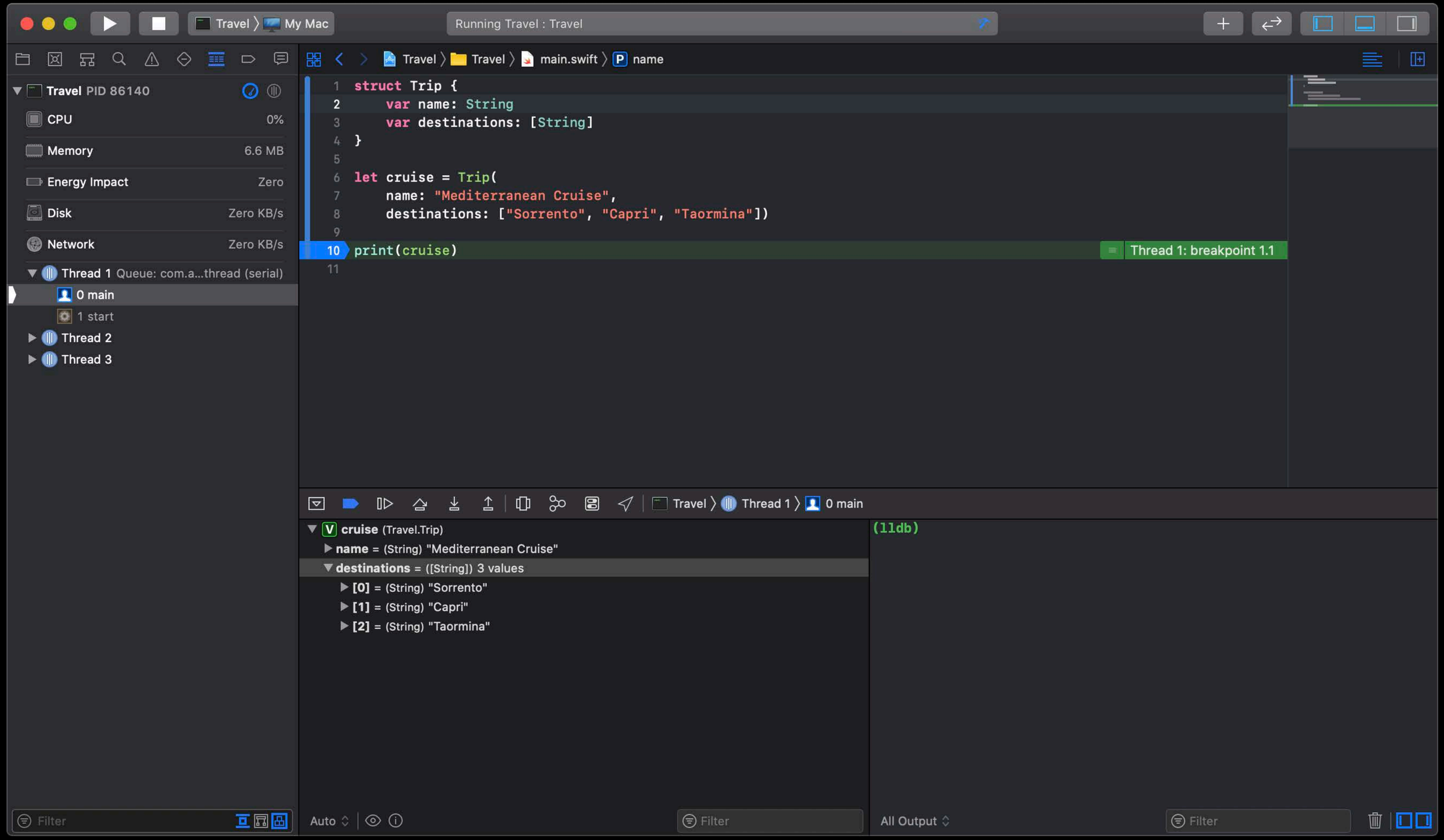
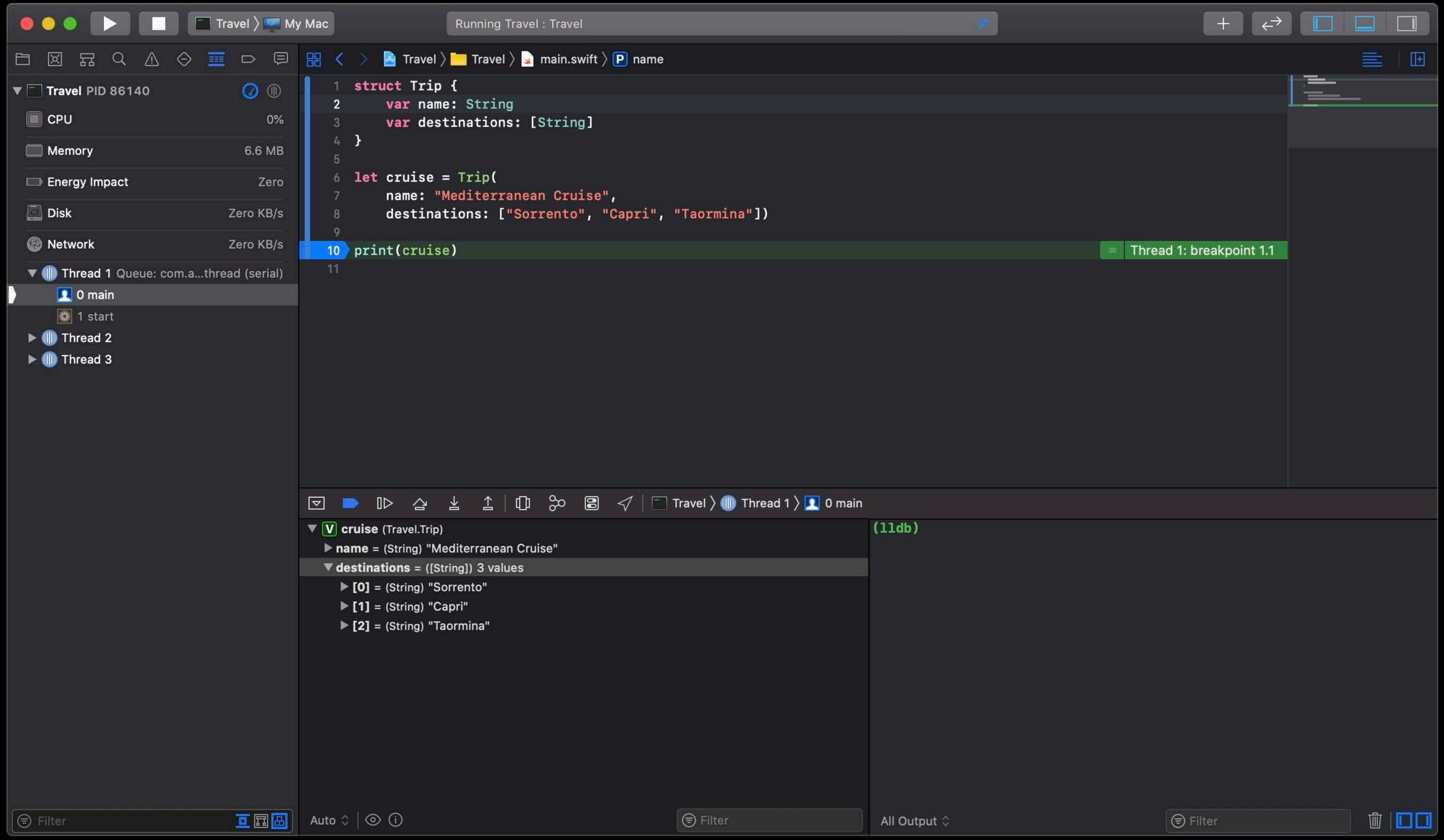
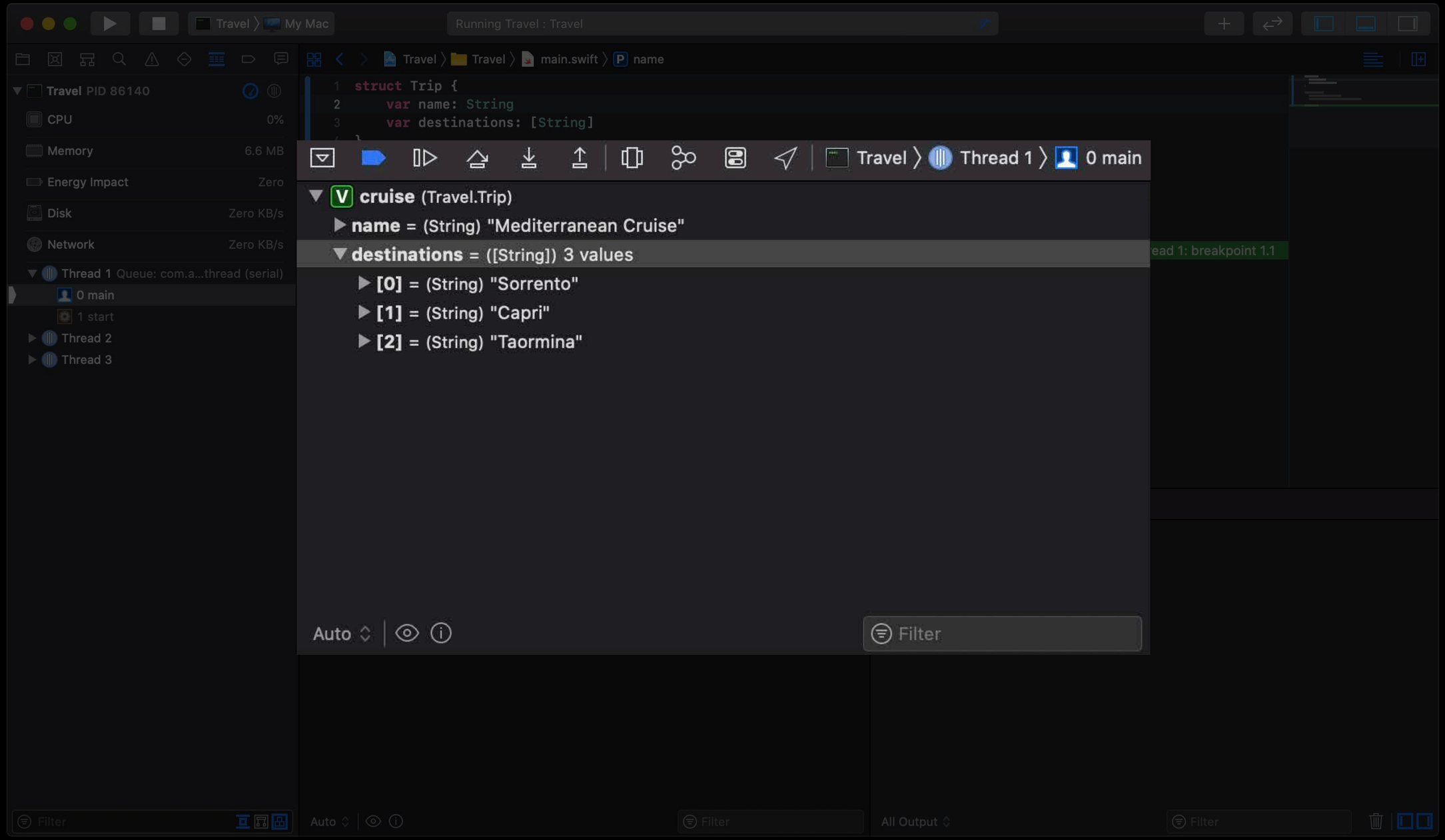
#WWDC19

