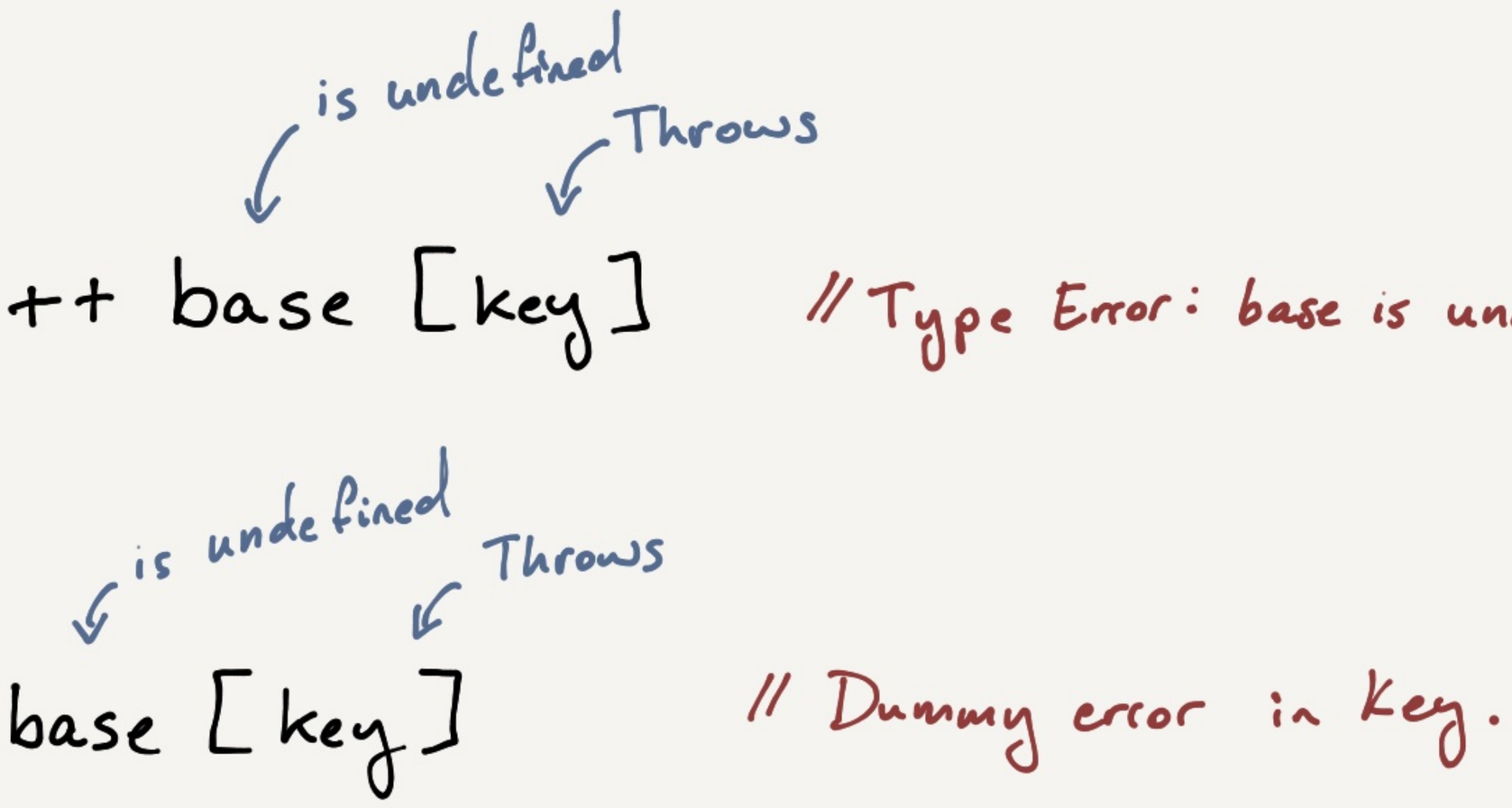
# Review Time


is undefined
Throws
++ base [key]
// Type Error: base is undefined.


is undefined
Throws
base [key]
// Dummy error in key.

