

Code Golfing in Swift



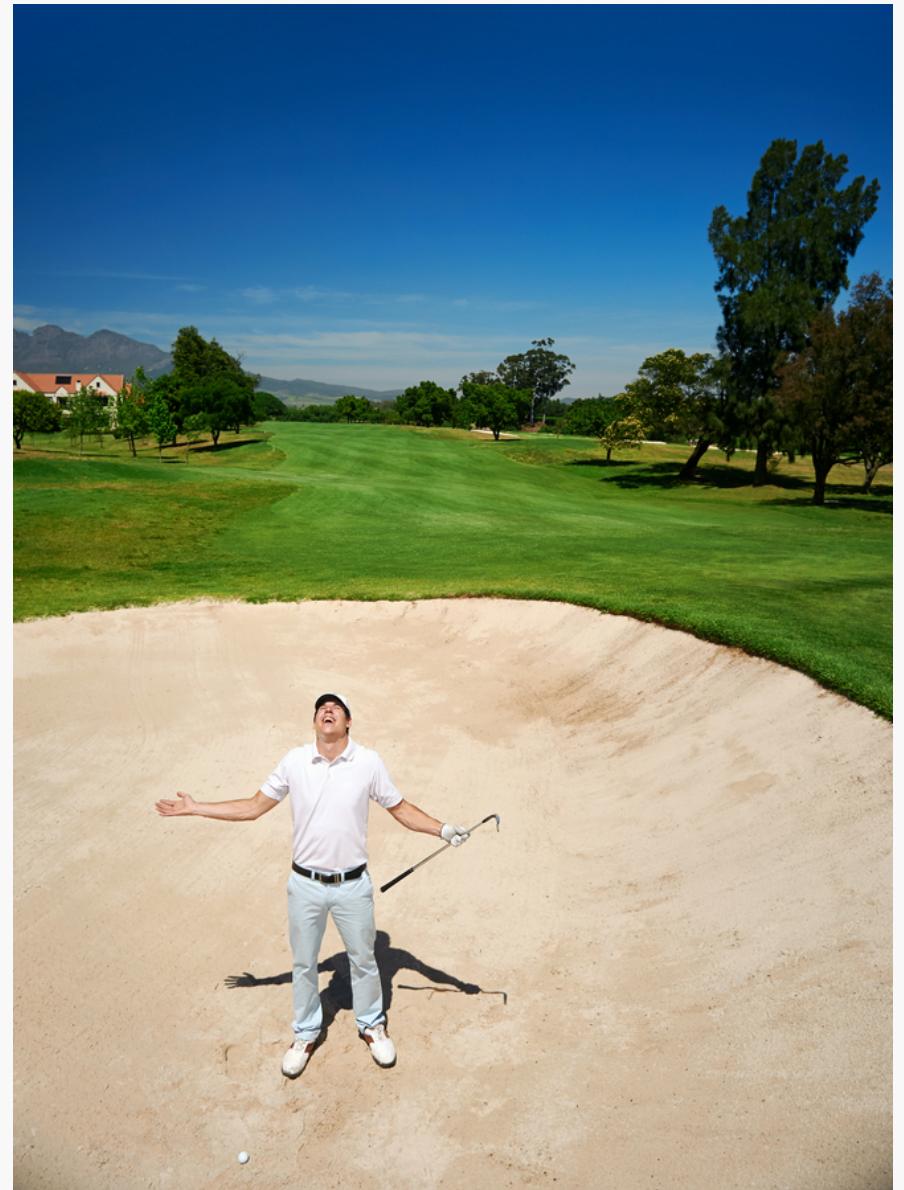
@jasonalexzurita

Goals

- What is Code Golf?
- Advantages of writing less code.
- Tips for writing less code.

Code Golf





**Code golf ... participants strive
to achieve the shortest possible
source code that implements a
certain algorithm.**

https://en.wikipedia.org/wiki/Code_golf

Evolution

Evolution

Examples



Evolution / Examples

```
perl -E  
'say"A"x46422,"BDRDAwMQFFTCBUT1  
JJVE8gU1BFQ0IGSUNBVEIPTg","A"x54  
, "Ew", "A"x2634, "/0NEMDAXAQ", "A"x27  
21, "BAAAAAYQ", "A"x30, "SVVVqogAAAA  
AAAEAF", "A"x2676, "LMBaACgB76gfbg  
TAM0Qv8D4uYAI86qqgcc+AXP45GA8S  
HIRPFB3DTeYSEhyBSwCa8CwicMB3rS  
GtkDNFSYwJHvc68MA", "VapVqlWq"x3  
30'|base64 -D>cd.iso
```