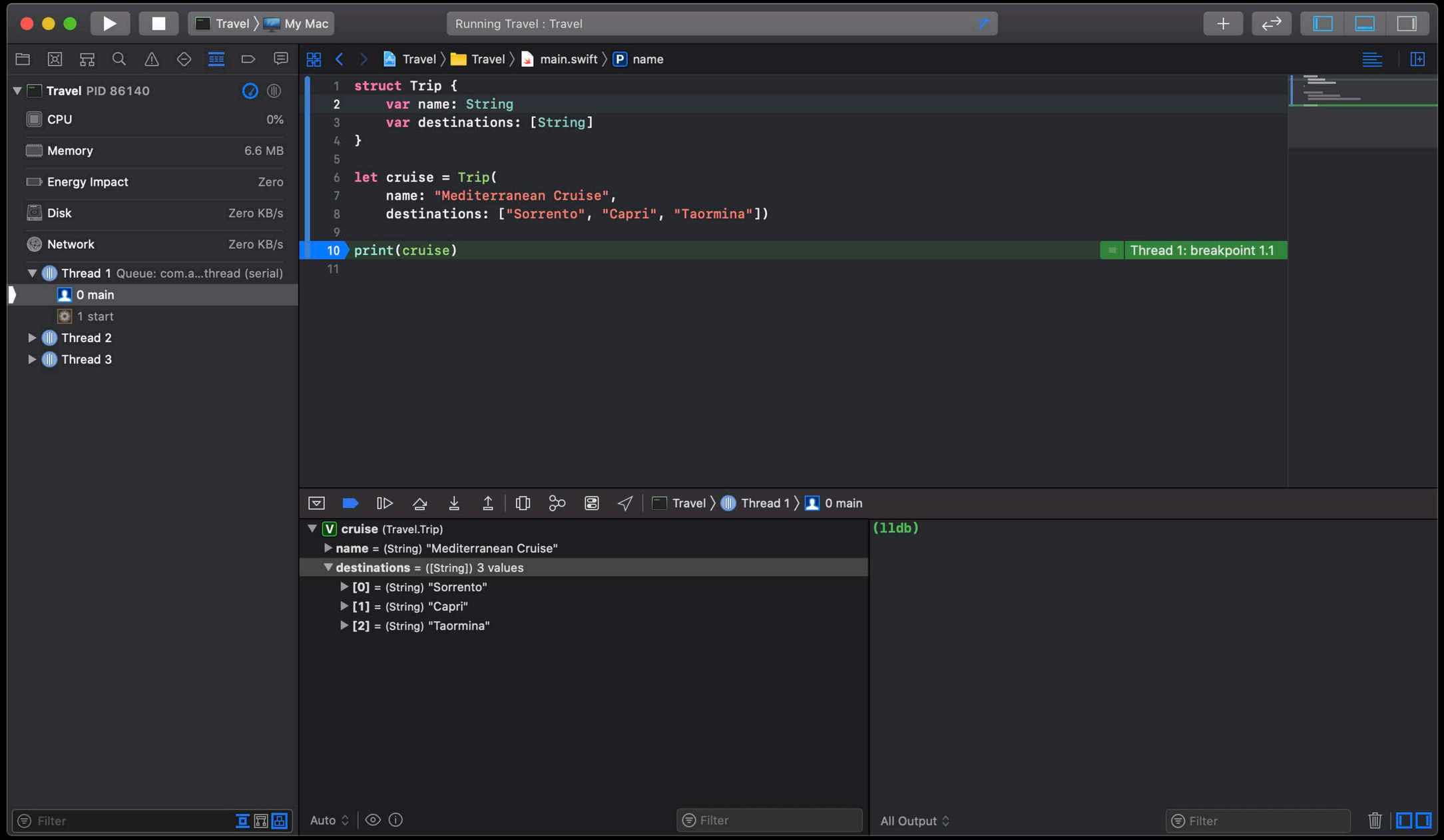
LLDB: Beyond “po”

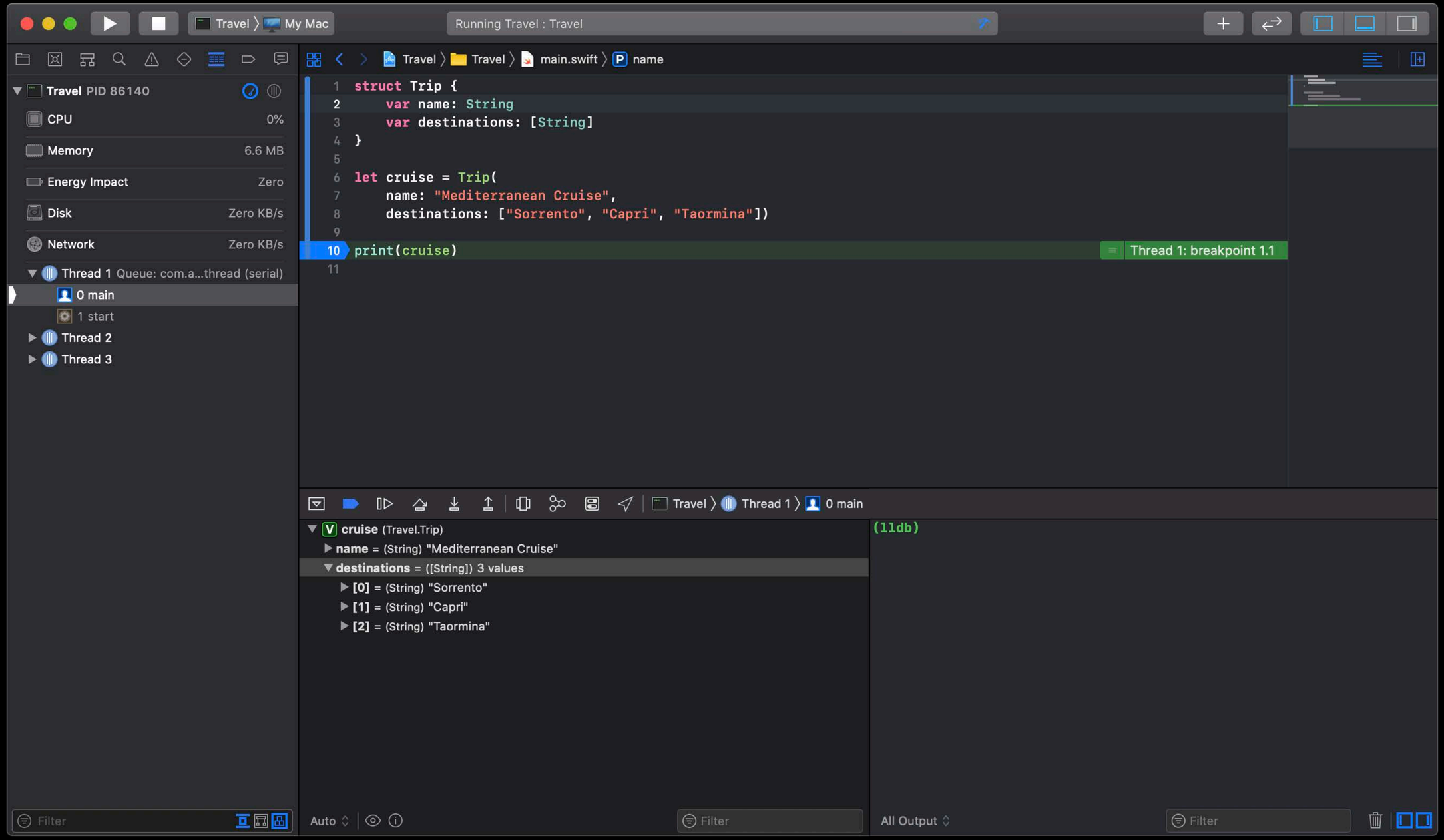
Davide Italiano, Software Engineer
Jonas Devlieghere, Software Engineer

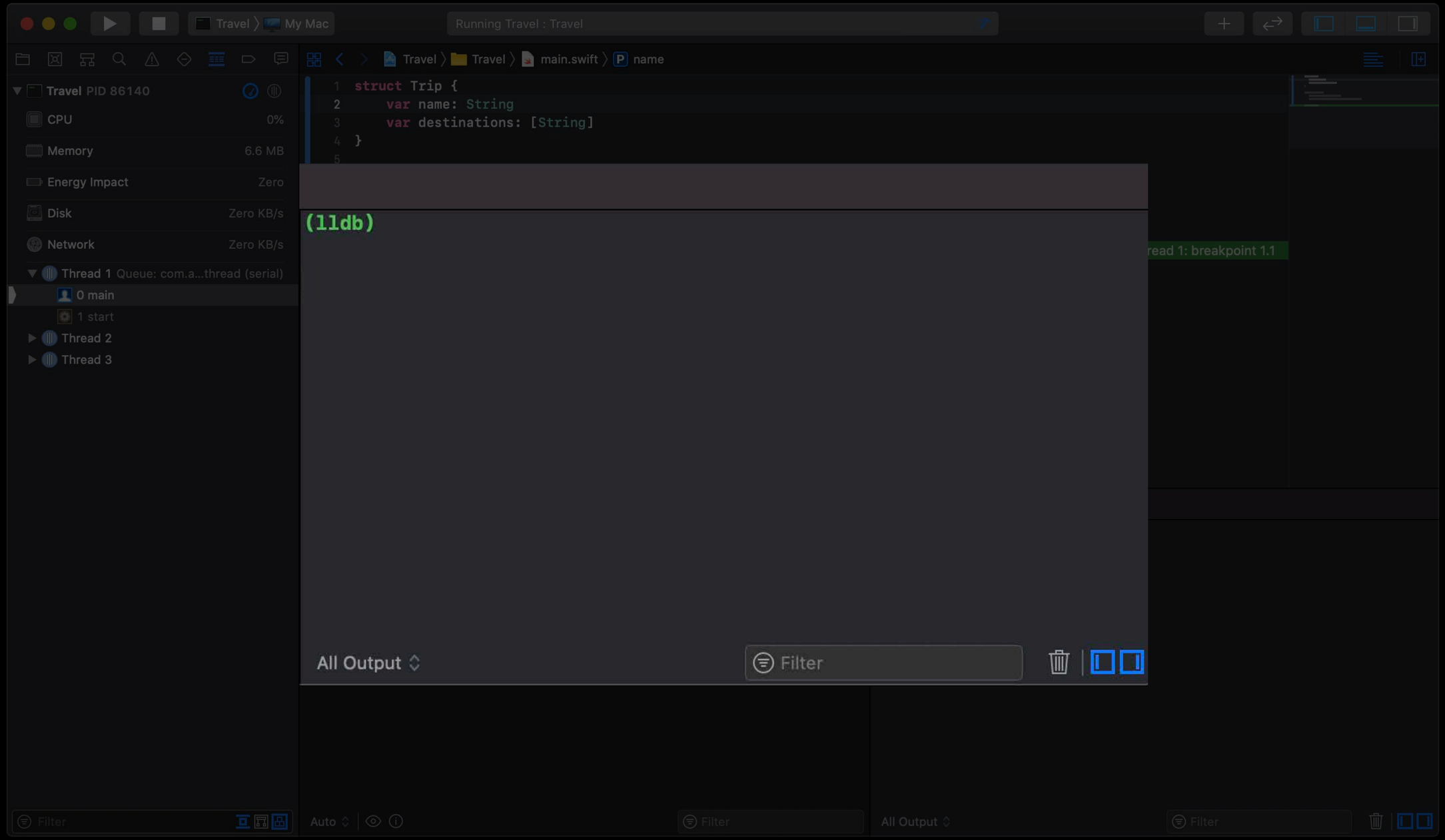

Travel 〉 My Mac
Running Travel : Travel
Travel 〉 Travel 〉 main.swift 〉 name
Travel PID 86140
CPU 0%
Memory 6.6 MB
Energy Impact Zero
Disk Zero KB/s
Network Zero KB/s
Thread 1 Queue: com.a...thread (serial)
0 main
1 start
Thread 2
Thread 3
1 struct Trip {
2 var name: String
3 var destinations: [String]
4 }
5
6 let cruise = Trip(
7 name: "Mediterranean Cruise",
8 destinations: ["Sorrento", "Capri", "Taormina"])
9
10 print(cruise)
11
Thread 1: breakpoint 1.1
Travel 〉 Thread 1 〉 0 main
cruise (Travel.Trip)
name = (String) "Mediterranean Cruise"
destinations = ([String]) 3 values
[0] = (String) "Sorrento"
[1] = (String) "Capri"
[2] = (String) "Taormina"
(lldb)
Filter
Auto
Filter
All Output
Filter

