Objective-C Memory Management

Data Types

Type	Size	Range
short	16bit	-32768 to 32768
int	32bit	-2147483648 to 2147483648
long	64bit	really big

Data Types

Type	Size	Range
`unsigned short`	16bit	0 to 65536
`unsigned int`	32bit	0 to 4294967296
`unsigned long`	64bit	really big

Data Types

Type	Size
`short`	16bit
`int`	32bit
`long`	64bit
`float`	32bit
`double`	64bit
`char`	8bit

Static Variable Declarations

```
int ageInYears = 5;

long reallyBigNumber = 1123094098234;

unsigned int itemCount;
```

Static Variable Declarations

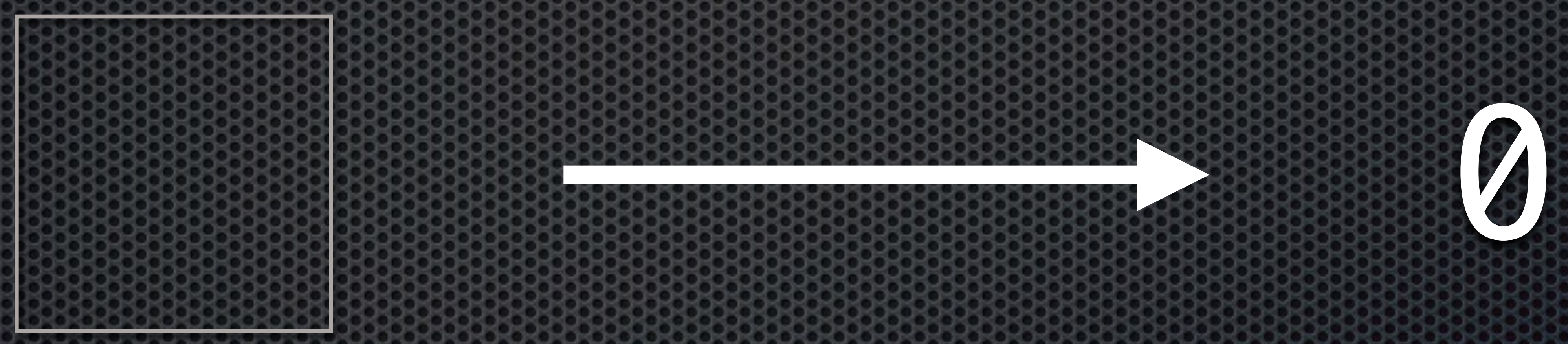
```
float objectWeight = 5.13f;

double veryPreciseWeight = 123.123089123;
```

Pointer Variable Declarations

```
NSString *myName;

NSArray *itemList;

NSDate *today;

NSNumber *someNumber;
```