# The problem ...

```
0000x:   Check Obj Coercibe   ← wrong spot!
000xx:   Get GName "x"
xxxxx:   ...
xxxxx:   To Property key
```

# Our Bytecode was in the wrong order...

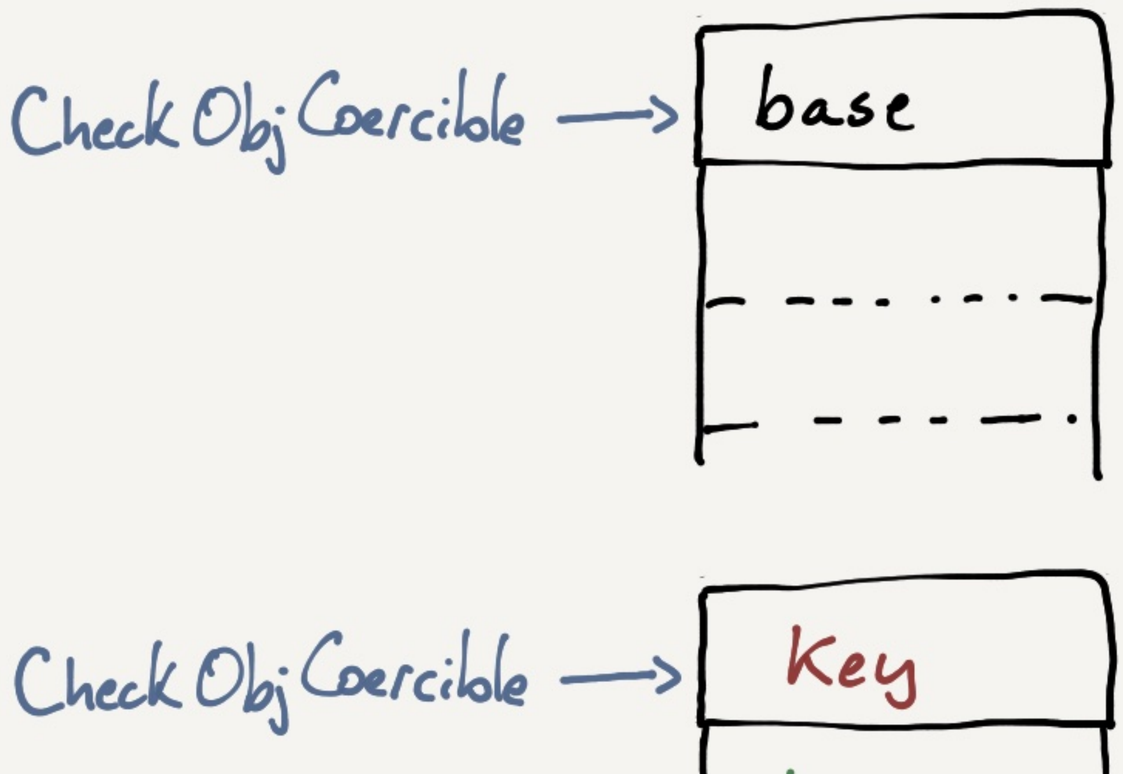
```
000xx:   Get GName "x"
xxxxx:   ...
xxxxx:   Check Obj Coercible
xxxxx:   To Property key
```

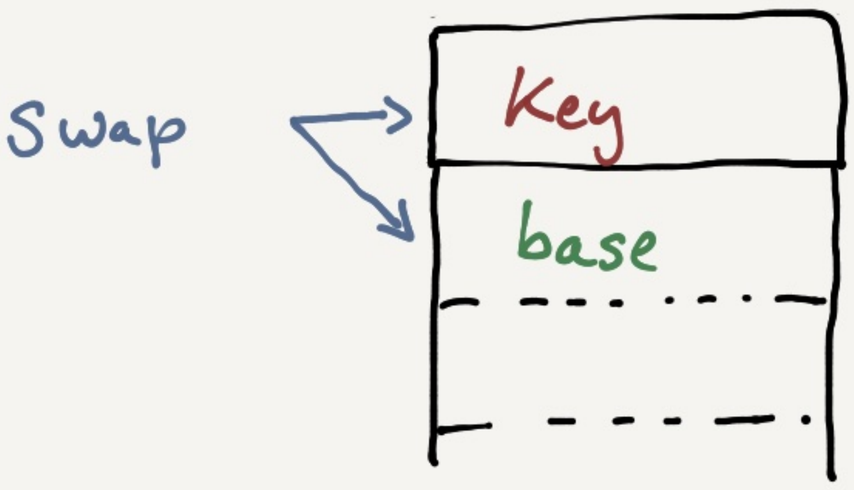
000xx: Get GName "x"