11:45 PM - 14 Jun 2018



alokmenghrajani @alokmenghrajani · 15 Jun 2018

A bootable CD with a retro game which fits in a tweet. [#x86](#) [#asm](#) [#codegolf](#) [#retrogaming](#)

Evolution / Examples

```
// 560 chars - Original reduced shader by Greg Rostami
/**/
vec3 R = iResolution,c;
float T = iDate.w, p = 3.14, t = p+p, a, f = 4.;

vec4 s(float X, float Y, vec2 U)
{
    float w = Y - X, u = (U.x - X)/w;
    vec3 c;
    c = mix(vec3 (.5, .9, 1),vec3(1, .6, .1), u);
    c *= w / sqrt( (1. - cos(t/f)) / 8. );
    c *= smoothstep(.05, .1, u) * smoothstep(.95, .9, u) + .5;
    U.y += T * -.9;
    c *= sign(sin(u * p * f) * cos(U.y * 16.)) / 8. + .7;
    return vec4(c, float((u > 0. && u < 1.) && X < Y));
}

void mainImage(out vec4 o,vec2 u)
{
    vec2 U, w = u / R.y;
```

Evolution / Examples

```
// 560 chars - Original reduced shader by Greg Rostami
/**/
vec3 R = iResolution.c;
float T = iDate.w, p = 3.14, t = p+p, a, f = 4.;

vec4 s(float X, float Y, vec2 U)
{
    float w = Y - X, u = (U.x - X)/w;
    vec3 c;
    c = mix(vec3 (.5, .9, 1),vec3(1, .6, .1), u);
    c *= w / sqrt( 1. - cos(t/f) ) / 8. ;
    c *= smoothstep(.05, .1, u) * smoothstep(.95, .9, u) + .5;
    U.y += T * -.9;
    c *= sign(sin(u * p * f) * cos(U.y * 16.)) / 8. + .7;
    return vec4(c, float((u > 0. && u < 1. ) && X < Y));
}

void mainImage(out vec4 o,vec2 u)
{
    vec2 U, w = u / R.y;
}
```

```
// 349 chars - DjinnKahn and GregRostami shaved a few more chars ...
/**/
float T = iDate.w, P = 1.57, r, a, w, A, B;

void mainImage(out vec4 o,vec2 U)
{
    r = length(U = (U+U-(O.xy=iResolution.xy)) / O.y * 4.) - 2.8;
    A = sin(B += r > cos(B = mod( sin(a = atan(U.y, U.x) - T) * sin(T) * P * 2.+ a + T+P
        - P*1.5) ? P : 0.));
    w = cos(B)- A;
    B = (r - A)/w;
    O = (smoothstep(.45, .4, A = abs(B-.5)) + .5)
        * (vec4(.5, .9, .95, P) - vec4(-.45, .3, .85, P) * B)
        * w
        * (.5 + sign(s
        * step( A, .5
        */
        // Master Fabrice reduced the shader to 316 chars!!
        /**
void mainImage(out vec4 O, vec2 U)
{
```

```
    float
        T = iTIME,
        r = length(U += U-(O.xy=iResolution.xy)) / O.y * 4. - 2.8,
        a = atan(U.y, U.x),
        B = mod( a+T + sin(a-=T) * sin(T) * 3.14, 1.57 ) - 2.;

        r -= T = sin( B += r > cos(B) ? 1.6 : 0. );
        B = cos(B)- T;

        O = vec4(5, 9, 9, 0)*B - vec4(-5, 3, 9, 0) * r;
        O *= smoothstep(.45, .4, T = abs(r/B-.5) ) + .5;

        O *= T<.5 ? sin(r/B * 13.) * cos(a * 16.) < 0. ? .04 : .05
            : 0.;
```

```
// 382 chars - DjinnKahn showed us the power of the DARK SIDE!
/**/
float T = iDate.w, P = 3.14, r, a, u, v, A, B;

#define s(A, w) ( u = (r - A)/w, v = abs(u-.5), mix(vec4(.5, .9, .95, P), vec4(.95, .6, .1, P), u) * w * w )

void mainImage(out vec4 O,vec2 p)
{
    r = length(p = (p+p-(O.xy=iResolution.xy)) / O.y * 4.) - 2.8;
    a = atan(p.y, p.x) + P - T;
    B = mod( a + T+T + sin(a) * sin(T) * P, P*.5 ) - P*.75;
    A = sin(B);
    B = cos(B);
    O = s(A, (r - A)) + s(B, (-A-B));
}
```

```
// 334 chars - Smallest shader with some cheating. ;)
/**/
float T = iDate.w, P = 1.57, r, a, w, A, B;

void mainImage(out vec4 O,vec2 U)
{
    r = length(u = (u/iResolution.y-.5) * 8.) - 2.8;
    A = sin(B += r > cos(B = mod( sin(a = atan(u.y, u.x) - T) * sin(T) * P * 2.+ a + T+T, P
        - P*1.5) ? P : 0.));
    w = cos(B)- A;
    B = (r - A)/w;
    O = (smoothstep(.45, .4, A = abs(B-.5)) + .5)
        * (vec4(.5, .9, .95, P) - vec4(-.45, .3, .85, P) * B)
```