Pointers

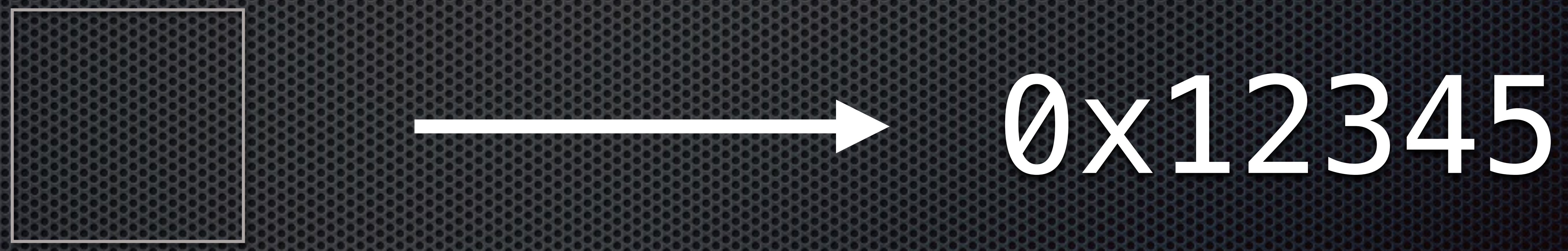
```
NSString *myName;
```



0

Pointers

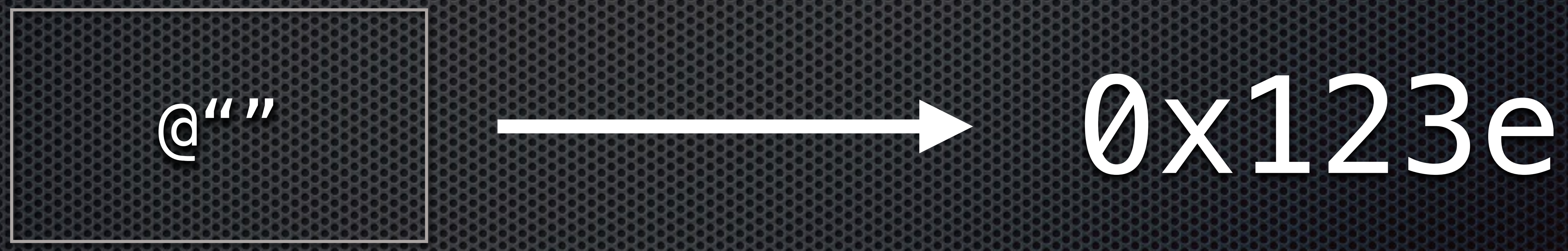
```
NSString *myName = 0x12345;
```



0x12345

Pointers

```
NSString *myName = [[NSString alloc] init];
```

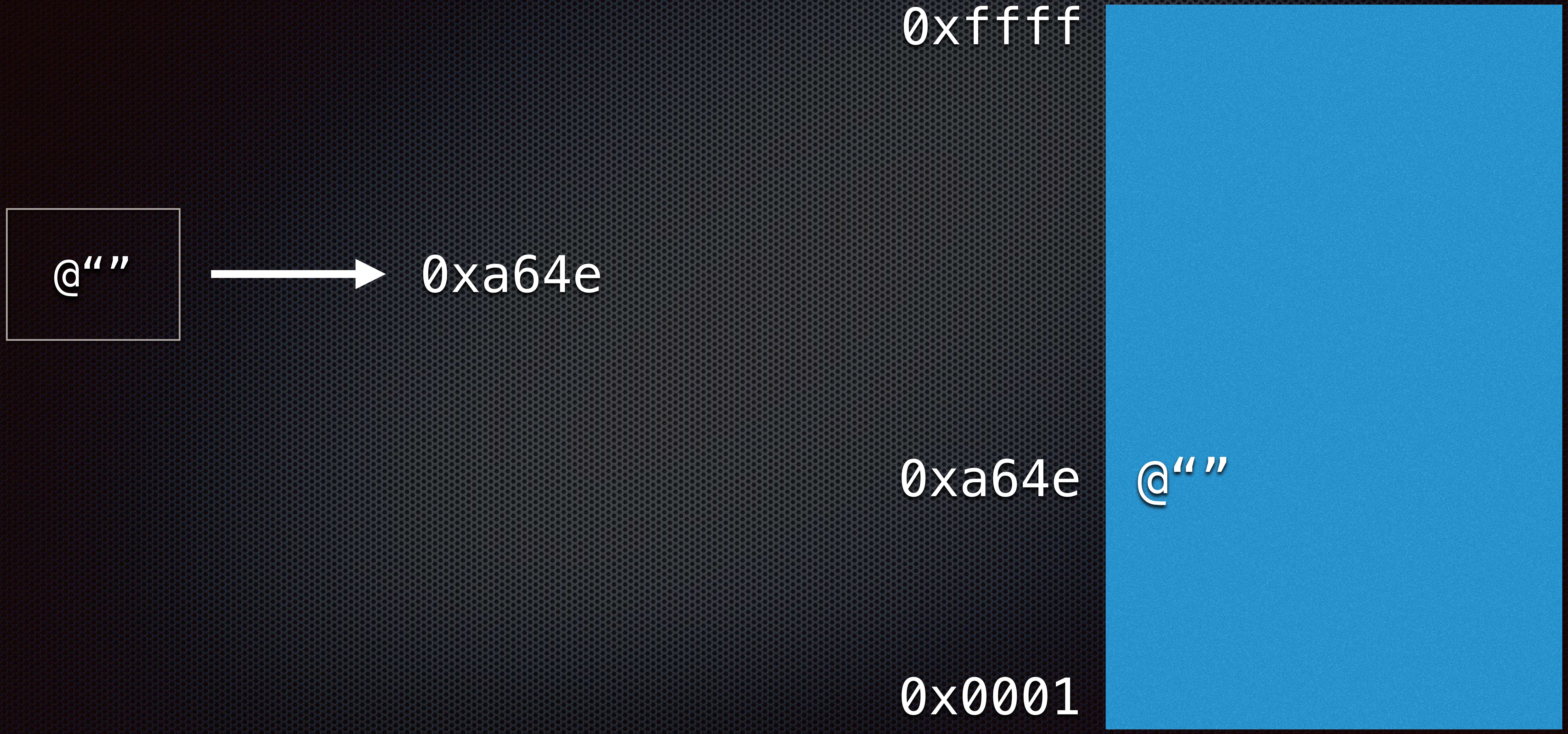


@“”

0x123e

Pointers

Heap



@“”

0xa64e

0xffff

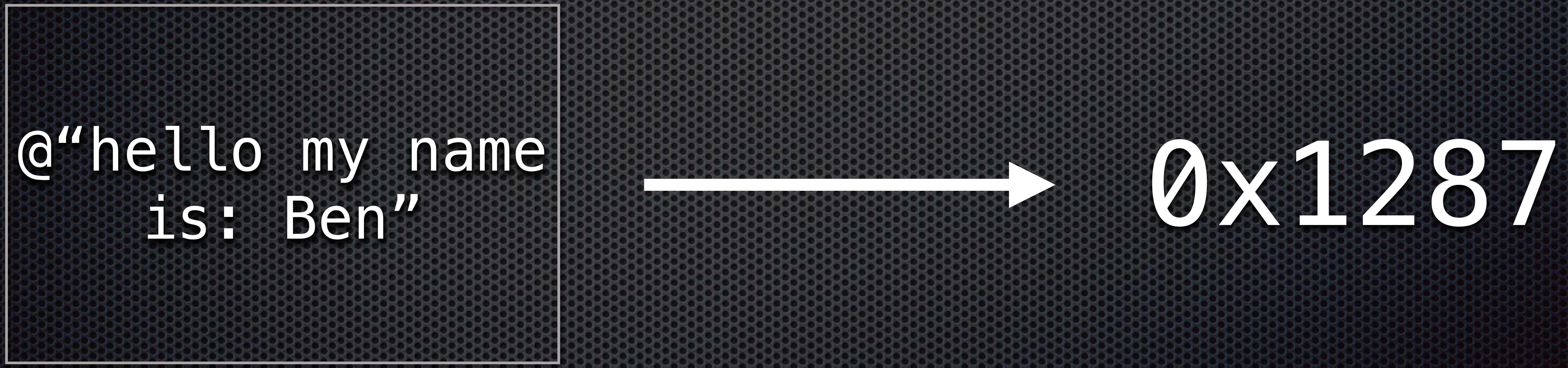
0xa64e

@“”

0x0001

Pointers

```
NSString *myName = [[NSString alloc]
         initWithFormat:@“hello my name is: %@“, name];
```