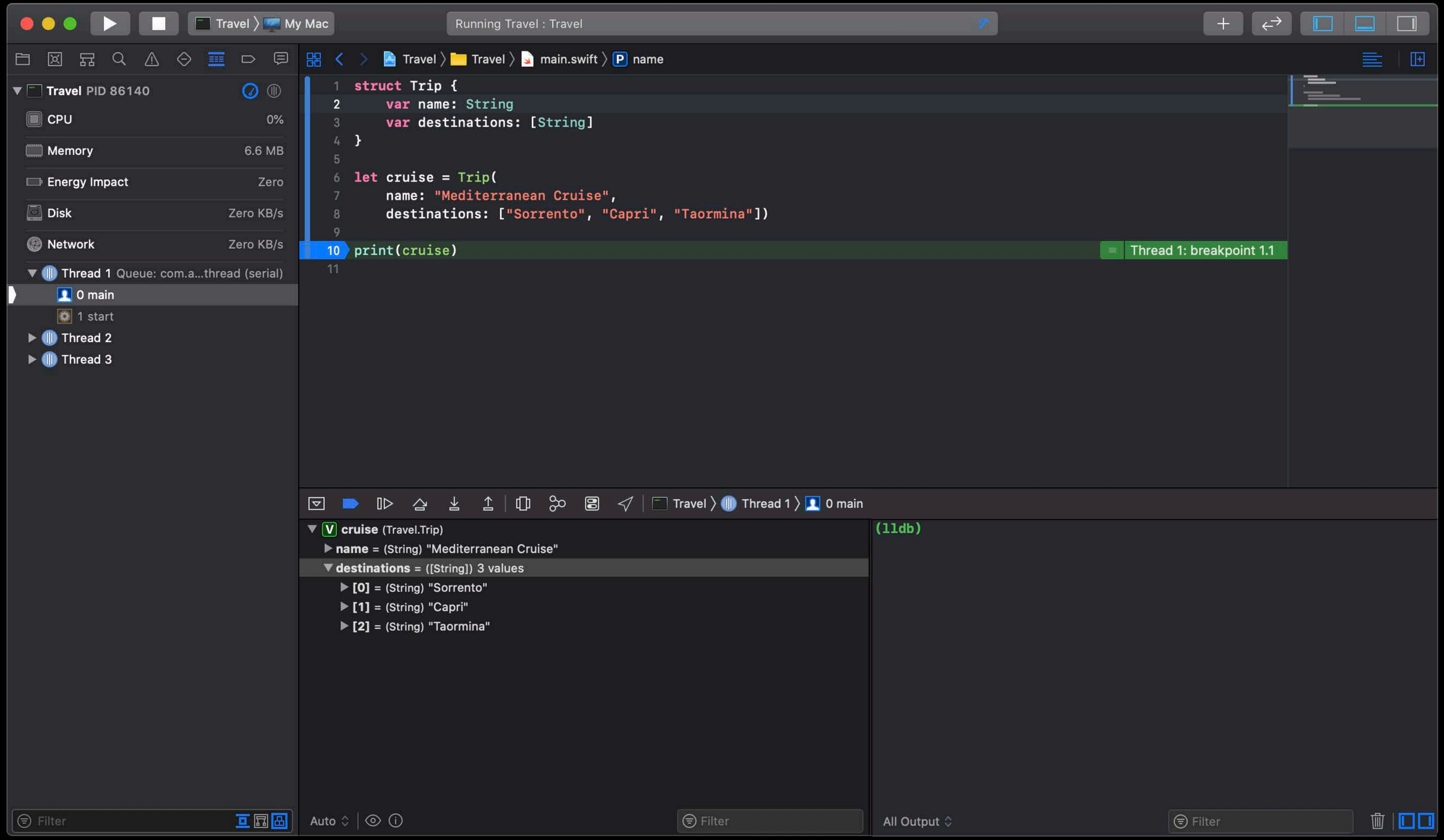

Travel 〉 My Mac
Running Travel : Travel
Travel 〉 Travel 〉 main.swift 〉 name
1 struct Trip {
2 var name: String
3 var destinations: [String]
4 }
5
6 let cruise = Trip(
7 name: "Mediterranean Cruise",
8 destinations: ["Sorrento", "Capri", "Taormina"])
9
10 print(cruise)
11
Thread 1: breakpoint 1.1
Travel PID 86140
CPU 0%
Memory 6.6 MB
Energy Impact Zero
Disk Zero KB/s
Network Zero KB/s
Thread 1 Queue: com.a...thread (serial)
0 main
1 start
Thread 2
Thread 3
Travel 〉 Thread 1 〉 0 main
cruise (Travel.Trip)
name = (String) "Mediterranean Cruise"
destinations = ([String]) 3 values
[0] = (String) "Sorrento"
[1] = (String) "Capri"
[2] = (String) "Taormina"
(lldb)
Filter
Auto
Filter
All Output
Filter


Travel 〉My Mac
Running Travel : Travel
Travel 〉Travel 〉main.swift 〉name
1 struct Trip {
2 var name: String
3 var destinations: [String]
Travel PID 86140
CPU 0%
Memory 6.6 MB
Energy Impact Zero
Disk Zero KB/s
Network Zero KB/s
Thread 1 Queue: com.a...thread (serial)
0 main
1 start
Thread 2
Thread 3
Travel 〉Thread 1 〉0 main
cruise (Travel.Trip)
name = (String) "Mediterranean Cruise"
destinations = ([String]) 3 values
[0] = (String) "Sorrento"
[1] = (String) "Capri"
[2] = (String) "Taormina"
Auto
Filter
ead 1: breakpoint 1.1
All Output


Travel 〉 My Mac
Running Travel : Travel
Travel 〉 Travel 〉 main.swift 〉 name
1 struct Trip {
2 var name: String
3 var destinations: [String]
4 }
5
6 let cruise = Trip(
7 name: "Mediterranean Cruise",
8 destinations: ["Sorrento", "Capri", "Taormina"])
9
10 print(cruise)
11
Thread 1: breakpoint 1.1
Travel PID 86140
CPU 0%
Memory 6.6 MB
Energy Impact Zero
Disk Zero KB/s
Network Zero KB/s
Thread 1 Queue: com.a...thread (serial)
0 main
1 start
Thread 2
Thread 3
Travel 〉 Thread 1 〉 0 main
cruise (Travel.Trip)
name = (String) "Mediterranean Cruise"
destinations = ([String]) 3 values
[0] = (String) "Sorrento"
[1] = (String) "Capri"
[2] = (String) "Taormina"
(lldb)
Filter
Auto
Filter
All Output
Filter


Travel 〉 My Mac
Running Travel : Travel
Travel 〉 Travel 〉 main.swift 〉 name
1 struct Trip {
2 var name: String
3 var destinations: [String]
4 }
5
6 let cruise = Trip(
7 name: "Mediterranean Cruise",
8 destinations: ["Sorrento", "Capri", "Taormina"])
9
10 print(cruise)
11
Thread 1: breakpoint 1.1
Travel PID 86140
CPU 0%
Memory 6.6 MB
Energy Impact Zero
Disk Zero KB/s
Network Zero KB/s
Thread 1 Queue: com.a...thread (serial)
0 main
1 start
Thread 2
Thread 3
Travel 〉 Thread 1 〉 0 main
cruise (Travel.Trip)
name = (String) "Mediterranean Cruise"
destinations = ([String]) 3 values
[0] = (String) "Sorrento"
[1] = (String) "Capri"
[2] = (String) "Taormina"
(lldb)
Filter
Auto
Filter
All Output
Filter


Travel
My Mac
Running Travel : Travel
Travel
Travel
main.swift
name
struct Trip {
var name: String
var destinations: [String]
}
Travel PID 86140
CPU 0%
Memory 6.6 MB
Energy Impact Zero
Disk Zero KB/s
Network Zero KB/s
Thread 1 Queue: com.a...thread (serial)
0 main
1 start
Thread 2
Thread 3
(lldb)
All Output
Filter
read 1: breakpoint 1.1
Auto
Filter
All Output
Filter


Travel 〉 My Mac
Running Travel : Travel
Travel 〉 Travel 〉 main.swift 〉 name
Travel PID 86140
CPU 0%
Memory 6.6 MB
Energy Impact Zero
Disk Zero KB/s
Network Zero KB/s
Thread 1 Queue: com.a...thread (serial)
0 main
1 start
Thread 2
Thread 3
1 struct Trip {
2 var name: String
3 var destinations: [String]
4 }
5
6 let cruise = Trip(
7 name: "Mediterranean Cruise",
8 destinations: ["Sorrento", "Capri", "Taormina"])
9
10 print(cruise)
11
Thread 1: breakpoint 1.1
Travel 〉 Thread 1 〉 0 main
cruise (Travel.Trip)
name = (String) "Mediterranean Cruise"
destinations = ([String]) 3 values
[0] = (String) "Sorrento"
[1] = (String) "Capri"
[2] = (String) "Taormina"
(lldb)
Filter
Auto
Filter
All Output
Filter

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```swift
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) po cruise
▿ Trip
  - name : "Mediterranean Cruise"
  ▿ destinations : 3 elements
    - 0 : "Sorrento"
    - 1 : "Capri"
    - 2 : "Taormina"
```

```
struct Trip {
    var name: String
    var destinations: [String]
}
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

extension Trip: CustomDebugStringConvertible {
    var debugDescription: String { "Trip description" }
}
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

extension Trip: CustomDebugStringConvertible {
    var debugDescription: String { "Trip description" }
}
```

```
(lldb) po cruise
▿ Trip description
  - name : "Mediterranean Cruise"
  ▿ destinations : 3 elements
    - 0 : "Sorrento"
    - 1 : "Capri"
    - 2 : "Taormina"
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

extension Trip: CustomDebugStringConvertible {
    var debugDescription: String { "Trip description" }
}
```

```
(lldb) po cruise
▿ Trip description
  - name : "Mediterranean Cruise"
  ▿ destinations : 3 elements
    - 0 : "Sorrento"
    - 1 : "Capri"
    - 2 : "Taormina"
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

extension Trip: CustomReflectable {
    …
}
```

```
(lldb) po cruise
▿ Trip description
  - name : "Mediterranean Cruise"
  ▿ destinations : 3 elements
    - 0 : "Sorrento"
    - 1 : "Capri"
    - 2 : "Taormina"
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