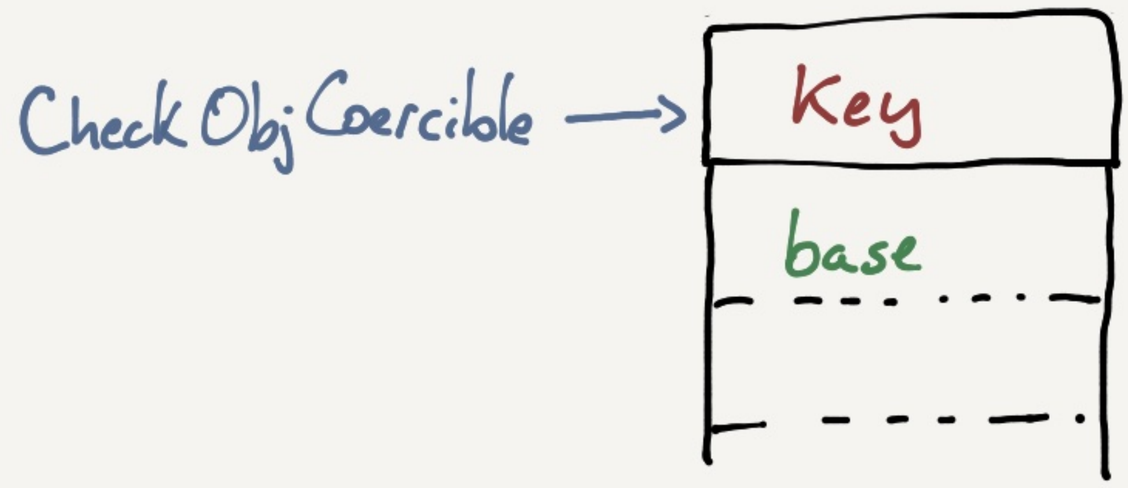
xxxxx: ...