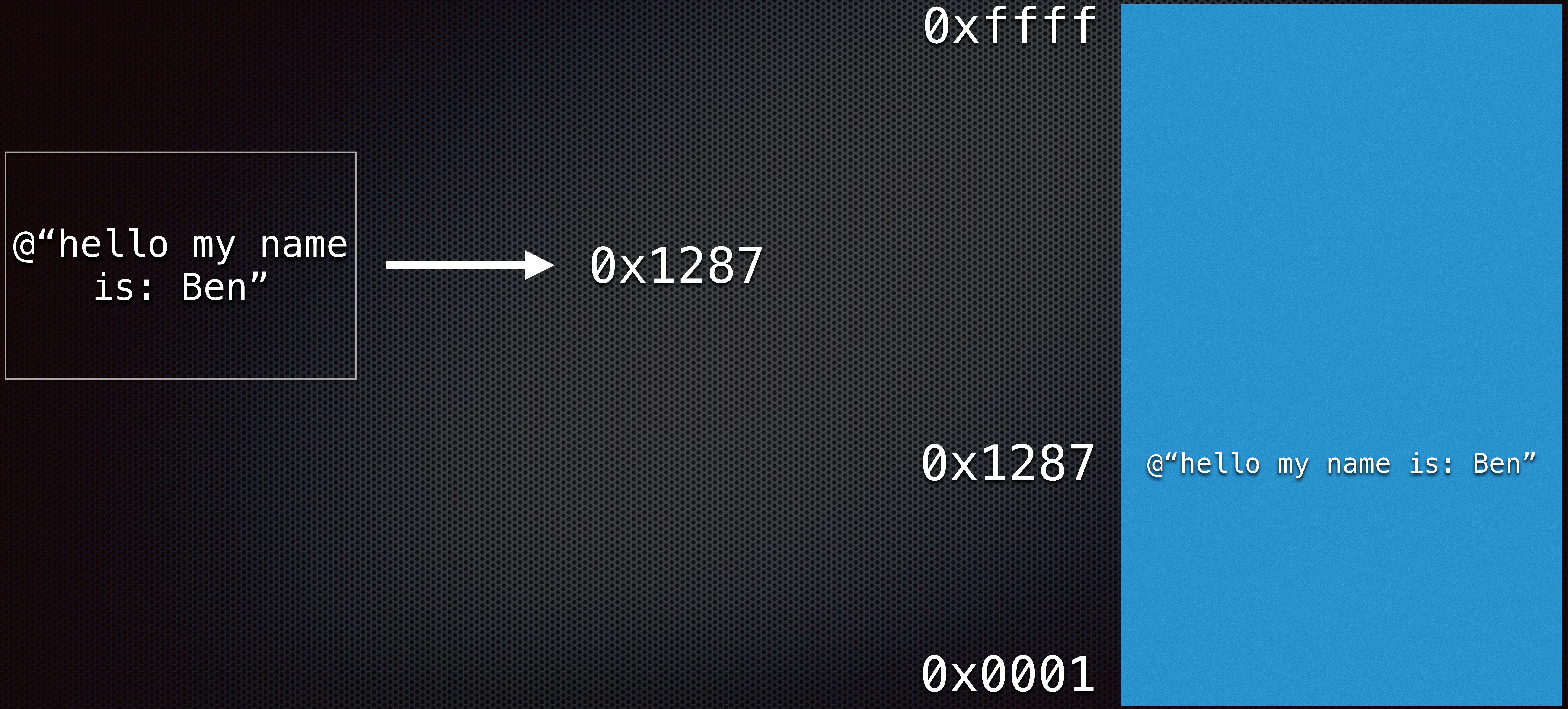


@“hello my name is: Ben”

0x1287

Pointers

Heap



@“hello my name is: Ben”

0x1287

0xffff

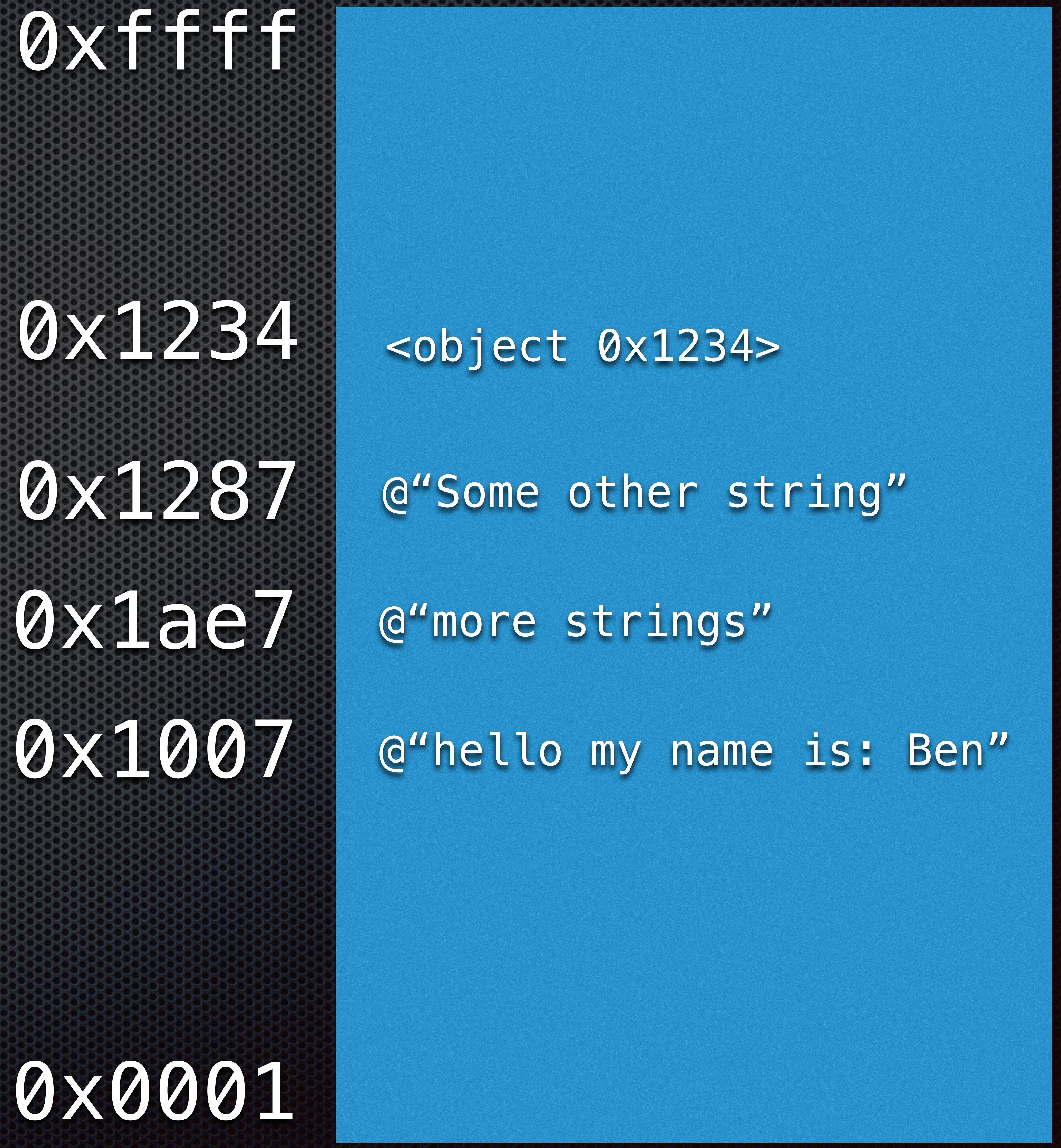
0x1287

@“hello my name is: Ben”