extension Trip: CustomReflectable {
    …
}
```

```
(lldb) po cruise
▿ Trip description
  - name : "Mediterranean Cruise"
  ▿ destinations : 3 elements
    - 0 : "Sorrento"
    - 1 : "Capri"
    - 2 : "Taormina"
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) po cruise.name.uppercased()
"MEDITERRANEAN CRUISE"
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) po cruise.name.uppercased()
"MEDITERRANEAN CRUISE"
(lldb) po cruise.destinations.sorted()
▿ 3 elements
  - 0 : "Capri"
  - 1 : "Sorrento"
  - 2 : "Taormina"
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) expression --object-description -- cruise
"Mediterranean Cruise"
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) expression --object-description -- cruise
"Mediterranean Cruise"
(lldb) command alias my_po expression --object-description
(lldb)
```

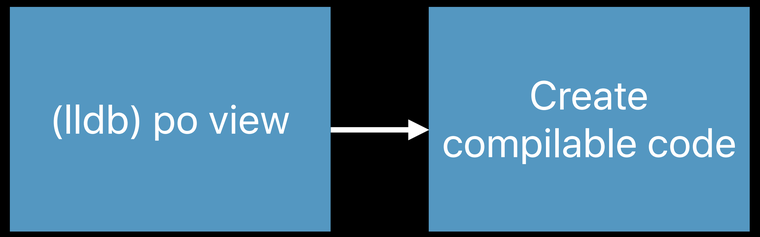
```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) expression --object-description -- cruise
"Mediterranean Cruise"
(lldb) command alias my_po expression --object-description
(lldb) my_po cruise
"Mediterranean Cruise"
```

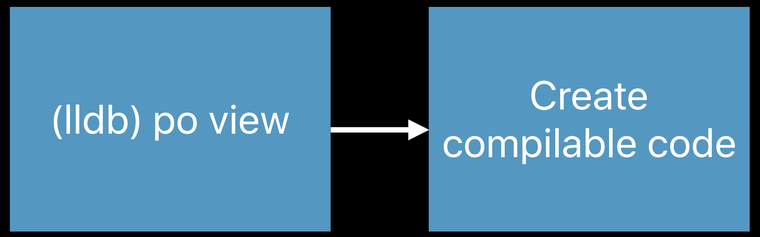
“po” Under the Hood

(lldb) po view

“po” Under the Hood

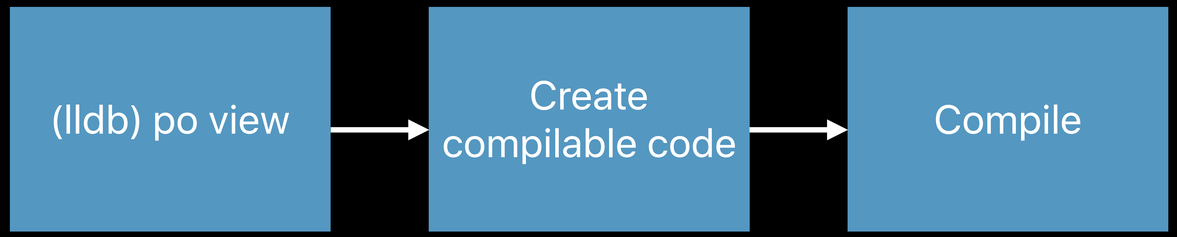


“po” Under the Hood

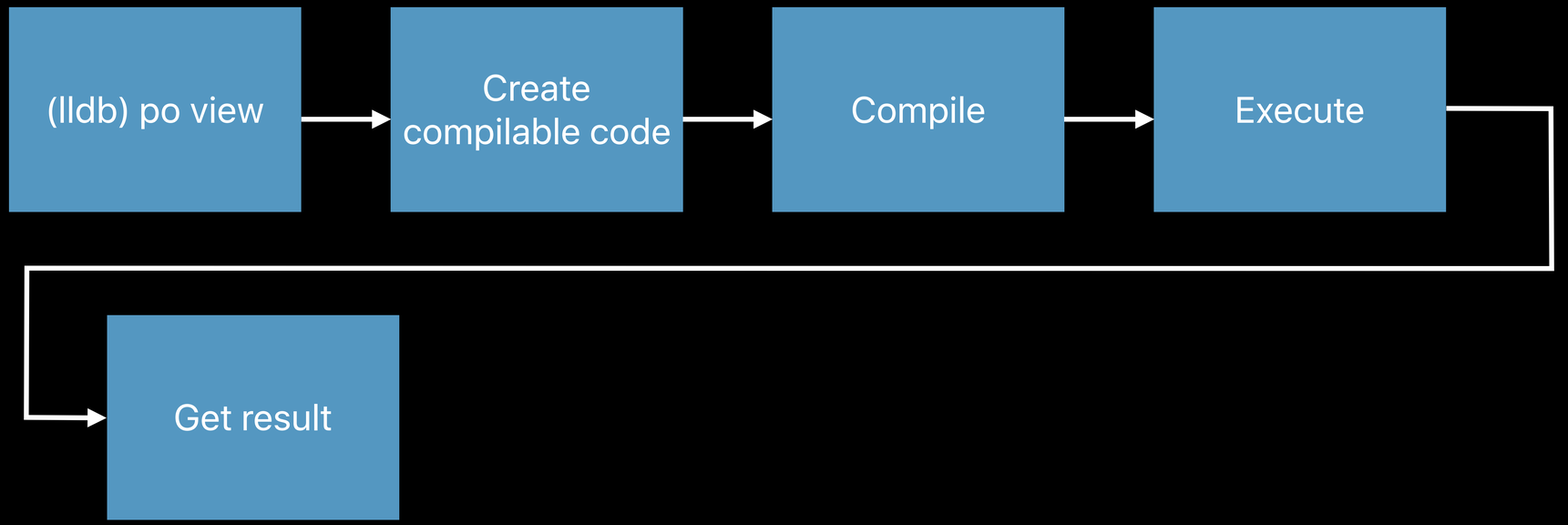



```
func __lldb_expr() {
    __lldb_res = view
}
```

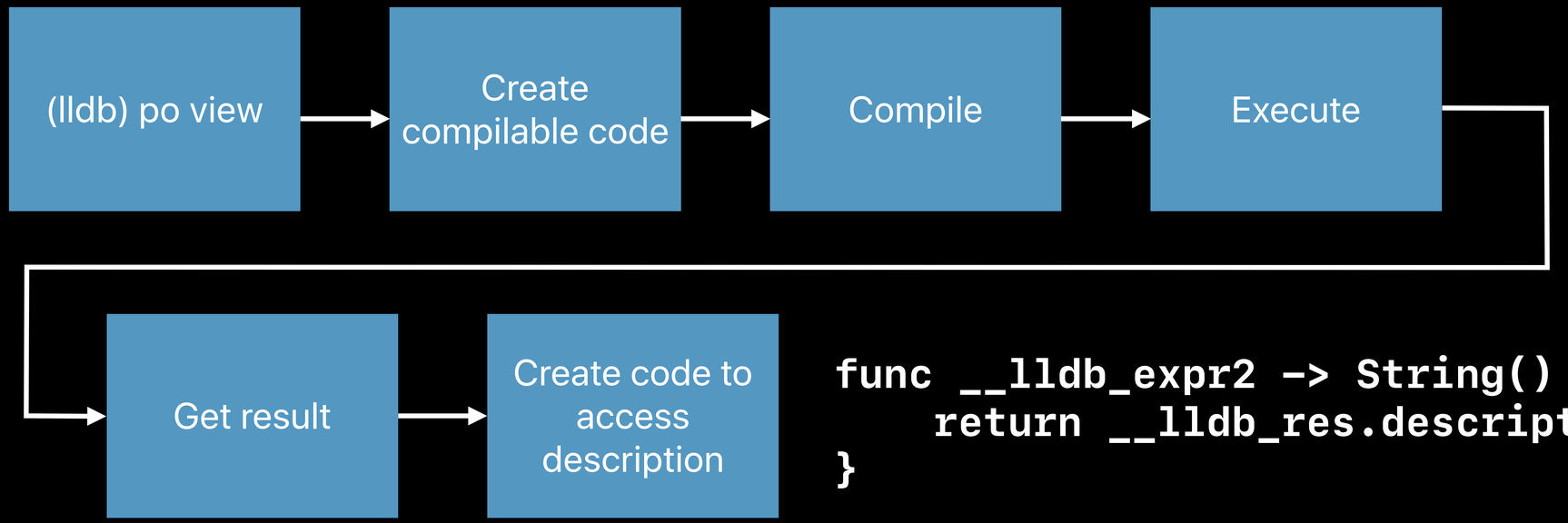
“po” Under the Hood



“po” Under the Hood

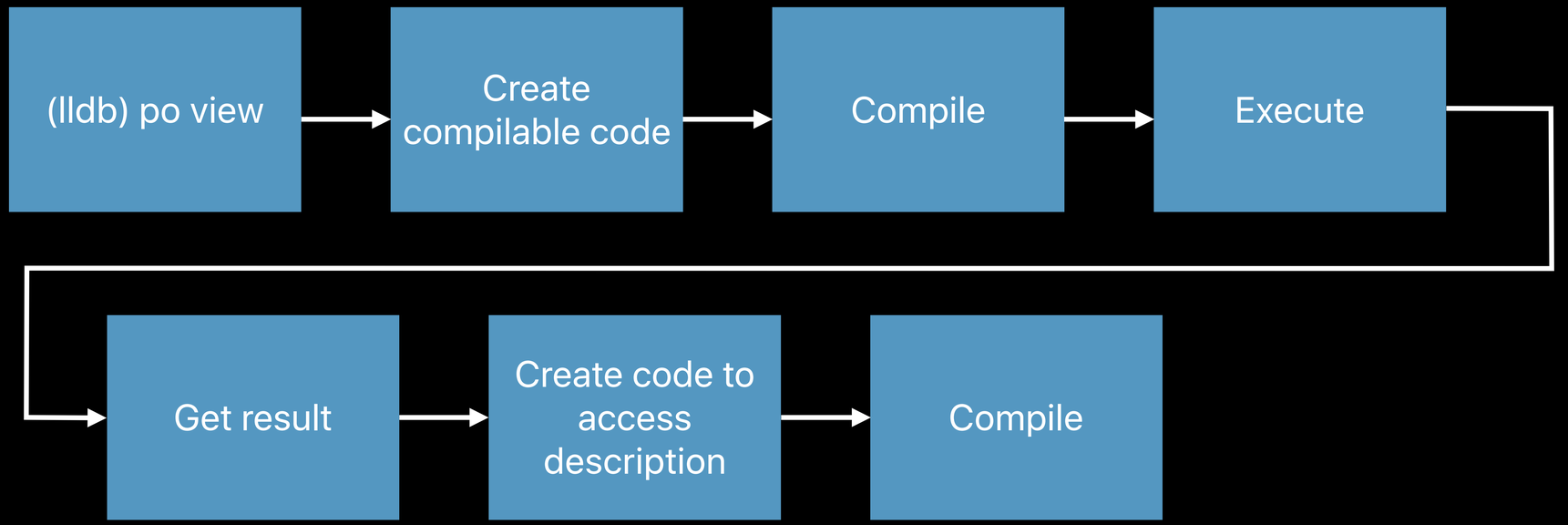


“po” Under the Hood

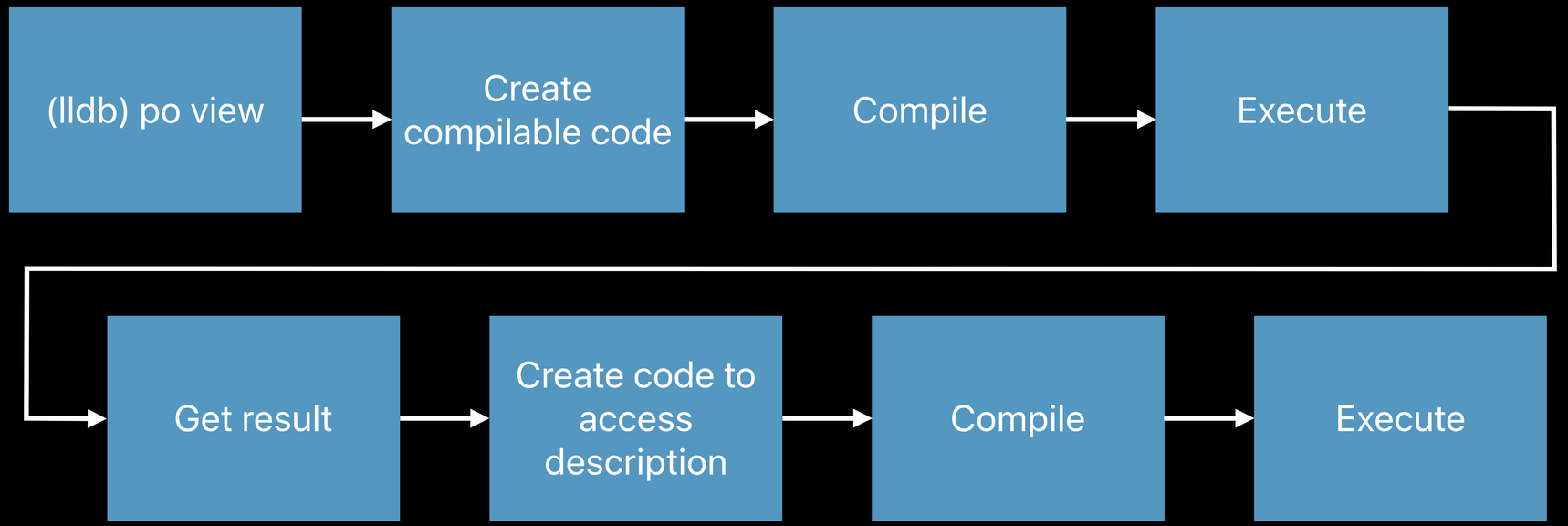