Evolution / Examples



99.88

60.1 fps

500 x 281



<https://www.shadertoy.com/view/lItXz2>

Evolution / Examples



99.88

60.1 fps

500 x 281



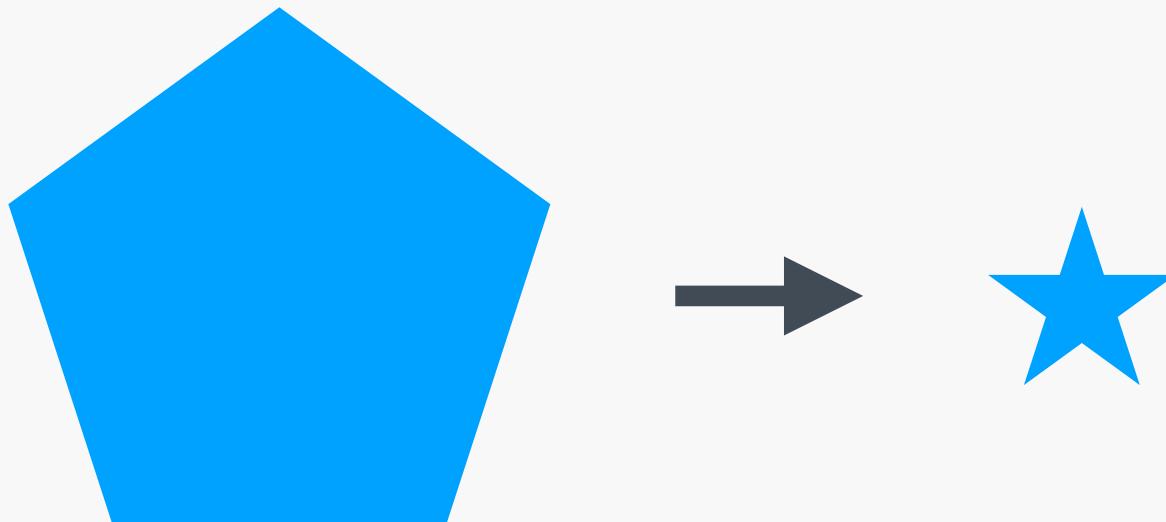
<https://www.shadertoy.com/view/lItXz2>

Evolution

Code Golfing Mindset

Evolution / Code Golfing Mindset

The Transformation

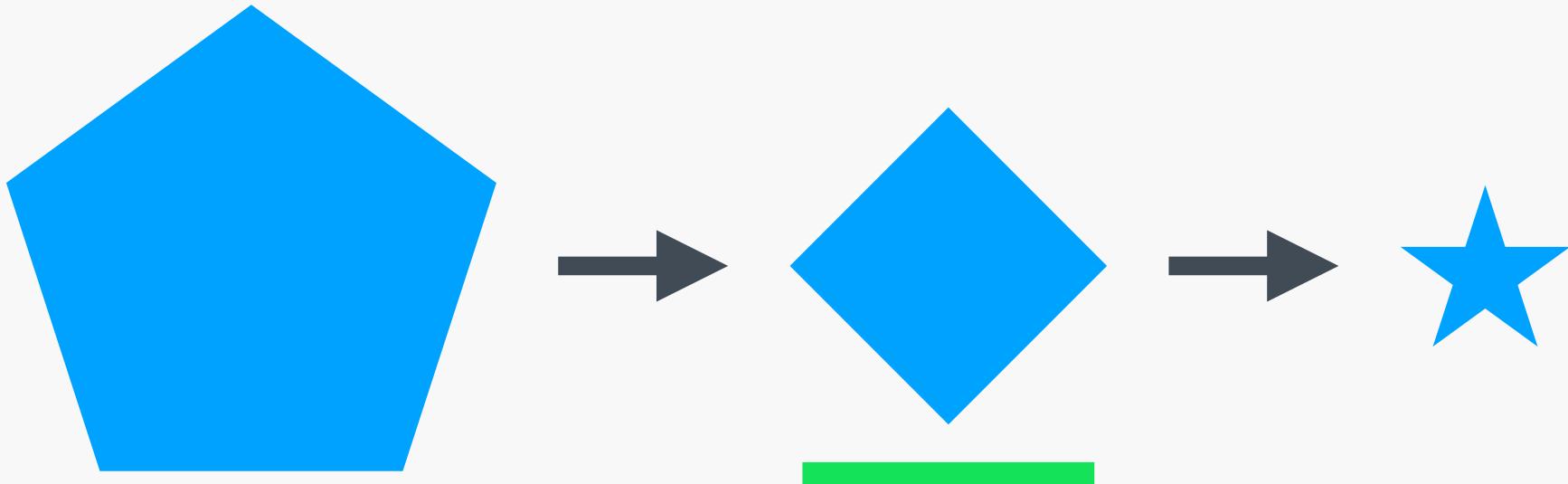


Sides = Complexity

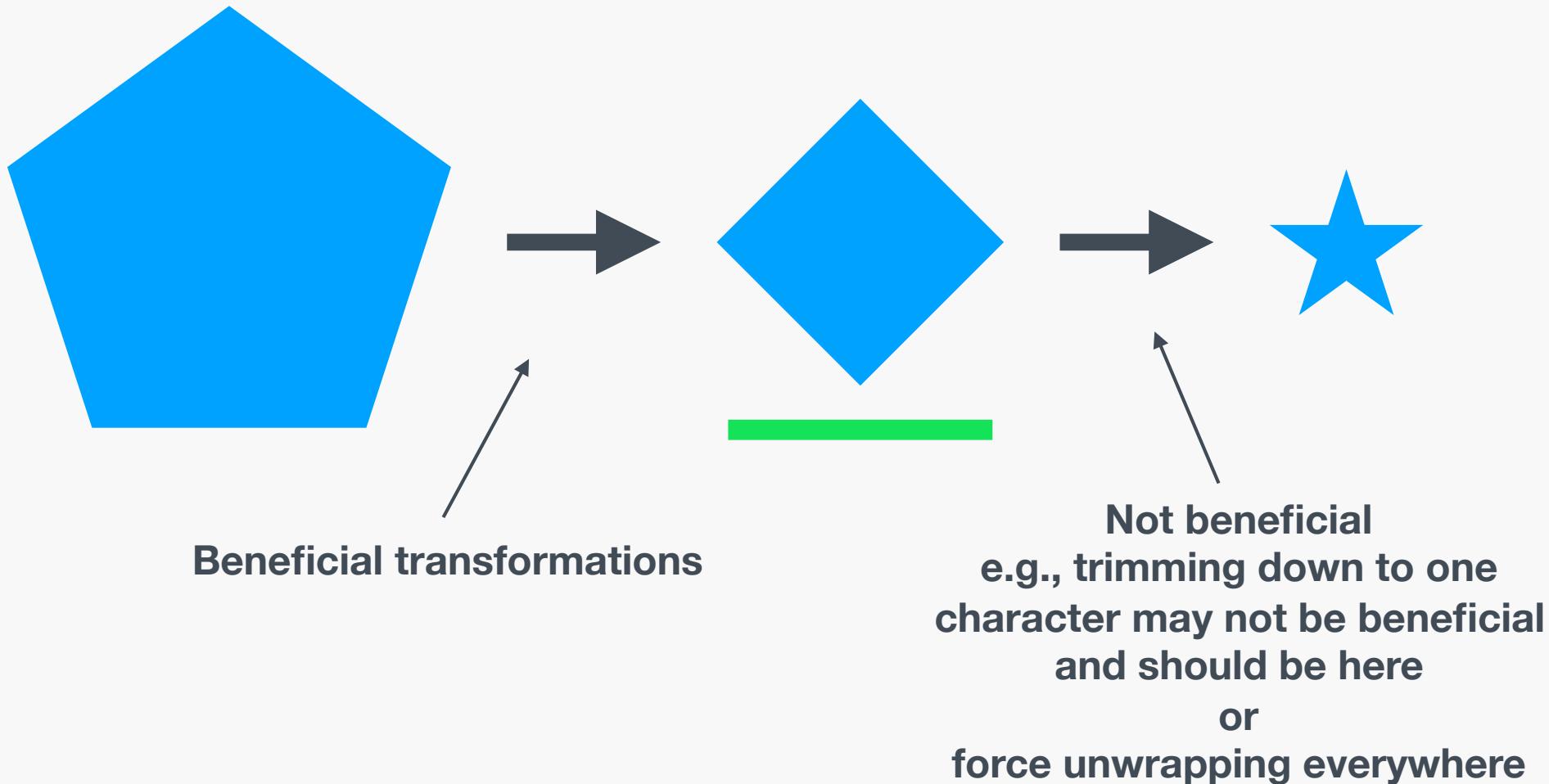
Area = Code golf score

Evolution / Code Golfing Mindset

The Transformation



The Transformation



Context

- Scoring considerations:
 - ◆ Points, tokens, lines of code, characters, bytes, etc.
 - ◆ Do dependencies & tests count? For how much?
 - ◆ Generated code — e.g., Sourcery & Swift generate your boiler plate (GYB).

Benefits



Maintaining Engineering Effectiveness



Robert Lefkowitz (R0ml)

Benefits

Maintaining Engineering Effectiveness



1. *An engineer can maintain ~25,000 lines of code.*
2. *An engineer can write about 1000 debugged lines of code (LOC) per month.*
 - *R0ml's laws (@r0ml)*

Benefits

Maintaining Engineering Effectiveness



1. *An engineer can maintain ~25,000 lines of code.*
 2. *An engineer can write about 1000 debugged lines of code (LOC) per month.*
 - *R0ml's laws (@r0ml)*