0x0001

Pointers

Heap



0xffff

0x1234 <object 0x1234>

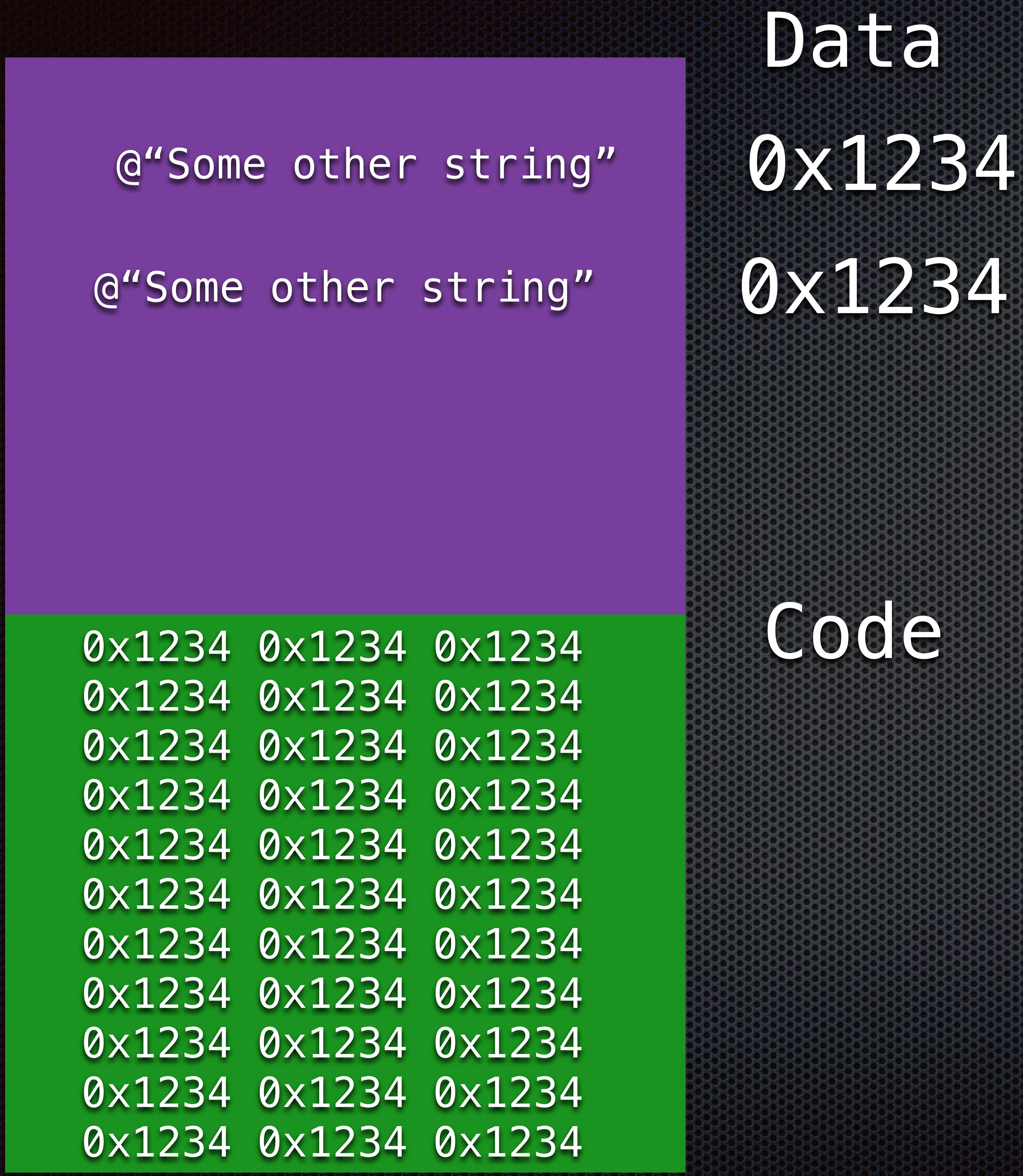
0x1287 @“Some other string”

0x1ae7 @“more strings”

0x1007 @“hello my name is: Ben”

0x0001

Executable



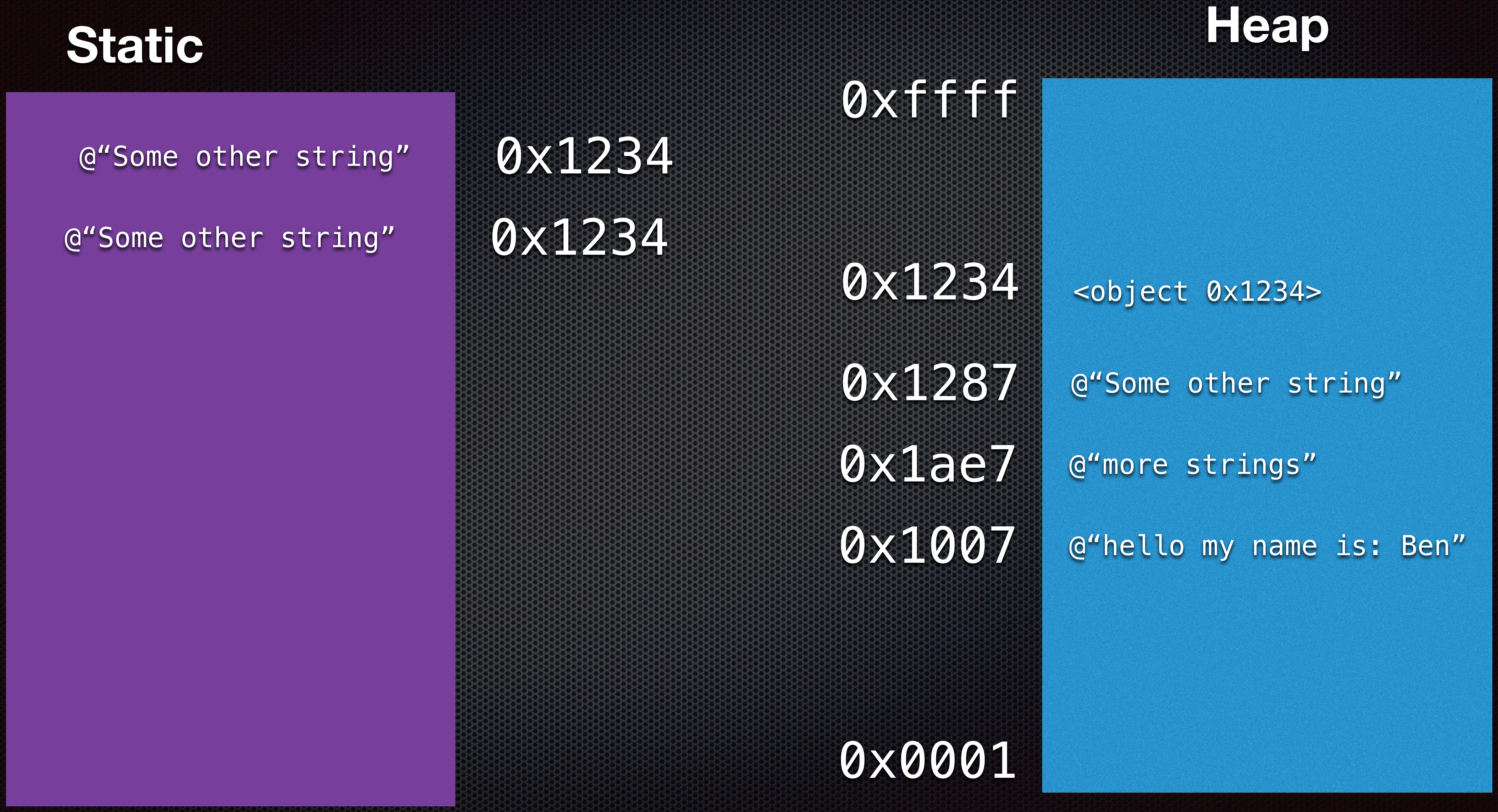
Data

@“Some other string” 0x1234

@“Some other string” 0x1234

Code

```
0x1234 0x1234 0x1234
0x1234 0x1234 0x1234
0x1234 0x1234 0x1234
0x1234 0x1234 0x1234
0x1234 0x1234 0x1234
0x1234 0x1234 0x1234
0x1234 0x1234 0x1234
0x1234 0x1234 0x1234
0x1234 0x1234 0x1234
0x1234 0x1234 0x1234
0x1234 0x1234 0x1234
```