```
func __lldb_expr2 -> String() {
    return __lldb_res.description
}
```

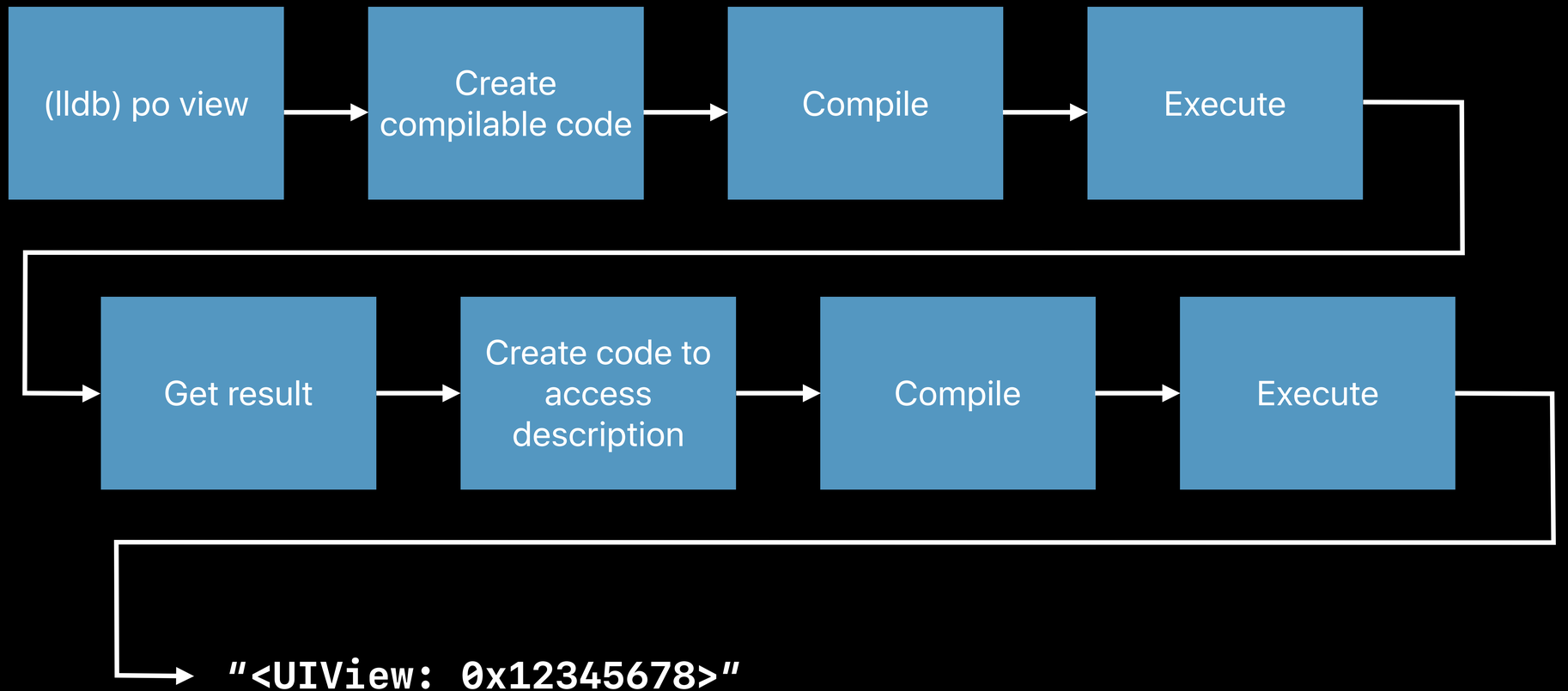
“po” Under the Hood



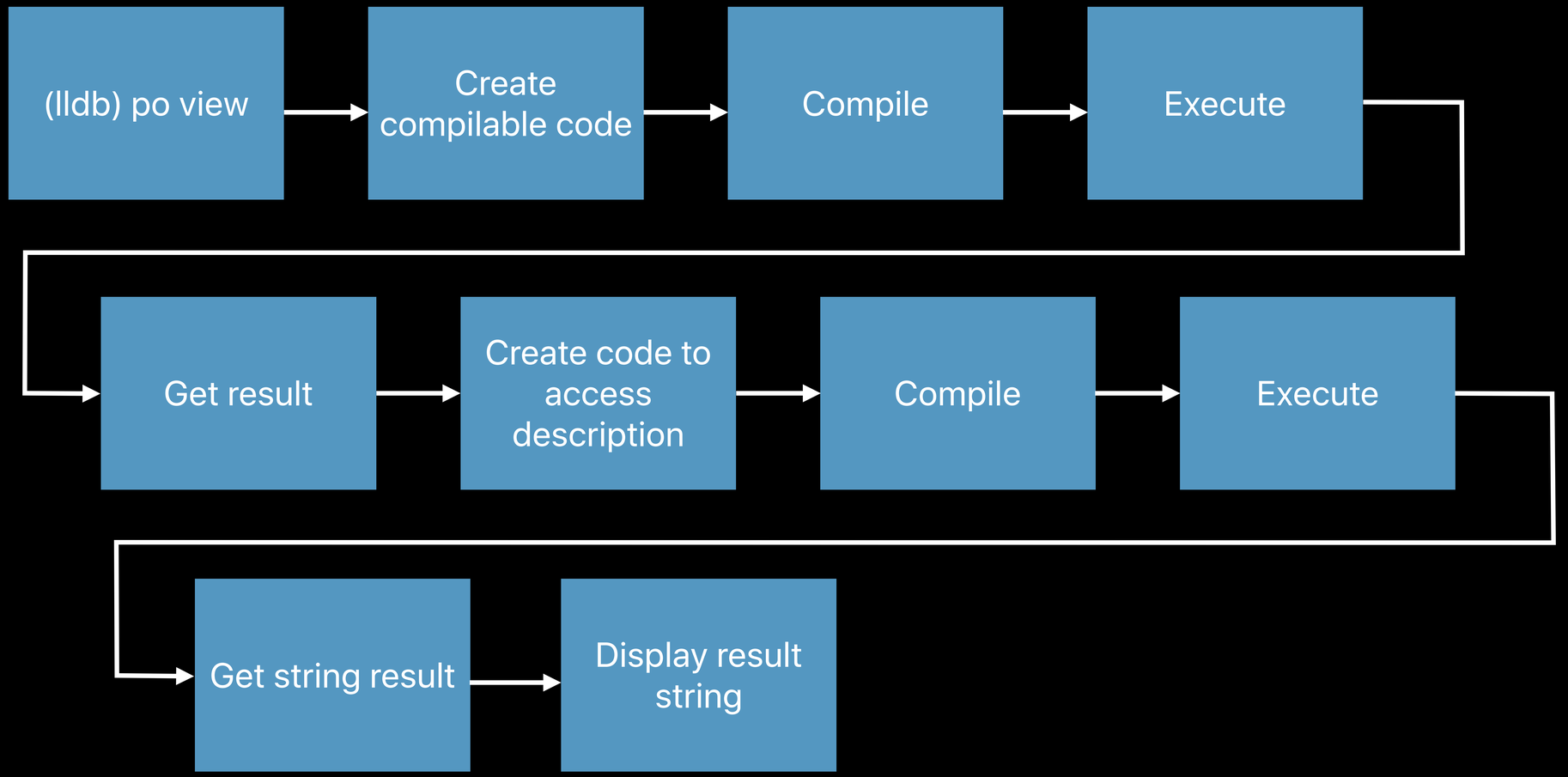
“po” Under the Hood



“po” Under the Hood



“po” Under the Hood



```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) p cruise
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) p cruise
(Travel.Trip) $R0 = {
  name = "Mediterranean Cruise"
  destinations = 3 values {
    [0] = "Sorrento"
    [1] = "Capri"
    [2] = "Taormina"
  }
}
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) p cruise
(Travel.Trip) $R0 = {
  name = "Mediterranean Cruise"
  destinations = 3 values {
    [0] = "Sorrento"
    [1] = "Capri"
    [2] = "Taormina"
  }
}
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) p $R0.destinations
([String]) $R1 = 3 values {
  [0] = "Sorrento"
  [1] = "Capri"
  [2] = "Taormina"
}
(lldb)
```

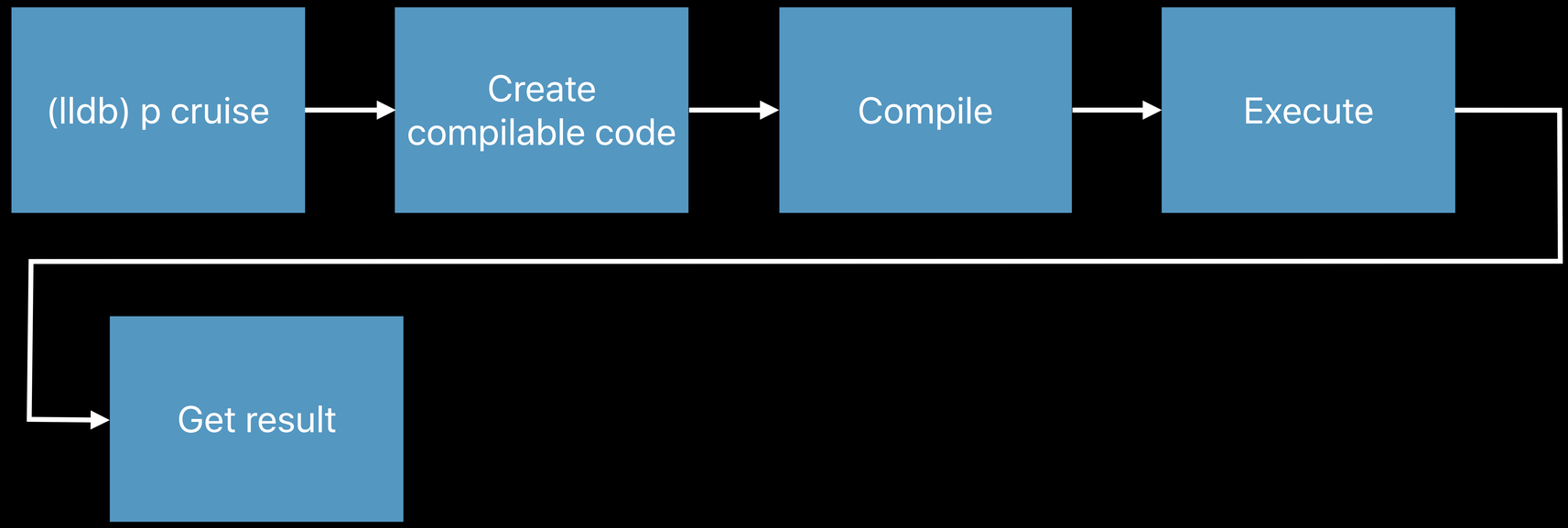
```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) p $R0.destinations
([String]) $R1 = 3 values {
  [0] = "Sorrento"
  [1] = "Capri"
  [2] = "Taormina"
}
(lldb) expression $R0.name
(String) $R2 = "Mediterranean Cruise”
```

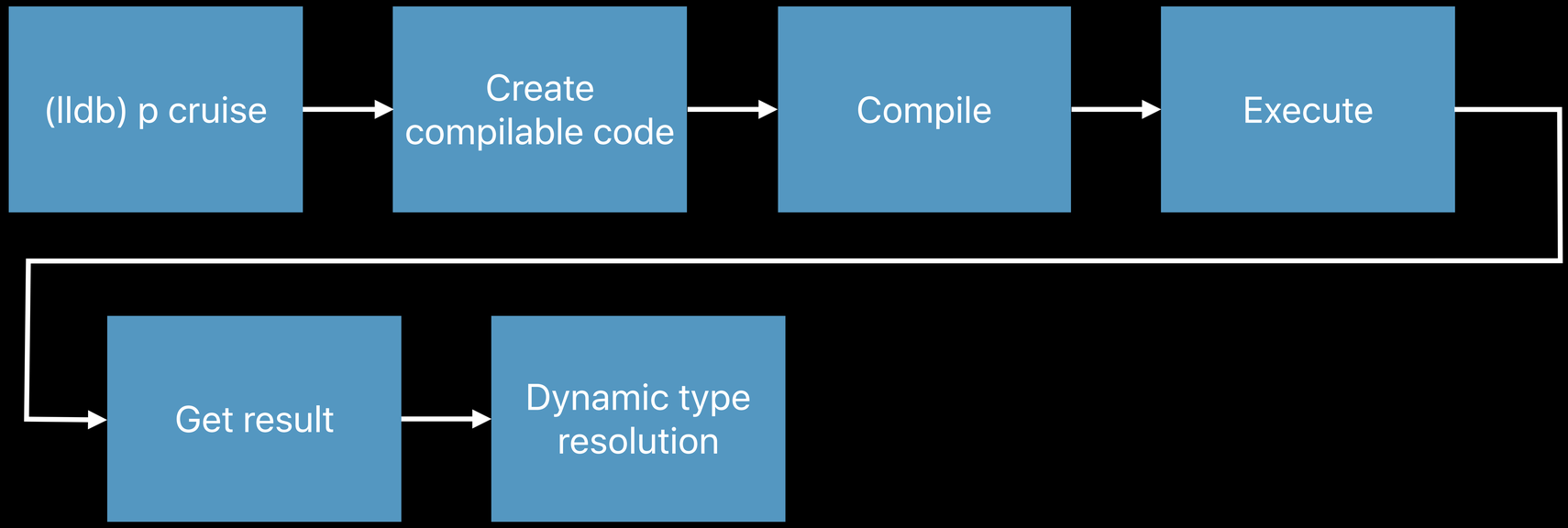
“p” Under the Hood

(lldb) p cruise

“p” Under the Hood



“p” Under the Hood



