Using LLDB in iOS

Watchpoint, Script, Command

raywenderlich.com

watchpoint

- List a watchpoints

```
(lldb) watchpoint list
Number of supported hardware watchpoints: 4
Current watchpoints:
Watchpoint 1: addr = 0x0a34dce4 size = 4 state = enabled type = w
    watchpoint spec = '_time'
    new value: 4
```

- Delete a watchpoint

```
(lldb) watchpoint delete 1
1 watchpoints deleted.
```

watchpoint (cont'd)

- Set a watchpoint

```
(lldb) watchpoint set variable _x
Watchpoint created: Watchpoint 3: addr = 0x0a159130 size = 4 state =
enabled type = w
    watchpoint spec = '_x'
    new value: 0
```

- Add a condition on an address

```
(lldb) watchpoint set expression -- my_pointer
Watchpoint created: Watchpoint 3: addr = 0x08e75960 size = 4 state =
enabled type = w
    new value: 0x00005944
```

watchpoint (cont'd)

- Add a condition on a watchpoint

```
(lldb) watchpoint modify -c "_x < 0" 1
1 watchpoints modified.
```

- Remove a condition from a watchpoint

```
(lldb) watchpoint modify -c "" 1
1 watchpoints modified.
```

script

- LLDB contains an embedded Python interpreter
- The entire API is exposed through Python scripting bindings
- The script command parses raw Python commands

```
(lldb) script print(sys.version)
2.7.5 (default, Aug 25 2013, 00:04:04)
[GCC 4.2.1 Compatible Apple LLVM 5.0 (clang-500.0.68)]
```

script (cont'd)

- Run python scripts from a breakpoint
  - LLDB creates a Python function to encapsulate the scripts.
  - If you want to access the script variables outside the breakpoint, you must declare them as global variables.

script (cont'd)

```
(lldb) breakpoint command add -s python 1
Enter your Python command(s). Type 'DONE' to end.
def function(frame,bp_loc,internal_dict):
    """frame: the SBFrame for the location at which you stopped
       bp_loc: an SBBreakpointLocation for the breakpoint location
information
       internal_dict: an LLDB support object not to be used"""
    print "hello"
    global my_name_is = "Inigo Montoya, prepare to crash."
    DONE
```

Breakpoint Functions

```
def breakpoint_func(frame, bp_loc, dict):
```

- **frame**: The current stack frame of the breakpoint
- **bp_loc**: The current breakpoint location
- **dict**: The python session dictionary

```
(lldb) breakpoint command add -F my.breakpoint_func
```

command

- Import existing scripts scripts to be used during your debugging session.

```
(lldb) command script import ~/my_script.py
```

- Create a new LLDB command by calling a Python function.

```
(lldb) command script add -f my_script.python_function cmd_name
```

command (cont'd)

- Import existing LLDB debugger scripts

```
(lldb) command import ~/my_lldb_commands.txt
```

- Delete user create aliases

```
(lldb) command unalias pf
```

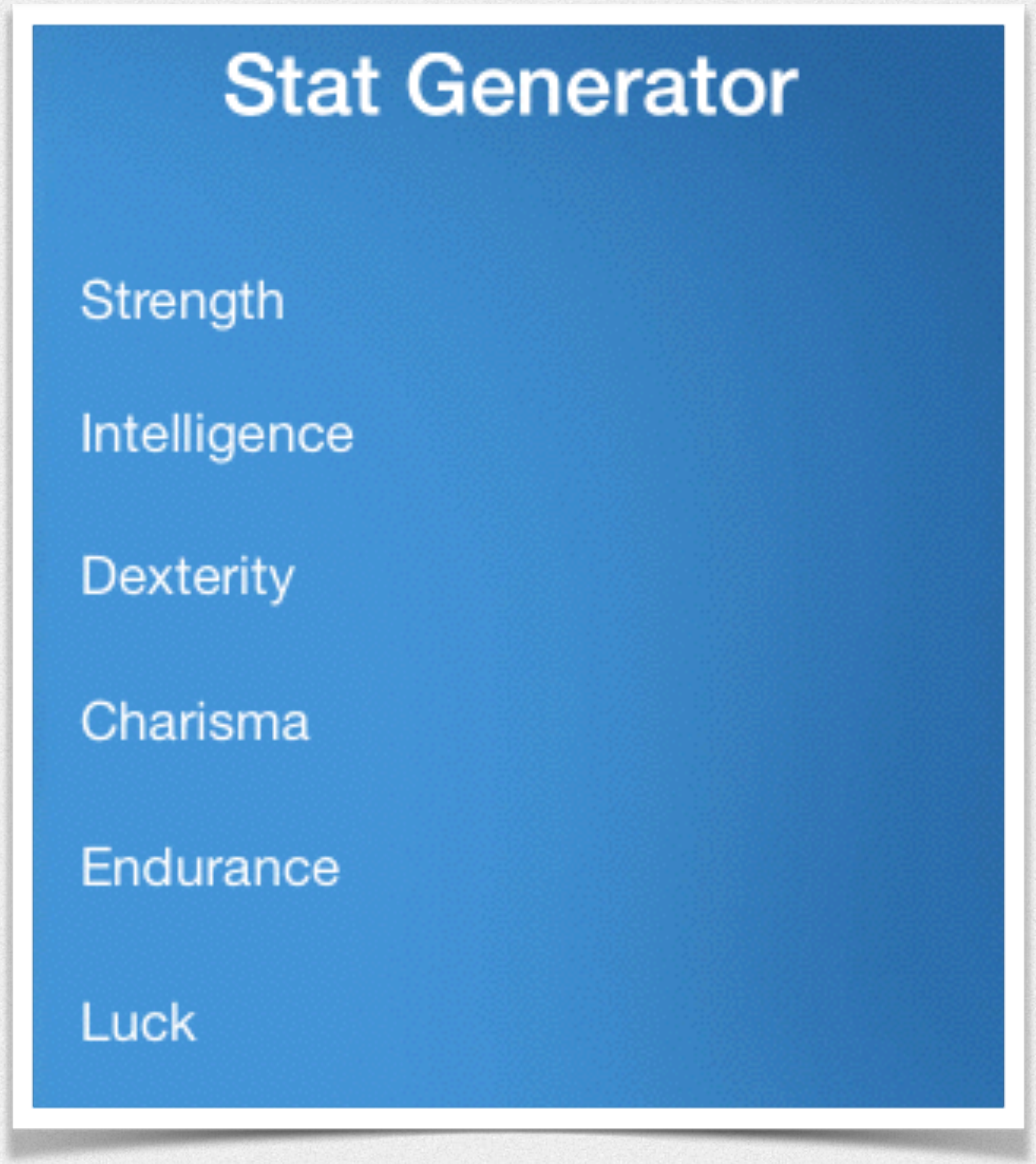
- Print out command history

```
(lldb) command history
```

Demo: Revealing the Data

- Create a watchpoint / add a condition / and delete a watchpoint. Try and exceed watchpoint limit.
- Demonstrate the script command - create a breakpoint script
- Create a custom command referencing a Python script

Challenge Time


Stat Generator

Strength

Intelligence

Dexterity

Charisma

Endurance

Luck