Static

@“Some other string” 0x1234

@“Some other string” 0x1234

Heap

0xffff

0x1234 <object 0x1234>

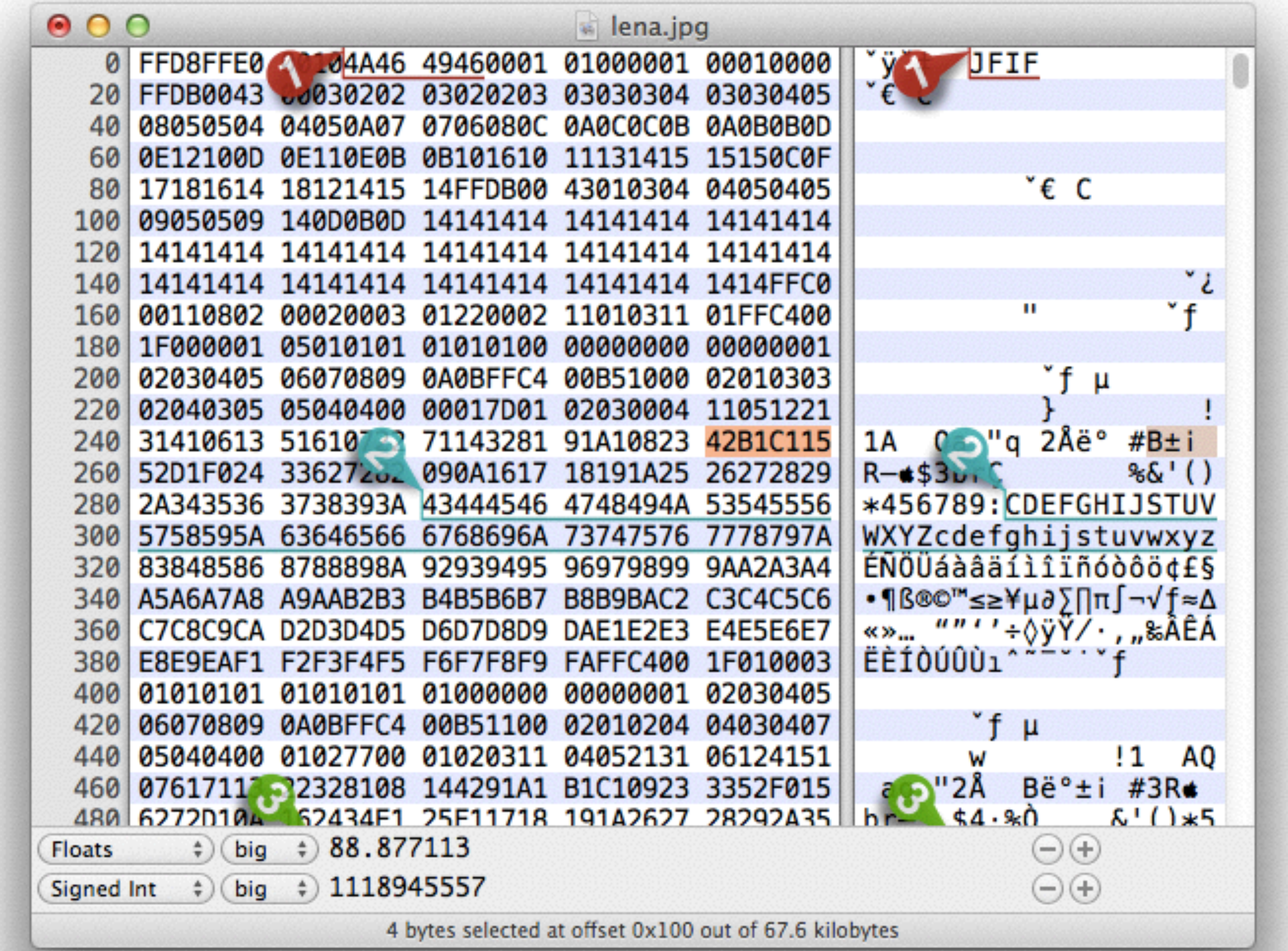
0x1287 @“Some other string”

0x1ae7 @“more strings”

0x1007 @“hello my name is: Ben”

0x0001

Static



lena.jpg

```
  0 FFD8FFE0 00104A46 49460001 01000001 00010000  ˇÿˇ‡  JFIF
 20 FFDB0043 00030202 03020203 03030304 03030405  ˇ€ C
 40 08050504 04050A07 0706080C 0A0C0C0B 0A0B0B0D
 60 0E12100D 0E110E0B 0B101610 11131415 15150C0F
 80 17181614 18121415 14FFDB00 43010304 04050405          ˇ€ C
100 09050509 140D0B0D 14141414 14141414 14141414
120 14141414 14141414 14141414 14141414 14141414
140 14141414 14141414 14141414 14141414 1414FFC0                     ˇ¿
160 00110802 00020003 01220002 11010311 01FFC400          "         ˇƒ
180 1F000001 05010101 01010100 00000000 00000001
200 02030405 06070809 0A0BFFC4 00B51000 02010303           ˇƒ µ
220 02040305 05040400 00017D01 02030004 11051221           }         !
240 31410613 51610722 71143281 91A10823 42B1C115  1A   Qa  "q 2Åë°  #B±i
260 52D1F024 33627282 090A1617 18191A25 26272829  R—$3br  C        %&'()
280 2A343536 3738393A 43444546 4748494A 53545556  *456789:CDEFGHIJSTUV
300 5758595A 63646566 6768696A 73747576 7778797A  WXYZcdefghijstuvwxyz
320 83848586 8788898A 92939495 96979899 9AA2A3A4  ÉÑÖÜáàâäíìîïñóòôö¢£§
340 A5A6A7A8 A9AAB2B3 B4B5B6B7 B8B9BAC2 C3C4C5C6  •¶ß®©™≤≥¥µ∂∑∏π∫¬√ƒ≈∆
360 C7C8C9CA D2D3D4D5 D6D7D8D9 DAE1E2E3 E4E5E6E7  «»… “”‘’÷◊ÿŸ/·‚„‰ÂÊÁ
380 E8E9EAF1 F2F3F4F5 F6F7F8F9 FAFFC400 1F010003  ËÈÍÒÚÛÙı˜¯˘˙ˇƒ
400 01010101 01010101 01000000 00000001 02030405
420 06070809 0A0BFFC4 00B51100 02010204 04030407         ˇƒ µ
440 05040400 01027700 01020311 04052131 06124151          w         !1   AQ
460 07617113 22328108 144291A1 B1C10923 3352F015    aq "2Å   Bë°±i  #3R
480 6272D10A 162434E1 25F11718 191A2627 28292A35  br—  $4·%Ò     &'()*5
```