```
protocol Activity {}
struct Trip: Activity {
    var name: String
    var destinations: [String]
}

let cruise: Activity = Trip(…)
```

```
(lldb)
```

```
protocol Activity {}
struct Trip: Activity {
    var name: String
    var destinations: [String]
}

let cruise: Activity = Trip(…)
```

```
(lldb)
```

```
protocol Activity {}
struct Trip: Activity {
    var name: String
    var destinations: [String]
}

let cruise: Activity = Trip(…)
```

```
(lldb)
```

```swift
protocol Activity {}
struct Trip: Activity {
    var name: String
    var destinations: [String]
}

let cruise: Activity = Trip(…)
```

```
(lldb) p cruise
(Travel.Trip) $R0 = {
  name = "Mediterranean Cruise"
  destinations = 3 values {
    [0] = "Sorrento"
    [1] = "Capri"
    [2] = "Taormina"
  }
}
```

```
protocol Activity {}
struct Trip: Activity {
    var name: String
    var destinations: [String]
}

let cruise: Activity = Trip(…)
```

```
(lldb)
```

```
protocol Activity {}
struct Trip: Activity {
    var name: String
    var destinations: [String]
}

let cruise: Activity = Trip(…)
```

```
(lldb) p cruise.name
error: <EXPR>:3:8: error: value of type 'Activity' has no member 'name'
(lldb)
```

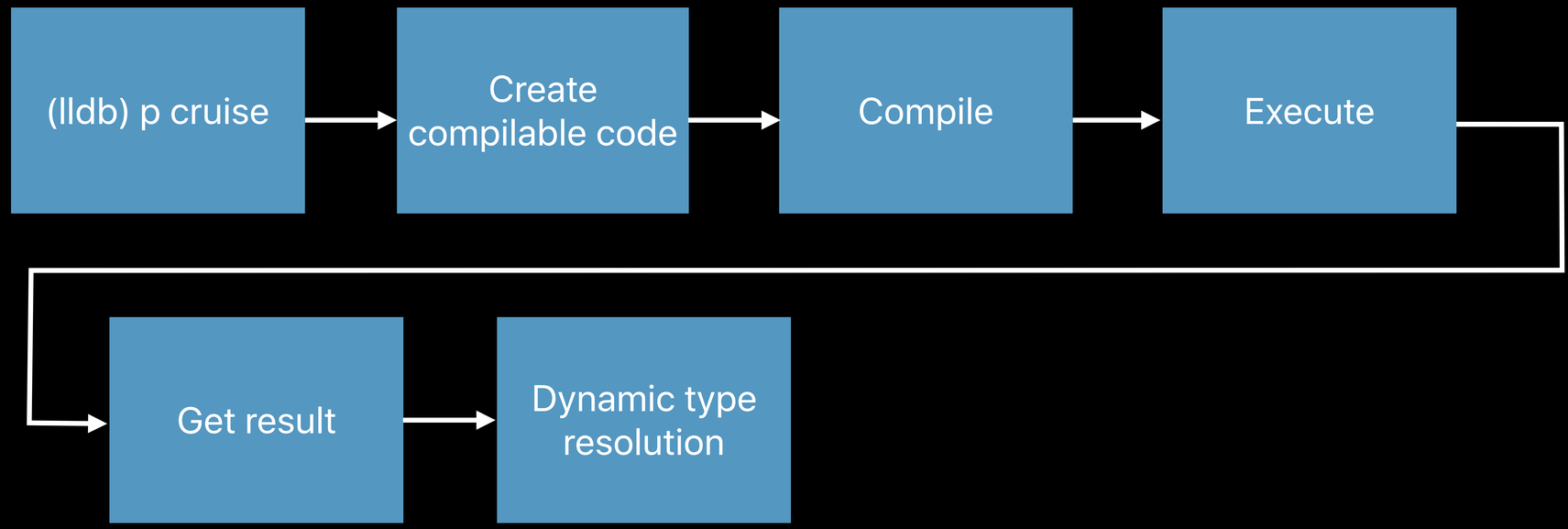
```
protocol Activity {}
struct Trip: Activity {
    var name: String
    var destinations: [String]
}

let cruise: Activity = Trip(…)
let name = cruise.name                    Value of type 'Activity' has no member 'name'
```

```
(lldb) p cruise.name
error: <EXPR>:3:8: error: value of type 'Activity' has no member 'name'
(lldb)
```

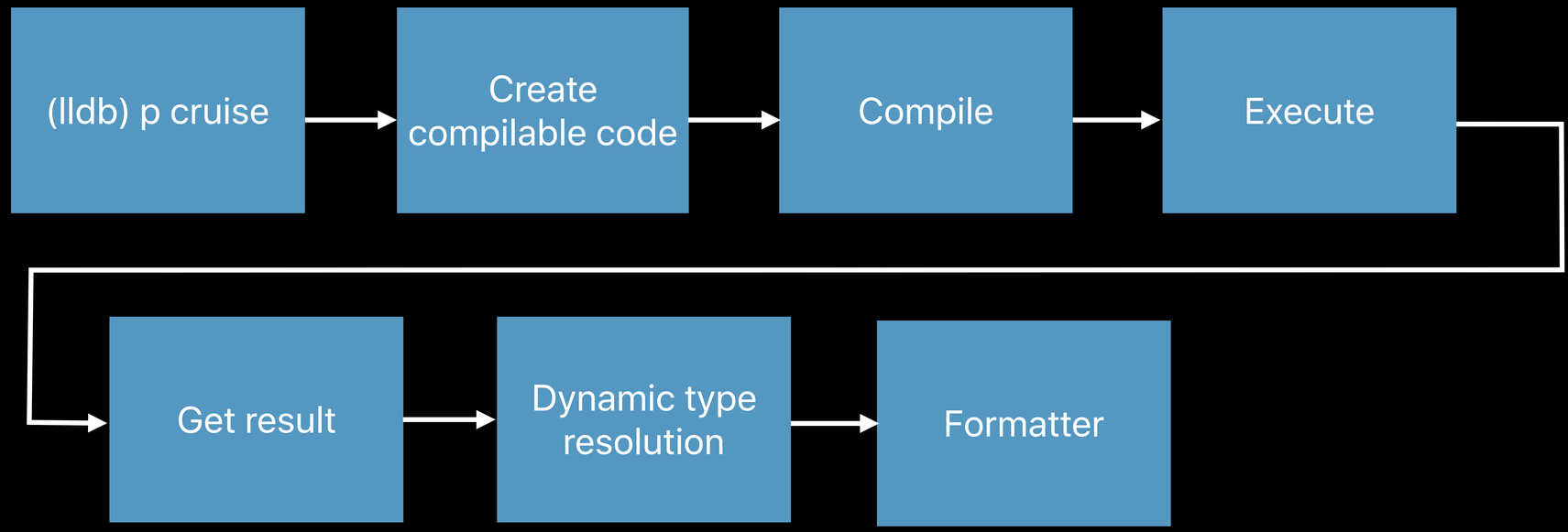
```
protocol Activity {}
struct Trip: Activity {
    var name: String
    var destinations: [String]
}

let cruise: Activity = Trip(…)
let name = cruise.name                    Value of type 'Activity' has no member 'name'
```

```
(lldb) p cruise.name
error: <EXPR>:3:8: error: value of type 'Activity' has no member 'name'
(lldb) p (cruise as! Trip).name
(String) $R0 = "Mediterranean Cruise"
```

“p” Under the Hood



“p” Under the Hood



```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) expression --raw -- cruise.name
(Swift.String) $R0 = {
  _guts = {
    _object = {
      _countAndFlagsBits = {
        _value = 7305804402515733574
      }
      …
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) expression --raw -- cruise.name
(Swift.String) $R0 = {
  _guts = {
    _object = {
      _countAndFlagsBits = {
        _value = 7305804402515733574
      }
      …
(lldb) p cruise.name
(String) $R1 = "Mediterranean Cruise"
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) v cruise
(Travel.Trip) $R0 = {
  name = "Mediterranean Cruise"
  destinations = 3 values {
    [0] = "Sorrento"
    [1] = "Capri"
    [2] = "Taormina"
  }
}
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) frame variable cruise
(Travel.Trip) $R0 = {
  name = "Mediterranean Cruise"
  destinations = 3 values {
    [0] = "Sorrento"
    [1] = "Capri"
    [2] = "Taormina"
  }
}
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) v cruise.name
(String) cruise.name = "Mediterranean Cruise"
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) v cruise.name
(String) cruise.name = "Mediterranean Cruise"
(lldb) p cruise.name.isEmpty()
(Bool) $R2 = false
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) v cruise.name
(String) cruise.name = "Mediterranean Cruise"
(lldb) p cruise.name.isEmpty()
(Bool) $R2 = false
(lldb) po cruise.name.count
8
```

“v” Under the Hood

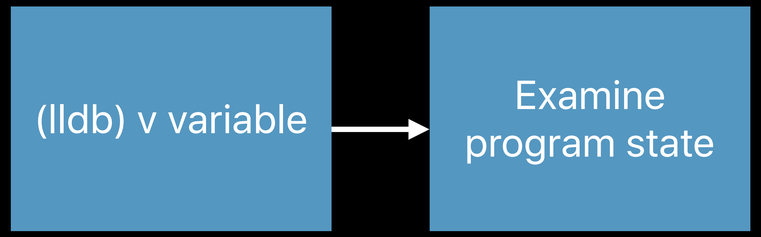
(lldb) v variable

“v” Under the Hood

(lldb) v variable

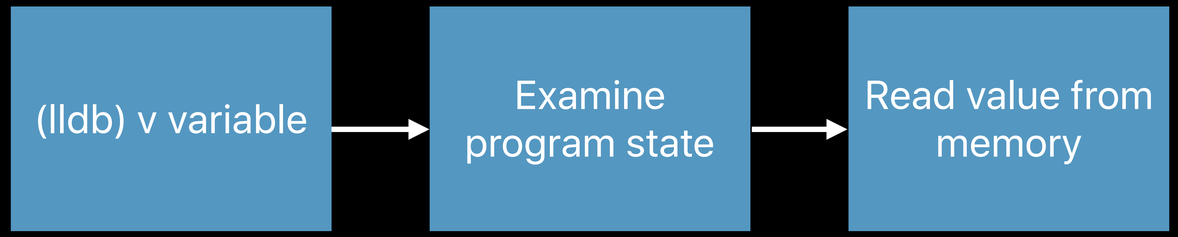
```
(lldb) v variable
```

“v” Under the Hood



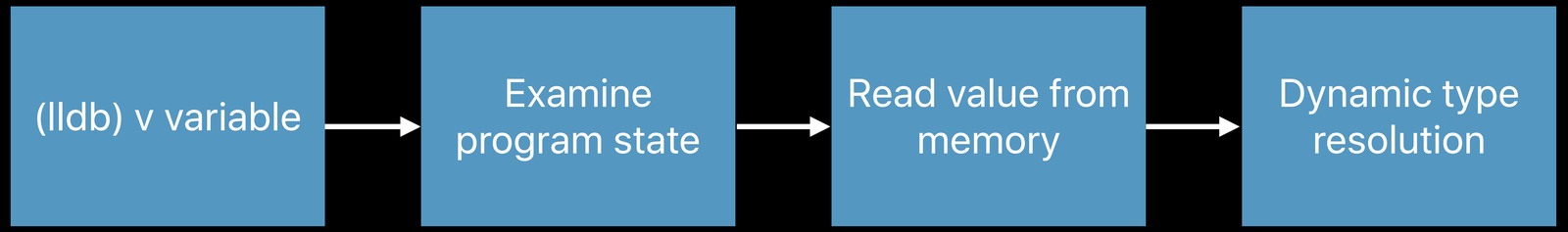

```
(lldb) v variable
```

“v” Under the Hood



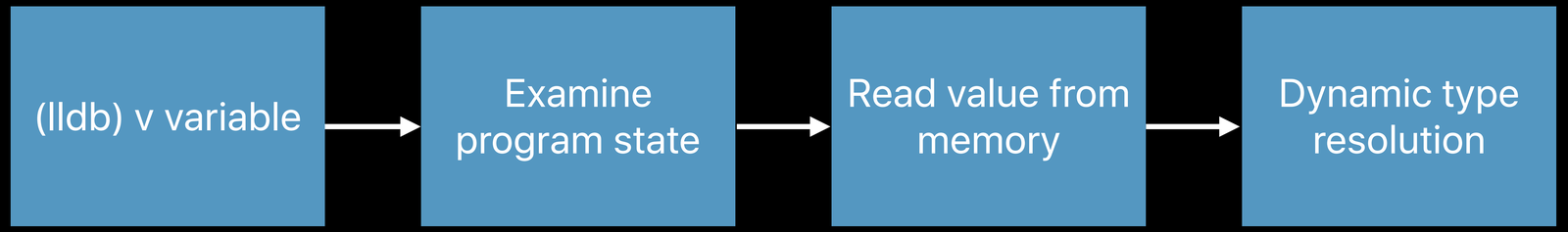