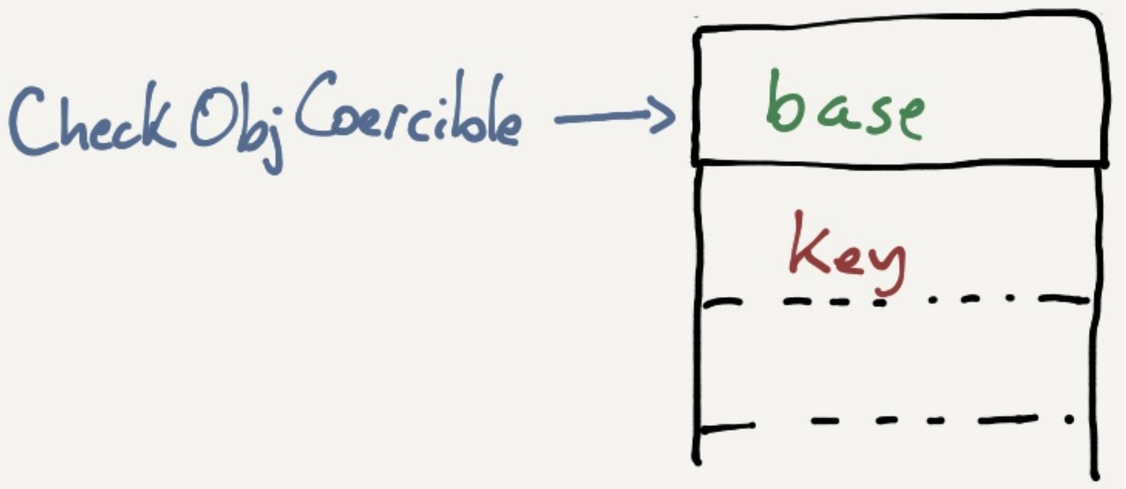
xxxxx: Check Obj Coercible

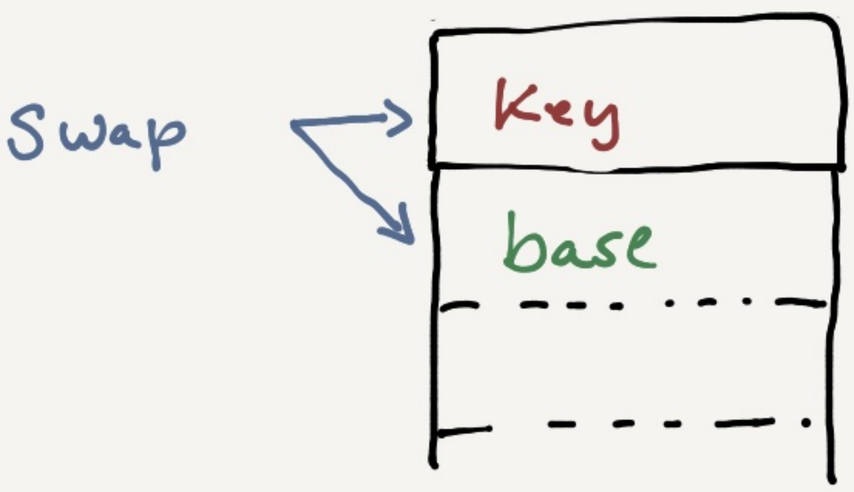
xxxxx: To Property key

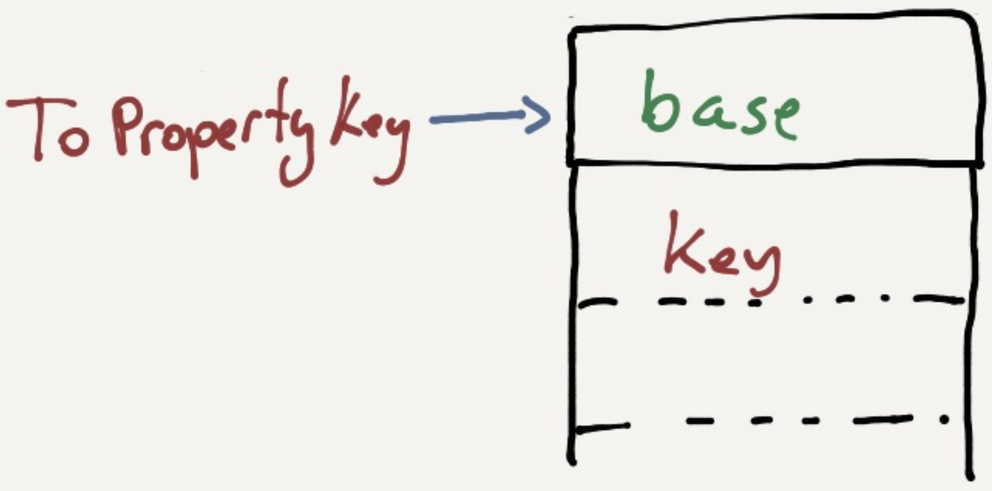


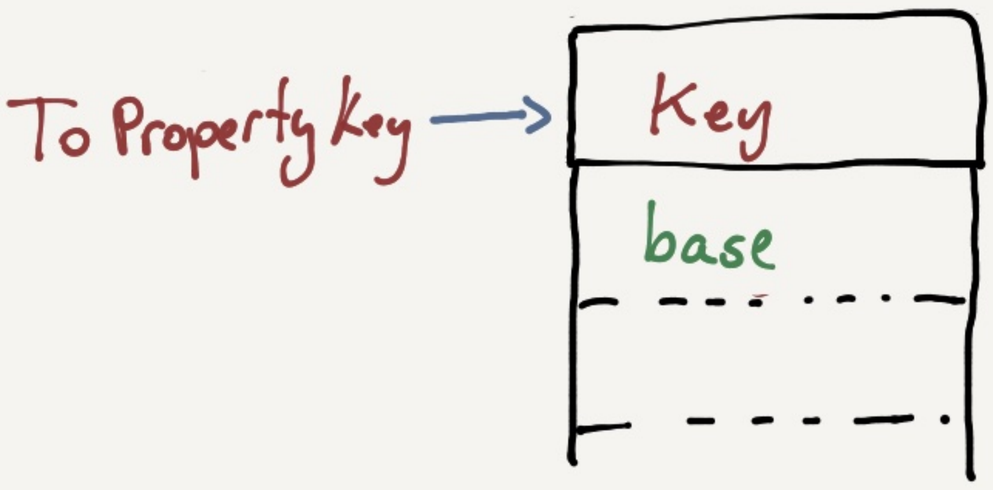



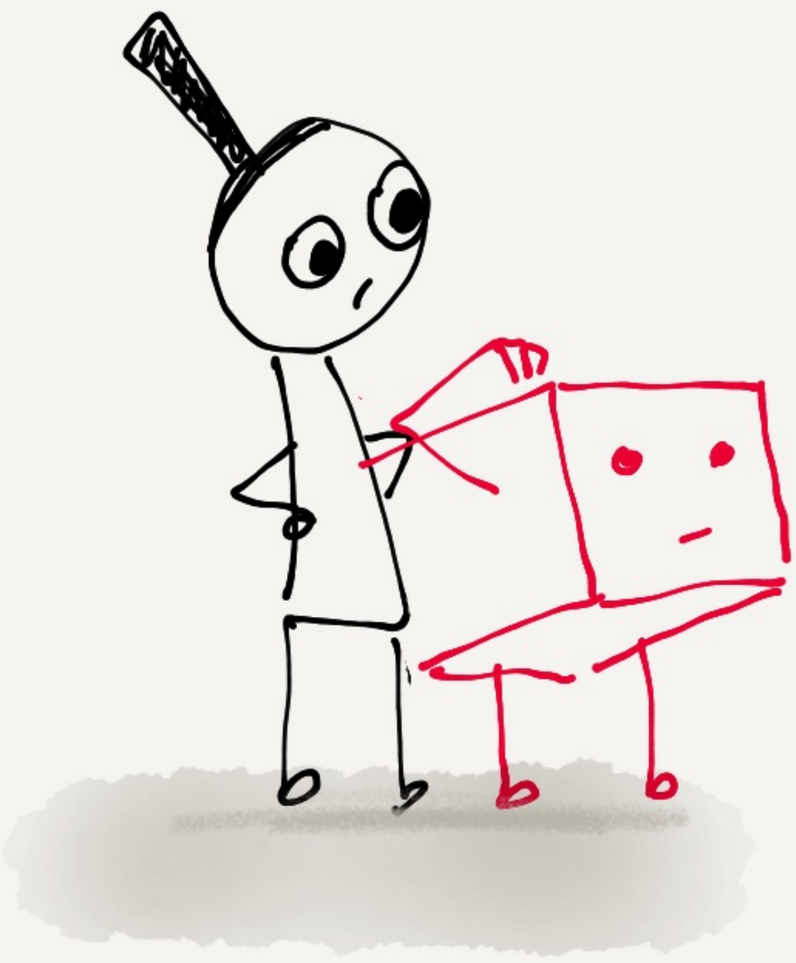
```
000xx:   Get GName "x"
xxxxx:   ...
xxxxx:   Check Obj Coercible   ← doesn't work
xxxxx:   To Property key
```

```
000xx:   Get GName "x"
xxxxx:   ...
xxxxx:   Swap
xxxxx:   Check Obj Coercible
xxxxy:   Swap
xxxxx:   To Property key
```