Floats big 88.877113

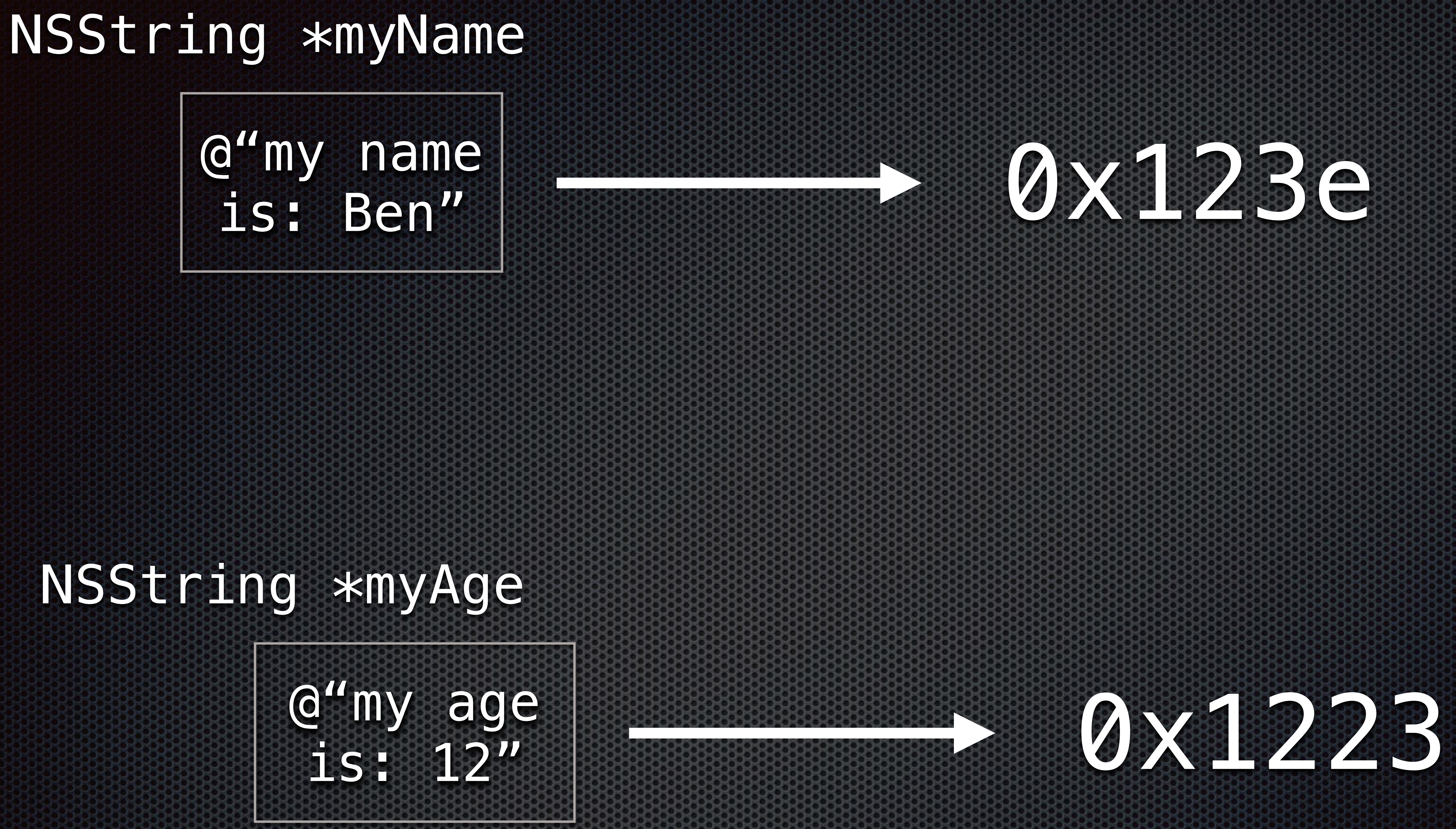
Signed Int big 1118945557

4 bytes selected at offset 0x100 out of 67.6 kilobytes

Pointers

```
NSString *myName = [[NSString alloc]
          initWithFormat:@“hello my name is: %@“, name];
```

```
NSString *myAge = [[NSString alloc]
          initWithFormat:@“hello my age is: %d“, age];
```



NSString *myName

@“my name is: Ben”

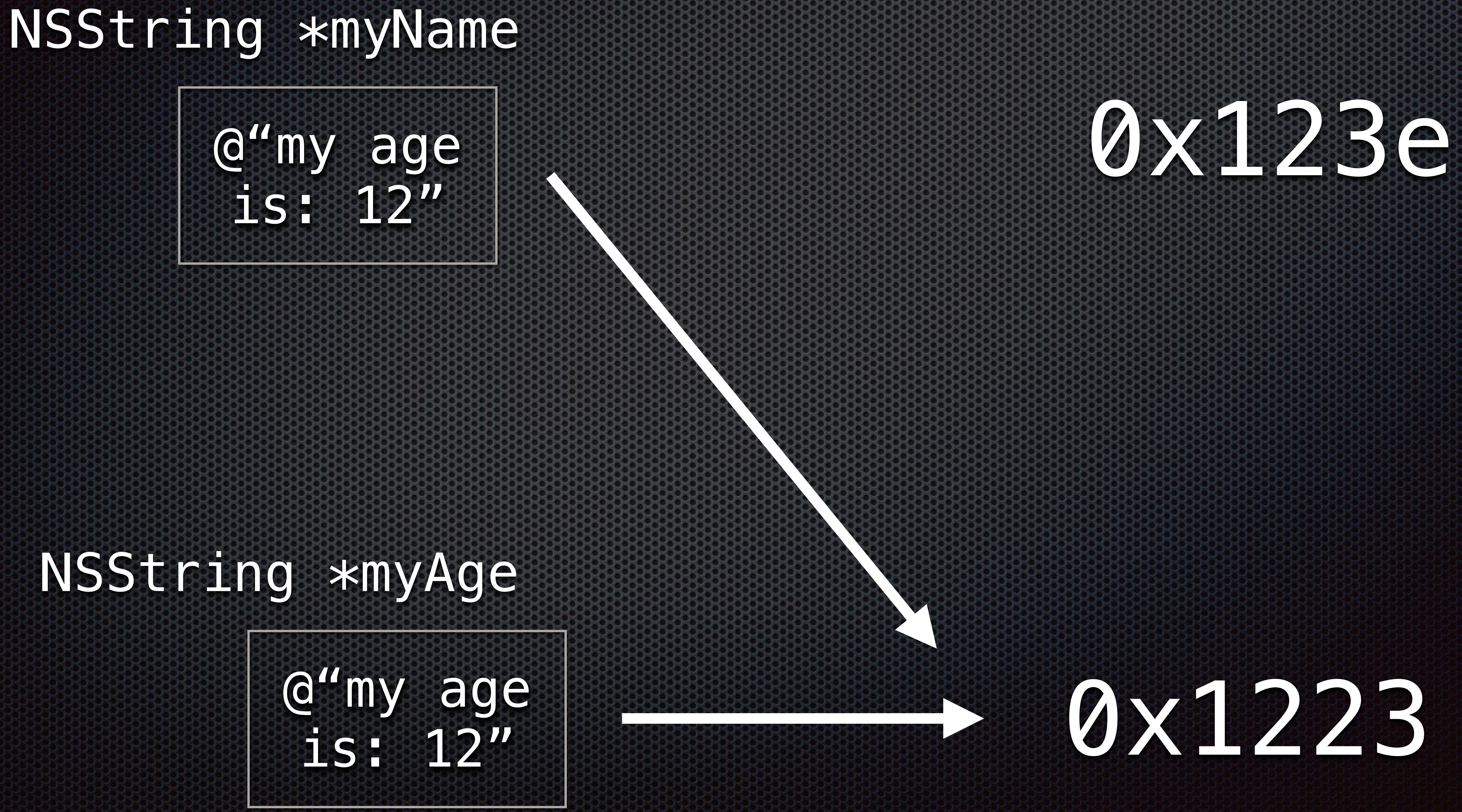
0x123e

NSString *myAge

@“my age is: 12”

