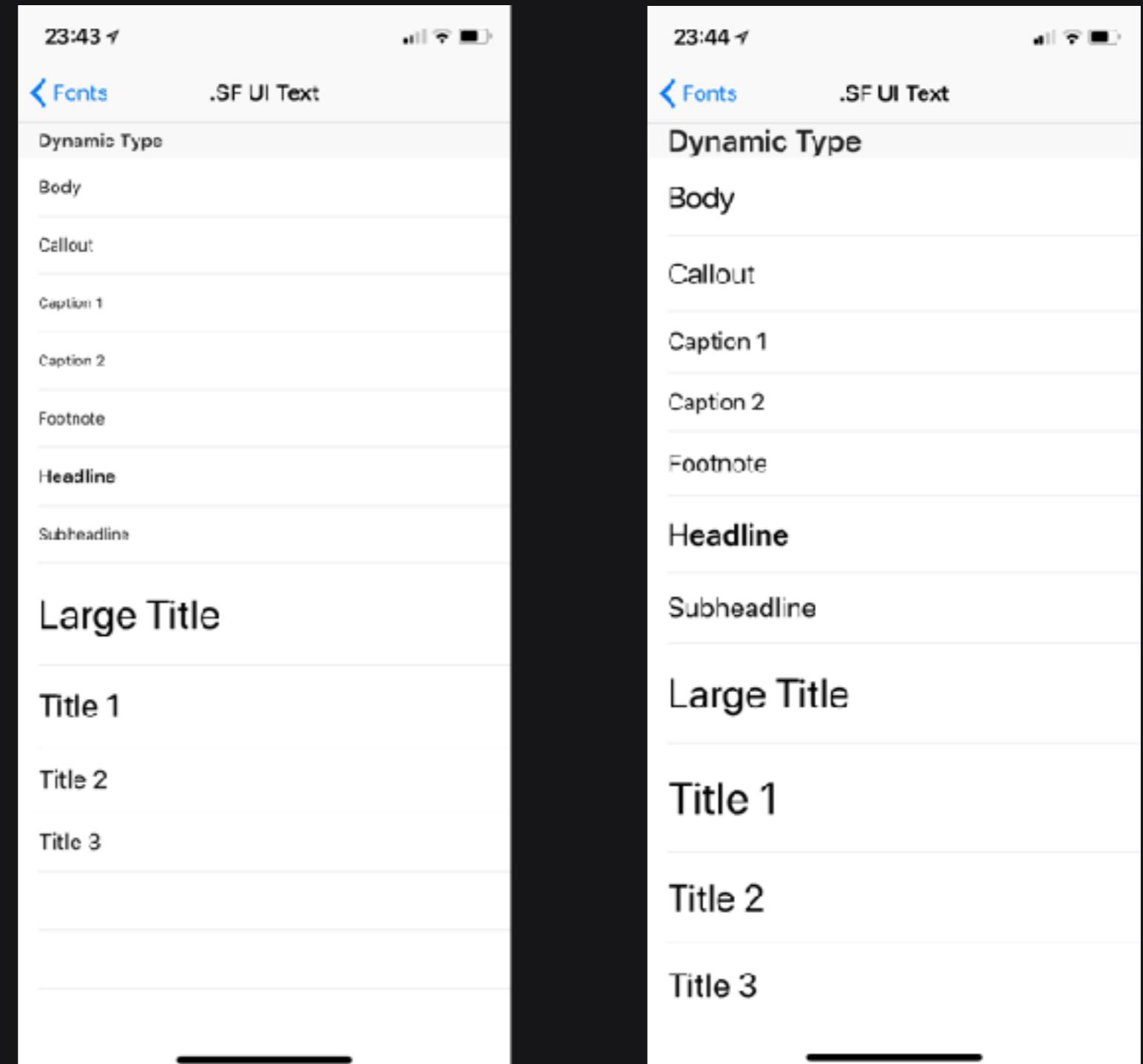
```
UIFont.preferredFont(forTextStyle: style)
```

Dynamic Type

The Art of the Font by Ayal Spitz

Dynamic Type

Preferred Font



Dynamic Type

The Art of the Font by Ayal Spitz

```
UIFont.preferredFont(forTextStyle: style)
```