- Therefore after ~2 years, an engineer has created enough code to stall their career maintaining it.

Benefits

Maintaining Engineering Effectiveness



1. *An engineer can maintain ~25,000 lines of code.*
 2. *An engineer can write about 1000 debugged lines of code (LOC) per month.*
 - *R0ml's laws (@r0ml)*
- Therefore after ~2 years, an engineer has created enough code to stall their career maintaining it.
 - Related to the ~2 year turnover in the tech industry? 🤔

Benefits

Maintaining Engineering Effectiveness



1. *An engineer can maintain ~25,000 lines of code.*
 2. *An engineer can write about 1000 debugged lines of code (LOC) per month.*
 - *R0ml's laws (@r0ml)*
- Therefore after ~2 years, an engineer has created enough code to stall their career maintaining it.
 - Related to the ~2 year turnover in the tech industry? 🤔
 - Fewer LOC = less to maintain and therefore more new feature development.

Benefits

Code Readability

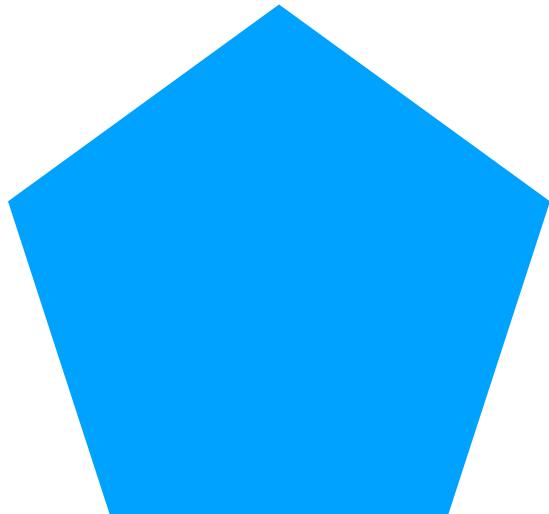
Benefits / Code Readability

```
internal func paymentInputMethodTextV1(for presentationType: ReaderInputOptions) -> String {  
  
    var paymentMethods = ""  
  
    if presentationType.contains(.swipeCard) {  
        paymentMethods.append("Swipe")  
    }  
  
    if presentationType.contains(.insertCard) {  
        if paymentMethods.isEmpty {  
            paymentMethods.append("Dip")  
        } else {  
            paymentMethods.append(", dip")  
        }  
    }  
  
    if presentationType.contains(.tapCard) {  
        if paymentMethods.isEmpty {  
            paymentMethods.append("Tap card")  
        } else {  
            paymentMethods.append(", tap card")  
        }  
    }  
  
    return paymentMethods  
}
```

Benefits / Code Readability

```
internal func paymentInputMethodTextV1(for presentationType: ReaderInputOptions) -> String {  
  
    var paymentMethods = ""  
  
    if presentationType.contains(.swipeCard) {  
        paymentMethods.append("Swipe")  
    }  
  
    if presentationType.contains(.insertCard) {  
        if paymentMethods.isEmpty {  
            paymentMethods.append("Dip")  
        } else {  
            paymentMethods.append(", dip")  
        }  
    }  
  
    if presentationType.contains(.tapCard) {  
        if paymentMethods.isEmpty {  
            paymentMethods.append("Tap card")  
        } else {  
            paymentMethods.append(", tap card")  
        }  
    }  
  
    return paymentMethods  
}
```