0x1223

```
myName = myAge;
```



NSString *myName

@“my age is: 12”

0x123e

NSString *myAge

@“my age is: 12”

0x1223

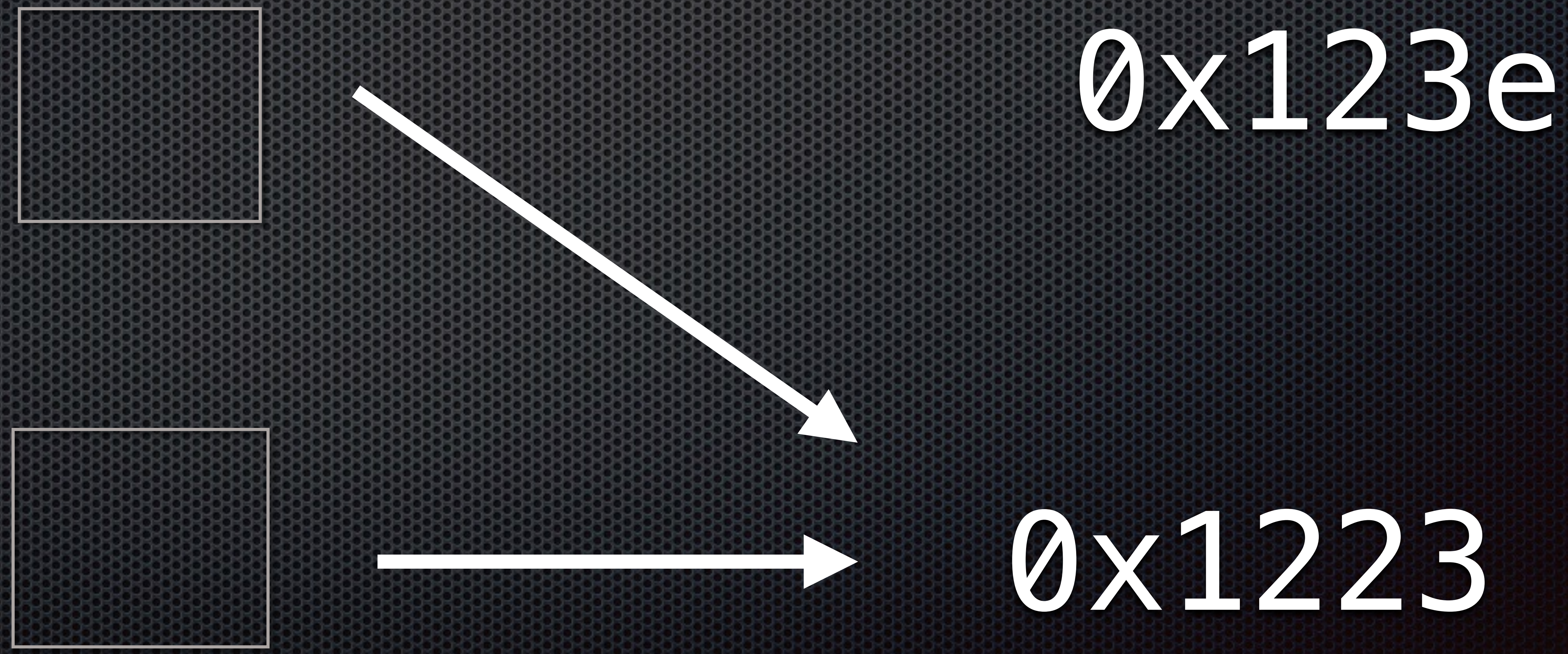
Memory Management Models

- Garbage Collected
- Reference Counting: (i.e. retain and release)
- Manual

Garbage Collected

myName is cleaned up when the garbage collector runs

```
myName = myAge;
```

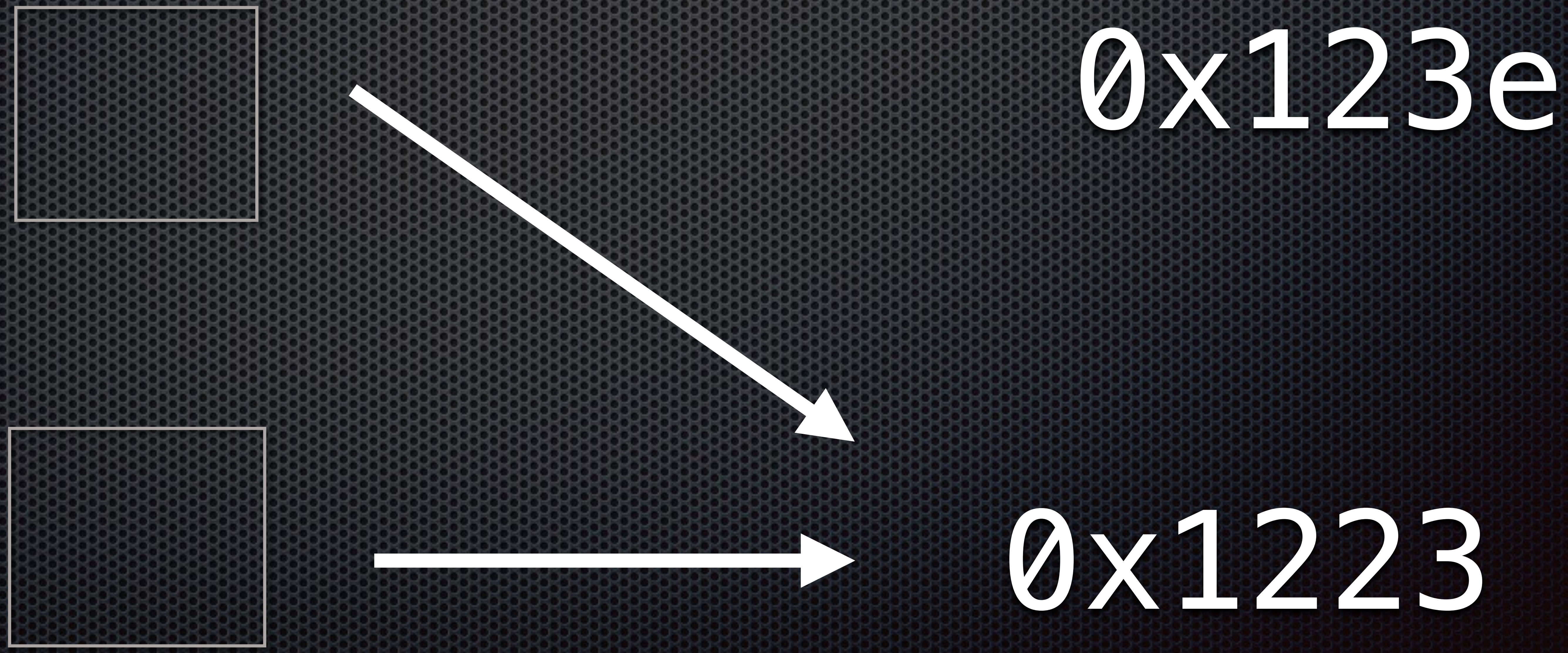


0x123e

0x1223

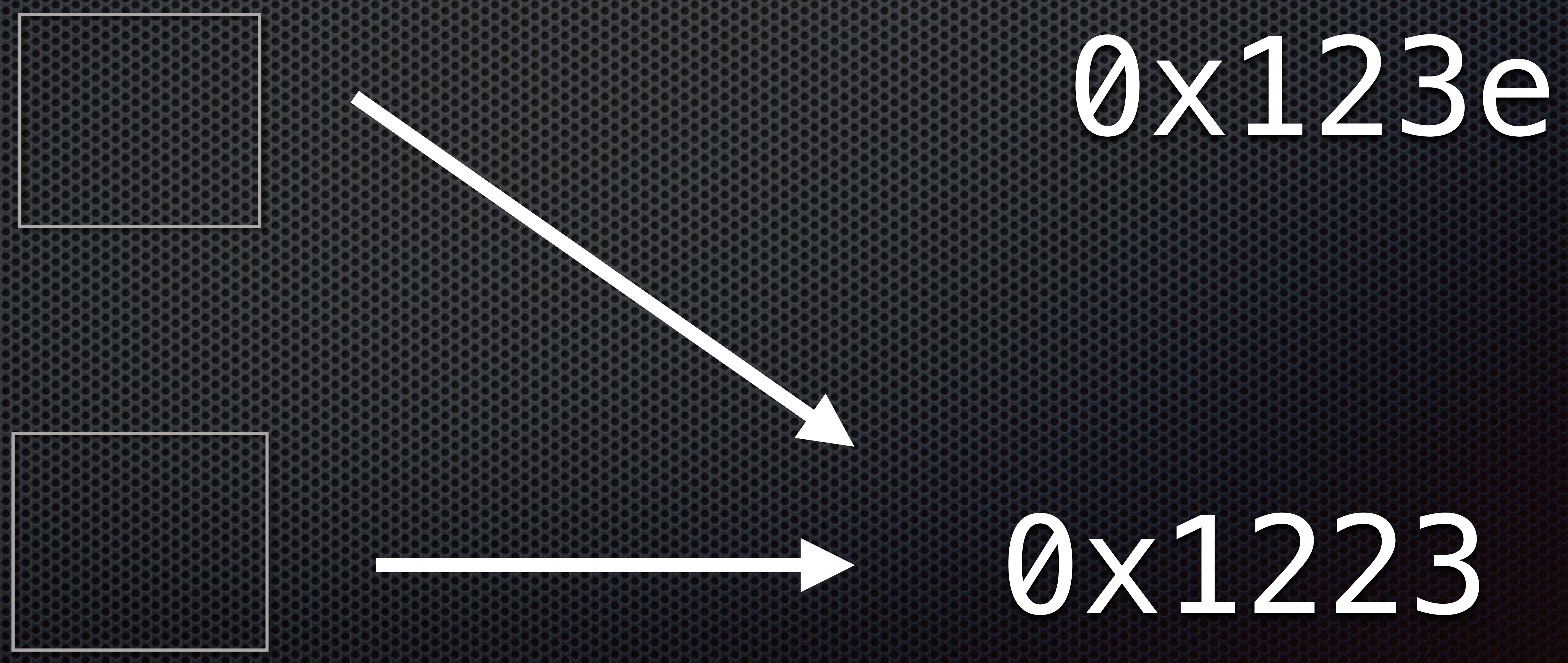
Reference Counting

```
[myName release];

myName = myAge;
```



Manual Memory Management

```
free(myName);

myName = myAge;
```



0x123e

0x1223