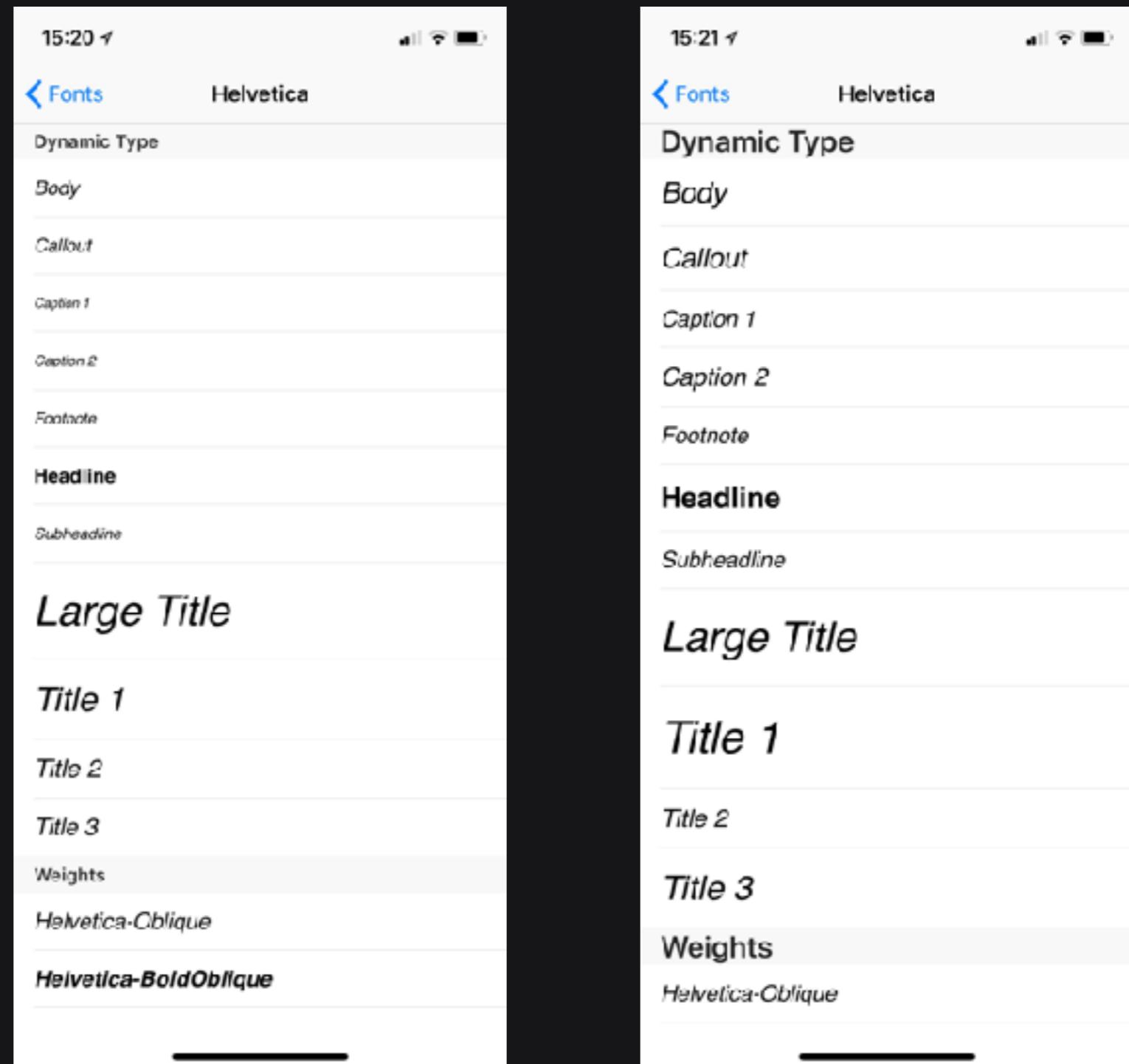
```
let font = UIFont(familyName: fontName, size: defaultSize)
let fontMetrics = UIFontMetrics(forTextStyle: style)
let scaledFont = fontMetrics.scaledFont(for: font)
```

Dynamic Type

The Art of the Font by Ayal Spitz

Dynamic Type

Helvetica



Dynamic Type

The Art of the Font by Ayal Spitz

```
UIFont.preferredFont(forTextStyle: style)

let font = UIFont(familyName: fontName, size: defaultSize)
let fontMetrics = UIFontMetrics(forTextStyle: style)
let scaledFont = fontMetrics.scaledFont(for: font)

let scaledValue = fontMetrics.scaledValue(for: 17.0)
```

Tips

- UIKit elements already make use of Dynamic Type
- Use AutoLayout
- Don't forget to set `adjustsFontForContentSizeCategory`
- Test Frequently
- <https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/typography/>

Closing Thoughts

07

Dynamic Type

The Art of the Font by Ayal Spitz

- Standing on the shoulders of giants
- Flexibility and freedom
- Don't be afraid to follow the white rabbit

Thank You!