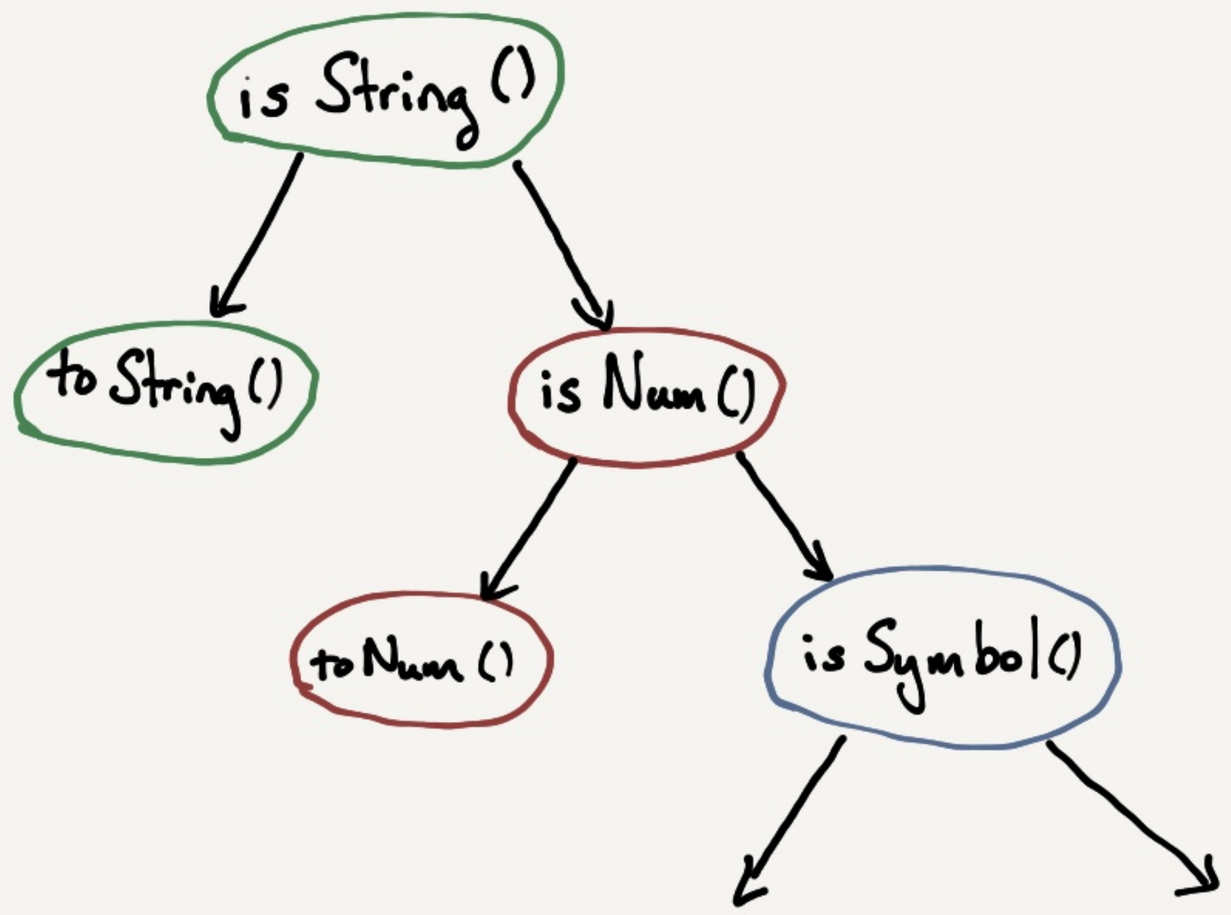
} 3 byte codes? Works, but not Ideal.

```
000xx:   Get GName "x"
xxxxx:   ...
xxxxx:   * Prepare Set Elem *   ← New Byte Code?
```