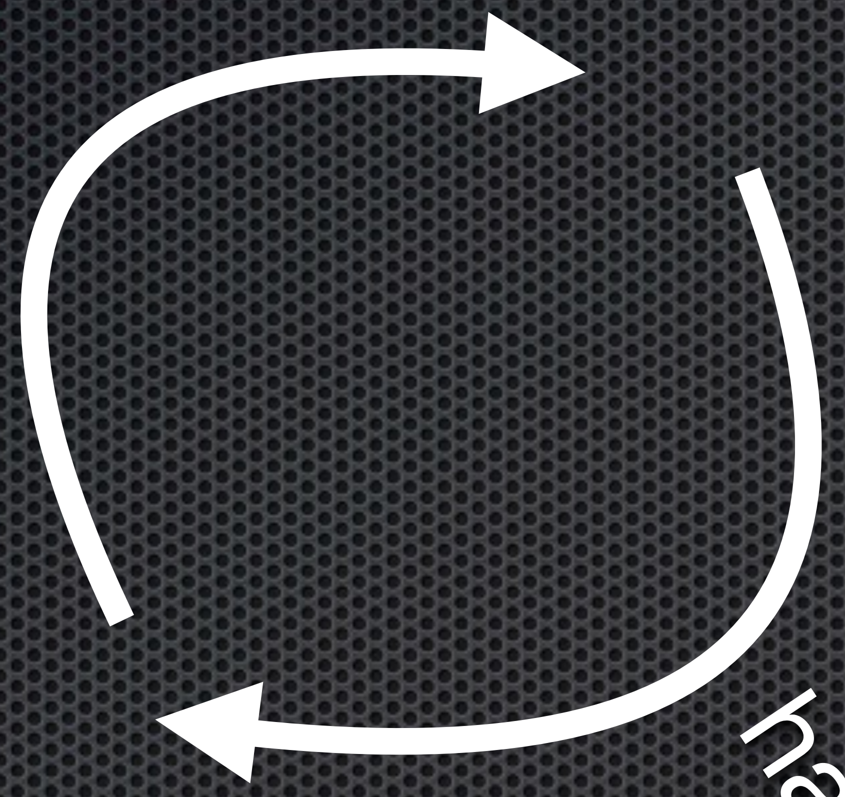
Retain Count

```
NSString *myName = [[NSString alloc] init];

[myName retain];   // retain count 2

[myName release]; // retain count 1

[myName release]; // retain count 0
```

Manual Memory Management

```
char *myName = malloc(sizeof(char)*20);

addChars(myName, @“ben”);

free(myName);
```

Run Loop



Handle events

handle clicks/taps

garbage collect

Autorelease

```
// retain count 1
NSString *myName = [[NSString alloc] init];

[myName autorelease];   // retain count 1
```

```
@autoreleasepool {




// at end of pool
// release is called once for each
// autorelease command
[myName release];

}
```