Using LLDB in iOS

Using Expressions and Variables

raywenderlich.com

expr

- Can type “raw” Objective-C code into LLDB parser which will be evaluated in real time.

```
(lldb) expr self.view.hidden = YES
```

- Use -- as a separator from options and the actual command

```
(lldb) expr -f bin -- 5+5
```

expr (cont'd)

- May change the running state of your program

```
(lldb) expr self.player.lives = 100
```

- Can be used to log messages to the console

```
(lldb) expr (void) NSLog(@"hello world!")
```

- Can call methods on objects

```
(lldb) expr (BOOL) [self.myArray containsObject:@“CarKeys"]
```

expr (cont'd)

- Print out structures in your code

```
(lldb) expr -- (CGRect) [self.view frame]
```

- Take "shortcuts" in your app

```
(lldb) expr [self prepareForSeuge:@"mySegue" sender:nil]
```

LLDB Variables

- Variable names must have a type and begin with a $

```
(lldb) expr int $meaningOfLife = 42
```

- Can be used with other expressions

```
(lldb) expr 100 + $meaningOfLife
```

- Some expressions provide results in LLDB variables

```
(int) $0 = 142
(lldb) p $0 + 200
(int) $1 = 242
```

Using Variables with expr

- Create and run code on the fly

```
(lldb) expr NSString * $json = [self fetchRemoteData];
```

```
(lldb) expr NSData * $data = [$json dataUsingEncoding:4]
```

```
(lldb) expr NSDictionary * $parsedJson = [NSJSONSerialization JSONObjectWithData:
$data options:0 error:NULL];
```

```
(lldb) po parsedData
{ username : Brian, password : 12345 }
```

Demo

- Start a project and change the state with expr
- Create new variable
- Update UI via. LLDB using a LLDB variable

Challenge Time

```
<?xml version="1.0"?>
<video_games>
    <video_game>
        <name>Minecraft</name>
        <genre>Worldbuilding FPS</genre>
        <rating>5</rating>
        <synopsis>Build your own voxel based world while ...</synopsis>
    </video_game>
</video_games>
```