

DEBUGGING

Zeke Abuhoff

Lead iOS Instructor, General Assembly 



LEARNING OBJECTIVES

- + Parse an error message
- + Create a breakpoint
- + Read a stack trace

DEBUGGING

BUGS

Bugs are elusive. They hide, wait and ruin your app.



DEBUGGING

BUGS

Bugs are elusive. They hide, wait and ruin your app.

KILL THEM



DEBUGGING

BUGS



Bugs may come in many forms.

COMPILE-TIME ERRORS	You get a red icon in Xcode
IMPORTANT WARNINGS	You get a yellow icon in Xcode
RUNTIME ERRORS	No warnings, then a crash
LOGIC BUGS	No warnings, then odd behavior

ERROR MESSAGES

Σημείωση: Η συνάρτηση `abs()` υπολογίζει την απόλυτη τιμή ενός αριθμού. Η συνάρτηση `len()` υπολογίζει το μήκος ενός πίνακα. Η συνάρτηση `range()` δημιουργεί μια ακολουθία αριθμών. Η συνάρτηση `sum()` υπολογίζει το άθροισμα των στοιχείων ενός πίνακα. Η συνάρτηση `max()` υπολογίζει το μέγιστο στοιχείο ενός πίνακα. Η συνάρτηση `min()` υπολογίζει το ελάχιστο στοιχείο ενός πίνακα. Η συνάρτηση `sorted()` επιστρέφει μια νέα λίστα με τα στοιχεία του αρχικού πίνακα ταξινομημένα. Η συνάρτηση `reversed()` επιστρέφει μια νέα λίστα με τα στοιχεία του αρχικού πίνακα αντιστρεφόμενα. Η συνάρτηση `enumerate()` επιστρέφει μια ακολουθία ζευγών (αριθμός, στοιχείο). Η συνάρτηση `zip()` επιστρέφει μια ακολουθία ζευγών (στοιχείο, στοιχείο). Η συνάρτηση `map()` επιστρέφει μια ακολουθία αντικείμενων που έχουν υποστεί μετασχηματισμό. Η συνάρτηση `filter()` επιστρέφει μια ακολουθία αντικείμενων που ικανοποιούν ένα κριτήριο. Η συνάρτηση `reduce()` επιστρέφει ένα αντικείμενο που προκύπτει από την εφαρμογή μιας συνάρτησης σε μια ακολουθία αντικείμενων.



Also, you can copy an error message into Google.

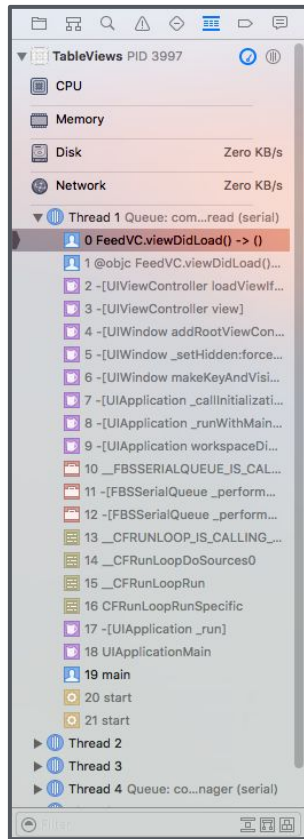
DEBUGGING

DEBUGGING TOOLS

Breakpoints

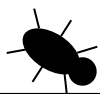
```
18  
19     override func viewDidLoad() {  
20         feedTable.delegate = self  
21         feedTable.dataSource = self  
22     }
```

Stack Trace



Step Through





DEBUGGING

BREAKPOINTS

Click on a line number to create a breakpoint.

When the code is running, it will stop just before executing that line.

The first piece of information this gives you is that the code reached that point. If the app

runs and it never hits your breakpoint, that part of the code is not running.

```
18  
19     override func viewDidLoad() {  
20         feedTable.delegate = self  
21         feedTable.dataSource = self  
22     }
```


DEBUGGING

BREAKPOINTS



The second piece of information you get from using a breakpoint is what the state of the app is when that line is reached.

You can look at the device screen to see if the UI looks as it's supposed to at that moment.

You can also use the console at the bottom of Xcode to read values of variables and constants.

```
2016-08-29 15:32:44.319646 TableViews[3997:226116] [User Defaults] CFPrefsPlistSource<0x6000000e0d00> (Domain: com.GA.TableViews, User: kCFPreferencesCurrentUser, ByHost: No, Container: (null)) is waiting for writes to complete so it can determine if new data is available
2016-08-29 15:32:44.320114 TableViews[3997:226116] [User Defaults] CFPrefsPlistSource<0x6000000e0d00> (Domain: kCFPreferencesAnyApplication, User: kCFPreferencesCurrentUser, ByHost: No, Container: (null)) is waiting for writes to complete so it can determine if new data is available
2016-08-29 15:32:44.320510 TableViews[3997:226116] [User Defaults] CFPrefsPlistSource<0x6000000e0d00> (Domain: kCFPreferencesAnyApplication, User: kCFPreferencesCurrentUser, ByHost: No, Container: (null)) is waiting for writes to complete so it can determine if new data is available
2016-08-29 15:32:44.324785 TableViews[3997:226116] [User Defaults] CFPrefsPlistSource<0x6000000e0d00> (Domain: com.GA.TableViews, User: kCFPreferencesCurrentUser, ByHost: No, Container: (null)) is waiting for writes to complete so it can determine if new data is available
2016-08-29 15:32:44.325926 TableViews[3997:226116] [User Defaults] CFPrefsPlistSource<0x6000000e0d00> (Domain: com.GA.TableViews, User: kCFPreferencesCurrentUser, ByHost: No, Container: (null)) is waiting for writes to complete so it can determine if new data is available
2016-08-29 15:32:44.326326 TableViews[3997:226116] [User Defaults] CFPrefsPlistSource<0x6000000e0d00> (Domain: kCFPreferencesAnyApplication, User: kCFPreferencesCurrentUser, ByHost: No, Container: (null)) is waiting for writes to complete so it can determine if new data is available
2016-08-29 15:32:44.326997 TableViews[3997:226116] [User Defaults] CFPrefsPlistSource<0x6000000e0d00> (Domain: kCFPreferencesAnyApplication, User: kCFPreferencesCurrentUser, ByHost: No, Container: (null)) is waiting for writes to complete so it can determine if new data is available
(lldb)
```

DEBUGGING

STEP THROUGH

Once an app is paused by a breakpoint, you can click on the step through buttons to make the app execute the next operation.



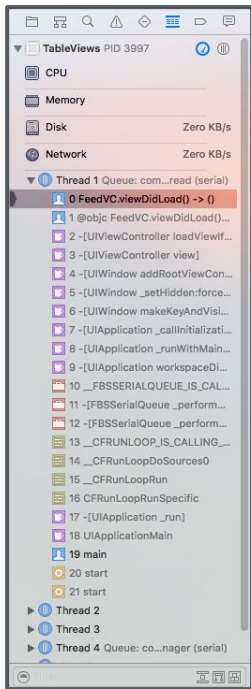
This lets you see the app's logic, line by line.



By using the step up and step down buttons, you can move to different scopes within the code, seeing how functions work.

DEBUGGING

STACK TRACE



The stack trace shows you every operation currently in progress in the app.

You can see which functions called which other functions, as well as the threads on which these operations are happening.