```
(lldb) v variable
```

“v” Under the Hood



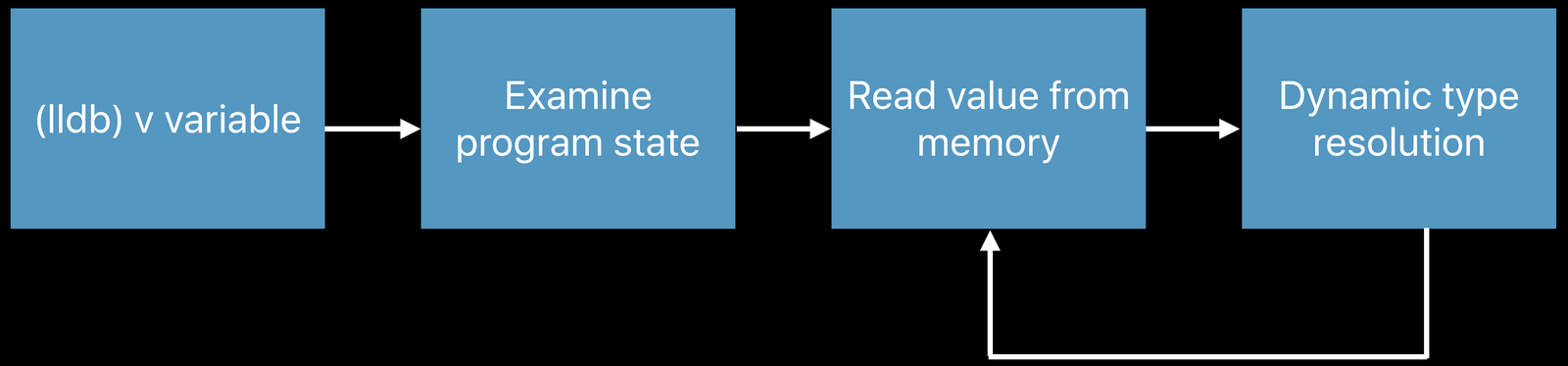

```
(lldb) v variable
```

“v” Under the Hood



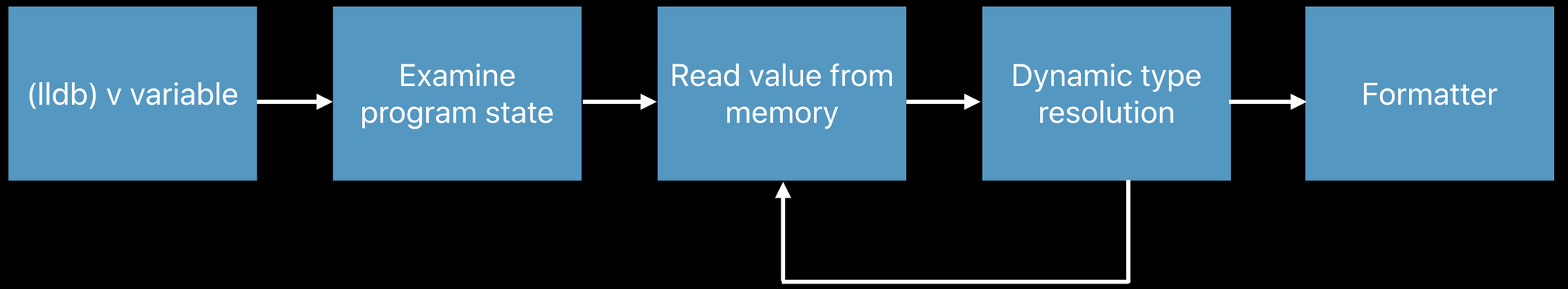

```
(lldb) v variable.field1
```

“v” Under the Hood




```
(lldb) v variable.field1.field2
```

“v” Under the Hood




```
(lldb) v variable.field1.field2
```

```
protocol Activity {}
struct Trip: Activity {
    var name: String
    var destinations: [String]
}

let cruise: Activity = Trip(…)
```

```
(lldb)
```

```
protocol Activity {}
struct Trip: Activity {
    var name: String
    var destinations: [String]
}

let cruise: Activity = Trip(…)
```

```
(lldb) v cruise.name
(String) cruise.name = "Mediterranean Cruise"
```

Displaying Variables

Displaying Variables

	po	p	v
Object description			
Data formatters			

Displaying Variables

	po	p	v
Object description	✓		
Data formatters		✓	✓
Access to the full language	✓	✓	
Iterative dynamic type resolution			✓

Customizing Data Formatters

Jonas Devlieghere

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```swift
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) v cruise.destinations
([String]) cruise.destinations = 3 values {
  [0] = "Sorrento"
  [1] = "Capri"
  [2] = "Taormina"
}
```

Customizing Data Formatters

Filters

String summaries

Synthetic children

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) type filter add Travel.Trip --child name
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) type filter add Travel.Trip --child name
(lldb) v cruise
(Travel.Trip) cruise = (name = "Mediterranean Cruise")
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

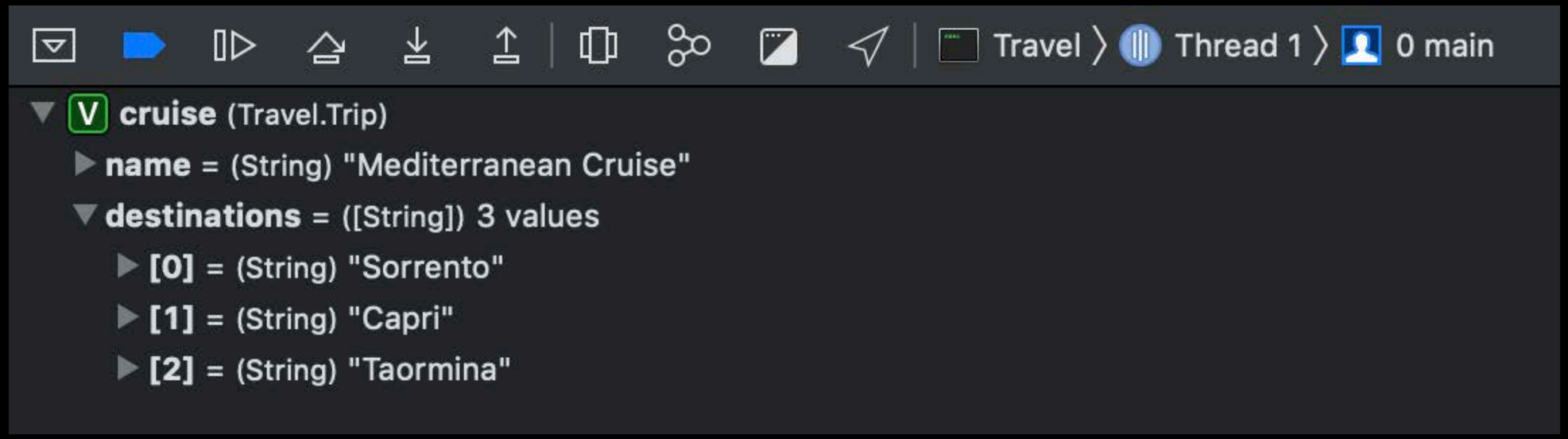
```
(lldb) type filter add Travel.Trip --child name
(lldb) v cruise
(Travel.Trip) cruise = (name = "Mediterranean Cruise")
(lldb) type filter delete Travel.Trip
```

String Summaries

String representation

Displayed in Xcode's variable view

Part of the data formatter

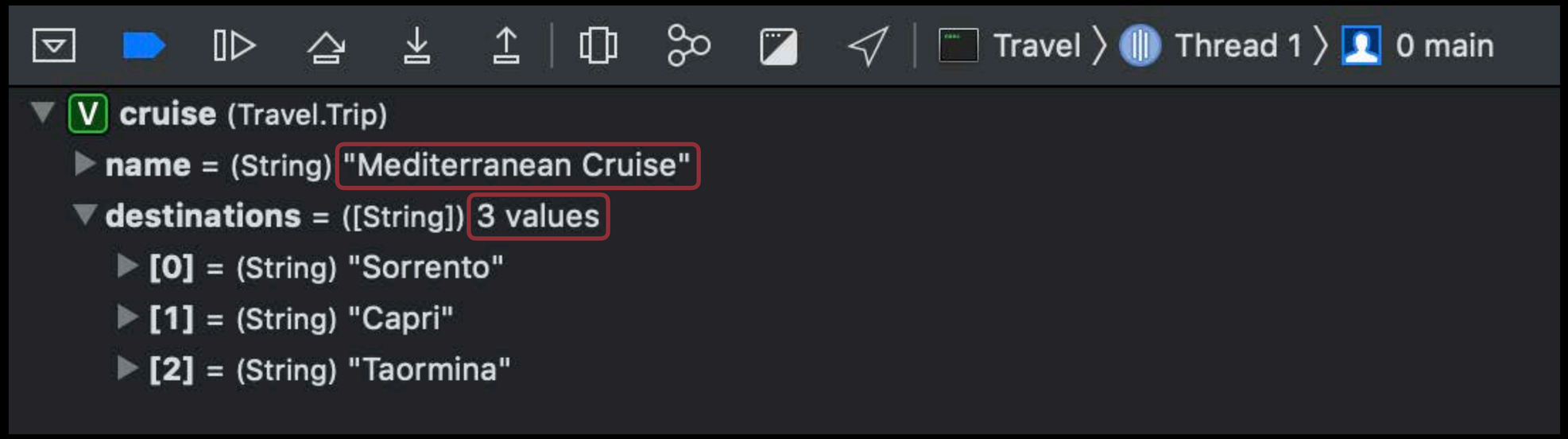


String Summaries

String representation

Displayed in Xcode's variable view

Part of the data formatter



```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) type summary add Travel.Trip --summary-string
"${var.name} from ${var.destinations[0]} to ${var.destinations[2]}"
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) type summary add Travel.Trip --summary-string
"${var.name} from ${var.destinations[0]} to ${var.destinations[2]}"
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) type summary add Travel.Trip --summary-string
"${var.name} from ${var.destinations[0]} to ${var.destinations[2]}"
(lldb) v cruise
(Travel.Trip) cruise = "Mediterranean Cruise" from "Sorrento" to "Taormina"
```

Python Formatter

Arbitrary computations with Python

Full access to LLDB's Python API

LLDB Scripting Bridge

LLDB Scripting Bridge

`SBTarget`	Target program running under the debugger

LLDB Scripting Bridge

`SBTarget`	Target program running under the debugger
`SBProcess`	Process associated with the target program

LLDB Scripting Bridge

`SBTarget`	Target program running under the debugger
`SBProcess`	Process associated with the target program
`SBThread`	Thread of execution

LLDB Scripting Bridge

`SBTarget`	Target program running under the debugger
`SBProcess`	Process associated with the target program
`SBThread`	Thread of execution
`SBFrame`	Stack frame associated with a thread

LLDB Scripting Bridge

`SBTarget`	Target program running under the debugger
`SBProcess`	Process associated with the target program
`SBThread`	Thread of execution
`SBFrame`	Stack frame associated with a thread
`SBValue`	Value of a variable, a register, or an expression

NEW

Python 3

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) script
>>>
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) script
>>> cruise = lldb.frame.FindVariable("cruise")
>>>
```

```swift
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) script
>>> cruise = lldb.frame.FindVariable("cruise")
>>> print(cruise)
(Travel.Trip) cruise = {
  name = "Mediterranean Cruise"
  destinations = 3 values {
    …
}
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
>>>
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
>>> destinations = cruise.GetChildMemberWithName("destinations")
>>>
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
>>> destinations = cruise.GetChildMemberWithName("destinations")
>>> print(destinations)
([String]) destinations = 3 values {
  [0] = "Sorrento"
  [1] = "Capri"
  [2] = "Taormina"
}
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
>>>
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
>>> count = destinations.GetNumChildren()
>>>
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
>>> count = destinations.GetNumChildren()
>>> begin = destinations.GetChildAtIndex(0)
>>> print(begin)
(String) [0] = "Sorrento"
>>>
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
>>> count = destinations.GetNumChildren()
>>> begin = destinations.GetChildAtIndex(0)
>>> print(begin)
(String) [0] = "Sorrento"
>>> end = destinations.GetChildAtIndex(count - 1)
>>> print(end)
(String) [2] = "Taormina"
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
>>>
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
>>> print("Trip from {} to {}".format(begin, end))
Trip from (String) name = "Sorrento" to (String) name = "Taormina"
>>>
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
>>> print("Trip from {} to {}".format(begin, end))
Trip from (String) name = "Sorrento" to (String) name = "Taormina"
>>> print("Trip from {} to {}".format(begin.GetSummary(), end.GetSummary()))
Trip from "Sorrento" to "Taormina"
```