LOC: 21



Benefits / Code Readability

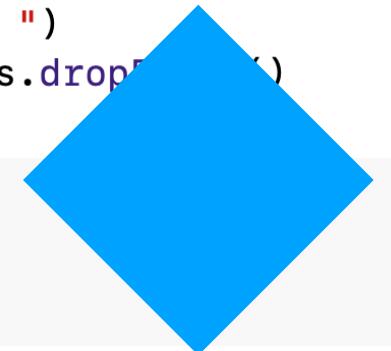
```
internal func paymentInputMethodTextV2(for presentationType: ReaderInputOptions) -> String {  
    let potentialPaymentMethods = [  
        presentationType.contains(.swipeCard) ? "swipe" : nil,  
        presentationType.contains(.insertCard) ? "dip" : nil,  
        presentationType.contains(.tapCard) ? "tap card" : nil,  
    ]  
  
    let filteredPaymentMethods = potentialPaymentMethods.compactMap { $0 }  
  
    let basePaymentMethods = filteredPaymentMethods.joined(separator: ", ")  
    return basePaymentMethods.prefix(1).uppercased() + basePaymentMethods.dropFirst()  
}
```

LOC: 10

Benefits / Code Readability

```
internal func paymentInputMethodTextV2(for presentationType: ReaderInputOptions) -> String {  
    let potentialPaymentMethods = [  
        presentationType.contains(.swipeCard) ? "swipe" : nil,  
        presentationType.contains(.insertCard) ? "dip" : nil,  
        presentationType.contains(.tapCard) ? "tap card" : nil,  
    ]  
  
    let filteredPaymentMethods = potentialPaymentMethods.compactMap { $0 }  
  
    let basePaymentMethods = filteredPaymentMethods.joined(separator: ", ")  
    return basePaymentMethods.prefix(1).uppercased() + basePaymentMethods.dropFirst()  
}
```

LOC: 10



```
internal func paymentInputMethodTextV3(for presentationType: ReaderInputOptions) -> String {  
    let paymentMethodsString = [  
        presentationType.contains(.swipeCard) ? "swipe" : nil,  
        presentationType.contains(.insertCard) ? "dip" : nil,  
        presentationType.contains(.tapCard) ? "tap card" : nil,  
    ].compactMap { $0 }.joined(separator: ", ")  
  
    return paymentMethodsString.prefix(1).uppercased() + paymentMethodsString.dropFirst()  
}
```

LOC: 10*

Benefits / Code Readability

```
internal func paymentInputMethodTextV3(for pt: ReaderInputOptions) -> String {  
    let r = [pt.contains(.swipeCard) ? "swipe" : nil, pt.contains(.insertCard) ? "dip" : nil,  
            pt.contains(.tapCard) ? "tap card" : nil,].compactMap { $0 }.joined(separator: ", ")  
    return r.prefix(1).uppercased() + r.dropFirst()  
}
```

LOC: 5



Benefits / Code Readability

```
internal func paymentInputMethodText(pt: PaymentInputMethod) -> String {  
    let r = [pt.contains(.swipeCard)  
            ? "swipe",  
            pt.contains(.tapCard) ? "tap card",  
            pt.contains(.dipCard) ? "dip",  
            pt.contains(.insertCard) ? "insert card",  
            nil].compactMap { $0 }.joined(separator: ", ")  
    return r.prefix(1).uppercased()  
}
```



Benefits

- Better master your language — Swift, Objective-C, etc.
- Faster compile times
- Lower memory footprint
- Runtime performance

Benefits

- Better master your language — Swift, Objective-C, etc.
- Faster compile times
- Lower memory footprint
- Runtime performance



Tips



Tips

You Define the Game



Language

- The number of lines of code (LOC) a developer can write is constant [1].
- Therefore, writing in a language where you can do more with N lines of code than in another language with the same number is more worthwhile.

Language

- The number of lines of code (LOC) a developer can write is constant [1].
- Therefore, writing in a language where you can do more with N lines of code than in another language with the same number is more worthwhile.

Long vs Short Functions



Joe Groff
@jckarter

Following ▾

A lot of people replying with “just write short functions” need to read number-none.com/blow/john_carmack_on_inlined_code.html...