```
// Trip.py
```

```
// Trip.py
def SummaryProvider(value, _):
```

```
// Trip.py
def SummaryProvider(value, _):
    destinations = value.GetChildMemberWithName("destinations")
    count = destinations.GetNumChildren()
```

```
// Trip.py
def SummaryProvider(value, _):
    destinations = value.GetChildMemberWithName("destinations")
    count = destinations.GetNumChildren()
    if count == 0:
        return "Empty trip"
```

```
// Trip.py
def SummaryProvider(value, _):
    destinations = value.GetChildMemberWithName("destinations")
    count = destinations.GetNumChildren()
    if count == 0:
        return "Empty trip"

    begin = destinations.GetChildAtIndex(0).GetSummary()
    end = destinations.GetChildAtIndex(count - 1).GetSummary()

    return "Trip with {} stops from {} to {}".format(count, begin, end)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) command script import Trip.py
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) command script import Trip.py
(lldb) type summary add Travel.Trip --python-function Trip.SummaryProvider
(lldb)
```

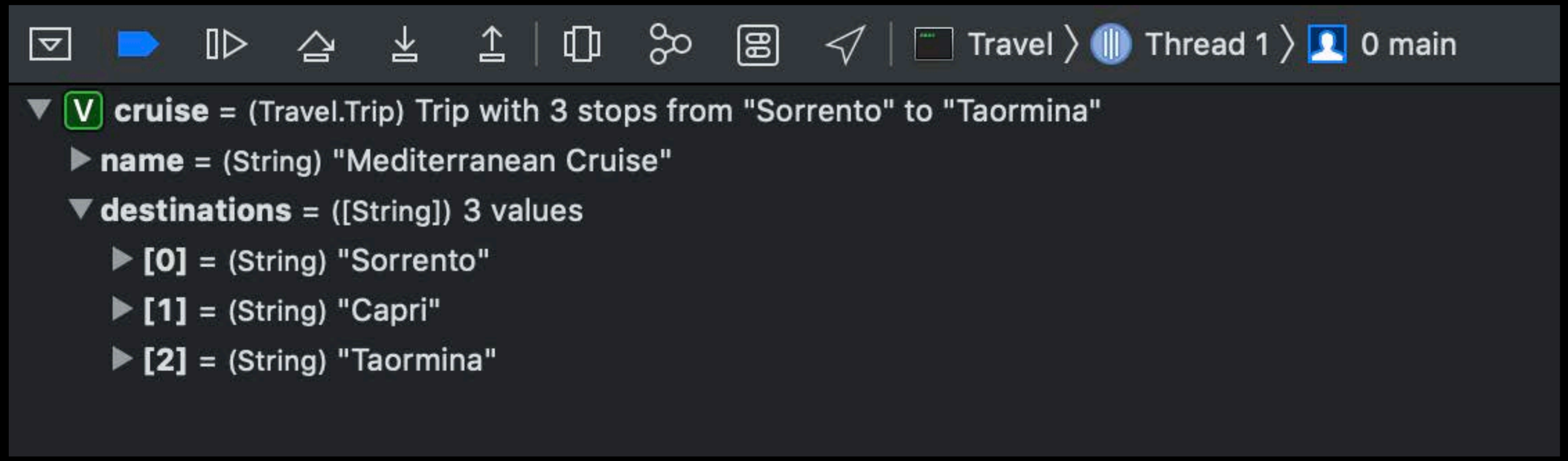
```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) command script import Trip.py
(lldb) type summary add Travel.Trip --python-function Trip.SummaryProvider
(lldb) v cruise
(Travel.Trip) cruise = Trip with 3 stops from "Sorrento" to "Capri"
```

Synthetic Children

Customize children

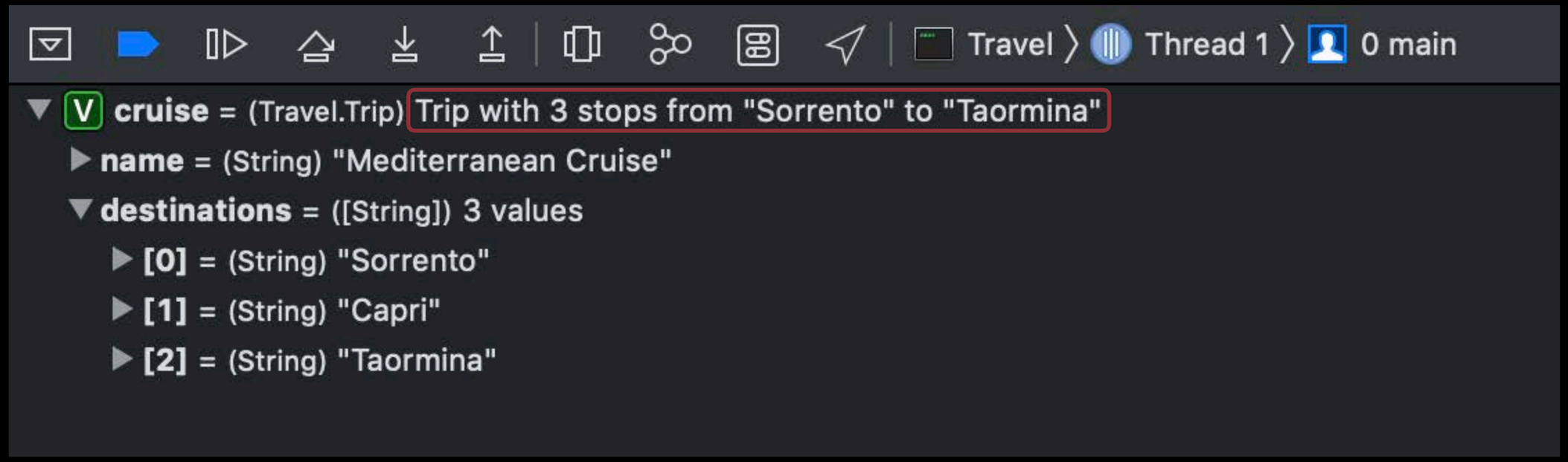
Full access to LLDB's Python API



Synthetic Children

Customize children

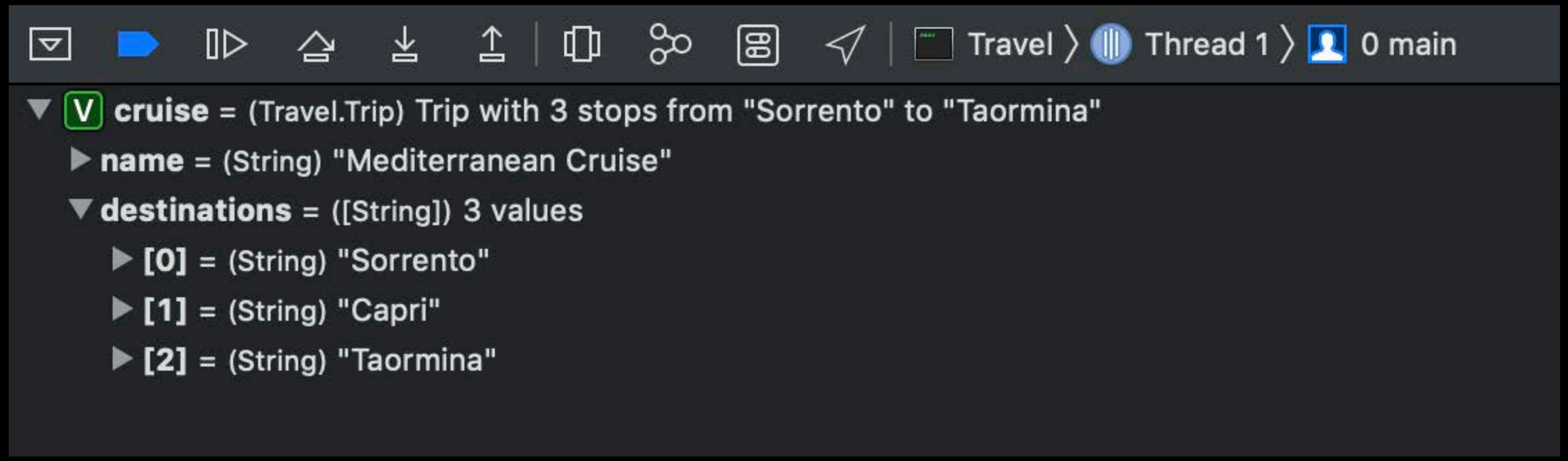
Full access to LLDB's Python API



Synthetic Children

Customize children

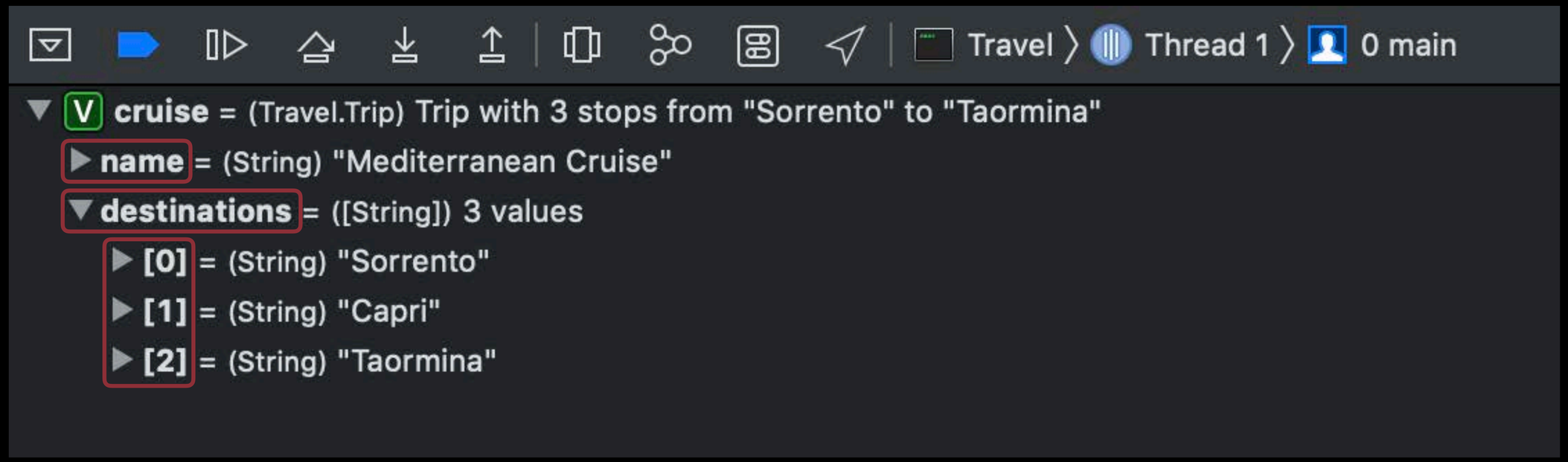
Full access to LLDB's Python API



Synthetic Children

Customize children

Full access to LLDB's Python API



```
// Trip.py
class ExampleSyntheticChildrenProvider:
   def __init__(self, value, _):
      …

   def num_children(self):
      …

   def get_child_at_index(self, index):
      …

   def get_child_index(self, name):
      …
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) command script import Trip.py
(lldb)
```

```
struct Trip {
    var name: String
    var destinations: [String]
}

let cruise = Trip(
    name: "Mediterranean Cruise",
    destinations: ["Sorrento", "Capri", "Taormina"])
```

```
(lldb) command script import Trip.py
(lldb) type synthetic add Travel.Trip --python-class Trip.ExampleSyntheticChildrenProvider
```

```
# ~/.lldbinit

# Load Trip.py
command script import Trip.py

# Register Trip summary provider
type summary add Travel.Trip --python-function Trip.SummaryProvider

# Register Trip child provider
type synthetic add Travel.Trip --python-class Trip.ExampleSyntheticChildrenProvider
```

Summary

Use `v`, `p`, and `po` to print variables

Customize data formatters with filter, summaries and synthetic children

Use Python 3 for scripting

More Information

developer.apple.com/wwdc19/429


WWDC19