8:39 AM - 16 Jun 2019

http://number-none.com/blow/john_carmack_on_inlined_code.html

Tips / Long vs Short Functions

----- style B:

```
void MajorFunction( void ) {  
    MinorFunction1();  
    MinorFunction2();  
    MinorFunction3();  
}  
  
void MinorFunction1( void ) {  
}  
  
void MinorFunction2( void ) {  
}  
  
void MinorFunction3( void ) {  
}
```

----- style C:

```
void MajorFunction( void ) {  
    // MinorFunction1  
    // MinorFunction2  
    // MinorFunction3  
}
```

Tips

Dependencies

Tips

Dependencies

- Importing a dependency can mean instant losing.

Tips

Dependencies

- Importing a dependency can mean instant losing.
- Not all dependencies are bad, but you are responsible for maintaining that dependency.

Dependencies

- Importing a dependency can mean instant losing.
- Not all dependencies are bad, but you are responsible for maintaining that dependency.
- Consider copying the core functionality that you need right into your project.

Tips

Dependencies

- Importing a dependency can mean instant losing.
- Not all dependencies are bad, but you are responsible for maintaining that dependency.
- Consider copying the core functionality that you need right into your project.
- Try and implement it yourself!

Higher order functions

Adding up the elements in an array with `for-in` loops:

```
var sum = 0

for item in array{
    sum += item
}
print(sum)
```

can be simplified to:

```
print(array.reduce(0, +))
```

Higher order functions

- map
 - contains
- flatmap
 - partition
- compactMap
 - allSatisfy
- filter
 - removeAll
- reduce
 - forEach
- sort

Adding up the elements in an array with `for-in` loops:

```
var sum = 0  
  
for item in array{  
    sum += item  
}  
print(sum)
```

can be simplified to:

```
print(array.reduce(0, +))
```

Tips

User interface

- Xib & Storyboard (...still code).
- Programmatically.
- SwiftUI .

Tips / User interface / Xib

```
<?xml version="1.0" encoding="UTF-8"?>
<document type="com.apple.InterfaceBuilder3.CocoaTouch.XIB" version="3.0" toolsVersion="14845"
    targetRuntime="iOS.CocoaTouch" propertyAccessControl="none" useAutolayout="YES" useTraitCollections="YES"
    useSafeAreas="YES" colorMatched="YES">
    <device id="retina6_1" orientation="portrait" appearance="light"/>
    <dependencies>
        <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="14799.2"/>
        <capability name="Safe area layout guides" minToolsVersion="9.0"/>
        <capability name="documents saved in the Xcode 8 format" minToolsVersion="8.0"/>
    </dependencies>
    <objects>
        <placeholder placeholderIdentifier="IBFilesOwner" id="-1" userLabel="File's Owner"/>
        <placeholder placeholderIdentifier="IBFirstResponder" id="-2" customClass="UIResponder"/>
        <view contentMode="scaleToFill" id="iN0-13-epB">
            <rect key="frame" x="0.0" y="0.0" width="414" height="896"/>
            <autoresizingMask key="autoresizingMask" widthSizable="YES" heightSizable="YES"/>
            <subviews>
                <label opaque="NO" userInteractionEnabled="NO" contentMode="left" horizontalHuggingPriority="251"
                    verticalHuggingPriority="251" text="I am in a Xib file" textAlignment="center"
                    lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
                    translatesAutoresizingMaskIntoConstraints="NO" id="sfz-hF-cLJ">
                    <rect key="frame" x="147" y="442.5" width="120" height="21"/>
                    <fontDescription key="fontDescription" type="system" pointSize="17"/>
                    <nil key="textColor"/>
                    <nil key="highlightedColor"/>
                </label>
            </subviews>
            <color key="backgroundColor" red="1" green="1" blue="1" alpha="1" colorSpace="custom"
                customColorSpace="sRGB"/>
            <constraints>
                <constraint firstItem="sfz-hF-cLJ" firstAttribute="centerY" secondItem="vUN-kp-3ea"
                    secondAttribute="centerY" id="Dqm-4k-tug"/>
                <constraint firstItem="sfz-hF-cLJ" firstAttribute="centerX" secondItem="vUN-kp-3ea"
                    secondAttribute="centerX" id="dfu-i7-MD9"/>
            </constraints>
            <viewLayoutGuide key="safeArea" id="vUN-kp-3ea"/>
            <point key="canvasLocation" x="139" y="137"/>
        </view>
    </objects>
</document>
```

Tips / User interface / Xib

```
- <?xml version="1.0" encoding="UTF-8" standalone="no"?>
- <document type="com.apple.InterfaceBuilder3.CocoaTouch.XIB" version="3.0" toolsVersion="13142" targetRuntime="iOS.C
+ <?xml version="1.0" encoding="UTF-8"?>
+ <document type="com.apple.InterfaceBuilder3.CocoaTouch.XIB" version="3.0" toolsVersion="14845" targetRuntime="iOS.C
+     <device id="retina6_1" orientation="portrait" appearance="light"/>
     <dependencies>
-         <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="12042"/>
+         <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="14799.2"/>
             <capability name="Safe area layout guides" minToolsVersion="9.0"/>
             <capability name="documents saved in the Xcode 8 format" minToolsVersion="8.0"/>
     </dependencies>
```

Hunk 2 : Lines 10-32

Stage hunk

```
0         <placeholder placeholderIdentifier="IBFilesOwner" id="-1" userLabel="File's Owner"/>
1         <placeholder placeholderIdentifier="IBFirstResponder" id="-2" customClass="UIResponder"/>
2         <view contentMode="scaleToFill" id="iN0-l3-epB">
-             <rect key="frame" x="0.0" y="0.0" width="375" height="667"/>
3             <rect key="frame" x="0.0" y="0.0" width="414" height="896"/>
4             <autoresizingMask key="autoresizingMask" widthSizable="YES" heightSizable="YES"/>
5             <subviews>
6                 <label opaque="NO" userInteractionEnabled="NO" contentMode="left" horizontalHuggingPriority="251"
7                     <rect key="frame" x="147" y="442.5" width="120" height="21"/>
8                     <fontDescription key="fontDescription" type="system" pointSize="17"/>
9                     <nil key="textColor"/>
0                     <nil key="highlightedColor"/>
1                 </label>
2             </subviews>
3             <color key="backgroundColor" red="1" green="1" blue="1" alpha="1" colorSpace="custom" customColorSpace="sRGB"/>
4             <constraints>
5                 <constraint firstItem="sfz-hF-cLJ" firstAttribute="centerY" secondItem="vUN-kp-3ea" secondAttribute="centerX"
6                     <constraint firstItem="sfz-hF-cLJ" firstAttribute="centerX" secondItem="vUN-kp-3ea" secondAttribute="centerY"/>
7             </constraints>
8             <viewLayoutGuide key="safeArea" id="vUN-kp-3ea"/>
9             <point key="canvasLocation" x="139" y="137"/>
0         </view>
1     </objects>
2 </document>
```

Tips / User interface / Programmatically

```
final class TestView: UIView {
    override init(frame: CGRect) {
        super.init(frame: frame)
        let label = UILabel()
        label.text = "👋 World!"
        label.textAlignment = .center
        NSLayoutConstraint.activate([
            label.centerXAnchor.constraint(equalTo: self.centerXAnchor),
            label.centerYAnchor.constraint(equalTo: self.centerYAnchor),
        ])
    }
    public required init?(coder aDecoder: NSCoder) { fatalError() }
}
```

Tips / User interface / SwiftUI

```
struct ContentView: View {  
    var body: some View {  
        Text("SwiftUI born and raised!")  
    }  
}
```

Tips

Tips

Custom defined operators.

Tips

Custom defined operators.

Setters / Getters (property wrappers).

Tips

Custom defined operators.

Setters / Getters (property wrappers).

Ternary & Nil coalescing operators.

Tips

Custom defined operators.

Setters / Getters (property wrappers).

Ternary & Nil coalescing operators.

Free functions are not all bad.

Tips

Custom defined operators.

Setters / Getters (property wrappers).

Ternary & Nil coalescing operators.

Free functions are not all bad.

Global state (e.g., Environment).

Tips

Custom defined operators.

Setters / Getters (property wrappers).

Ternary & Nil coalescing operators.

Free functions are not all bad.

Global state (e.g., Environment).

Testing.

Tips

Custom defined operators.

Setters / Getters (property wrappers).

Ternary & Nil coalescing operators.

Free functions are not all bad.

Global state (e.g., Environment).

Testing.

“.isEmpty” over “.count > 0”.

Tips

Custom defined operators.

Setters / Getters (property wrappers).

Ternary & Nil coalescing operators.

Free functions are not all bad.

Global state (e.g., Environment).

Testing.

“.isEmpty” over “.count > 0”.

Danger + SwiftLint = CI/CD ❤!

Tips

Custom defined operators.

Setters / Getters (property wrappers).

Ternary & Nil coalescing operators.

Free functions are not all bad.

Global state (e.g., Environment).

Testing.

“.isEmpty” over “.count > 0”.

Danger + SwiftLint = CI/CD ❤!

Using pull request reviews as an opportunity to Code Golf.

If I had more time, I would have
written you a shorter letter

— Pascal



@jasonalexzurita