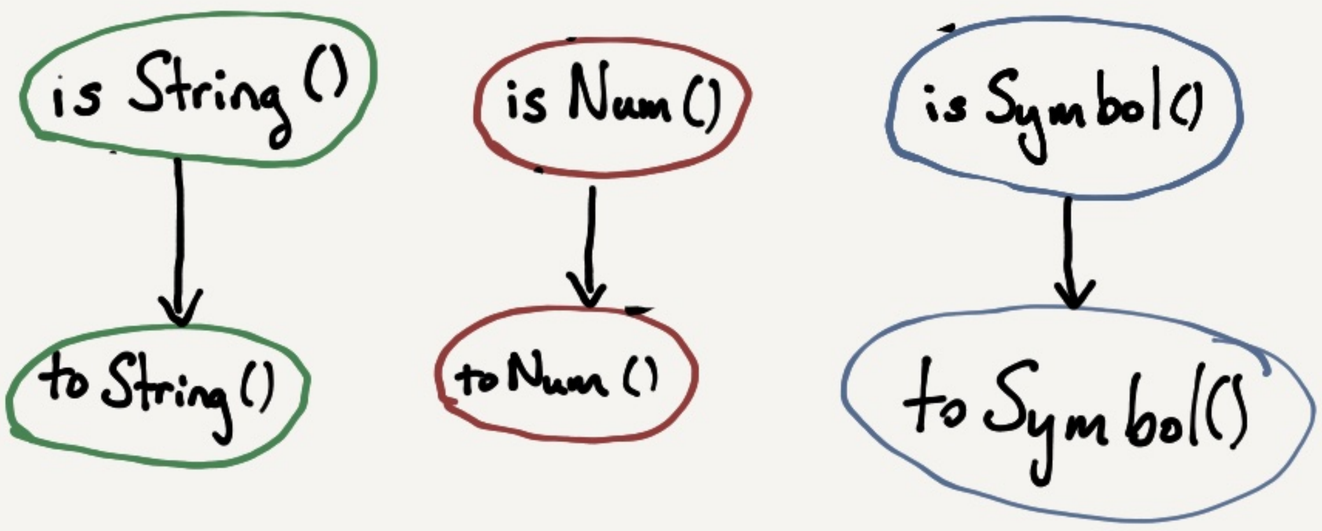
# How do we make This fast?



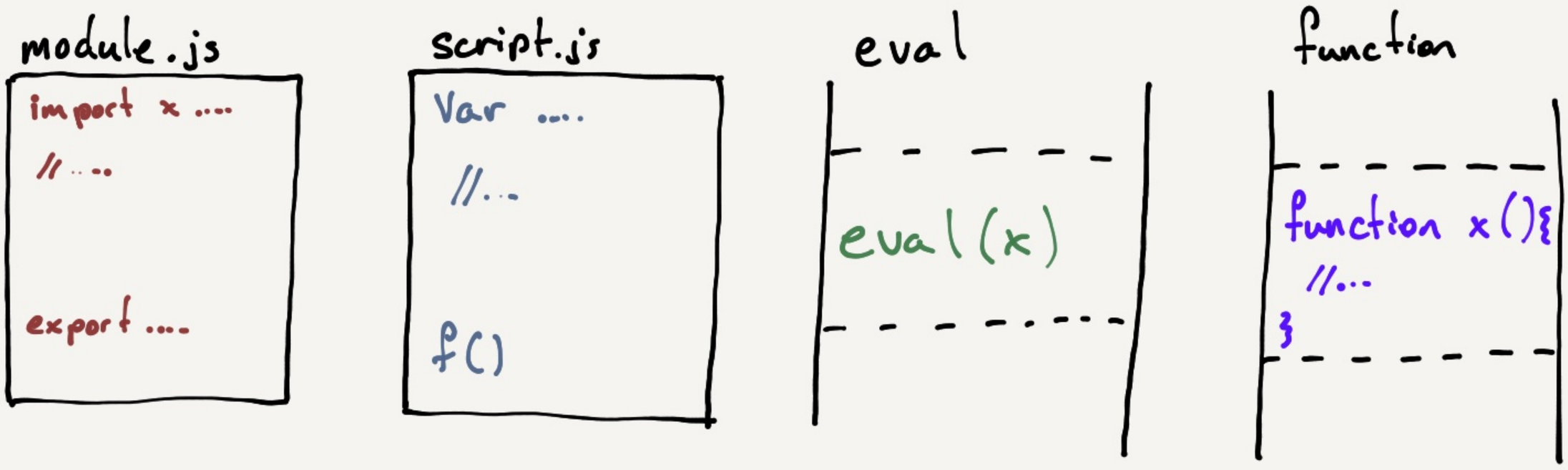
f [<something>]



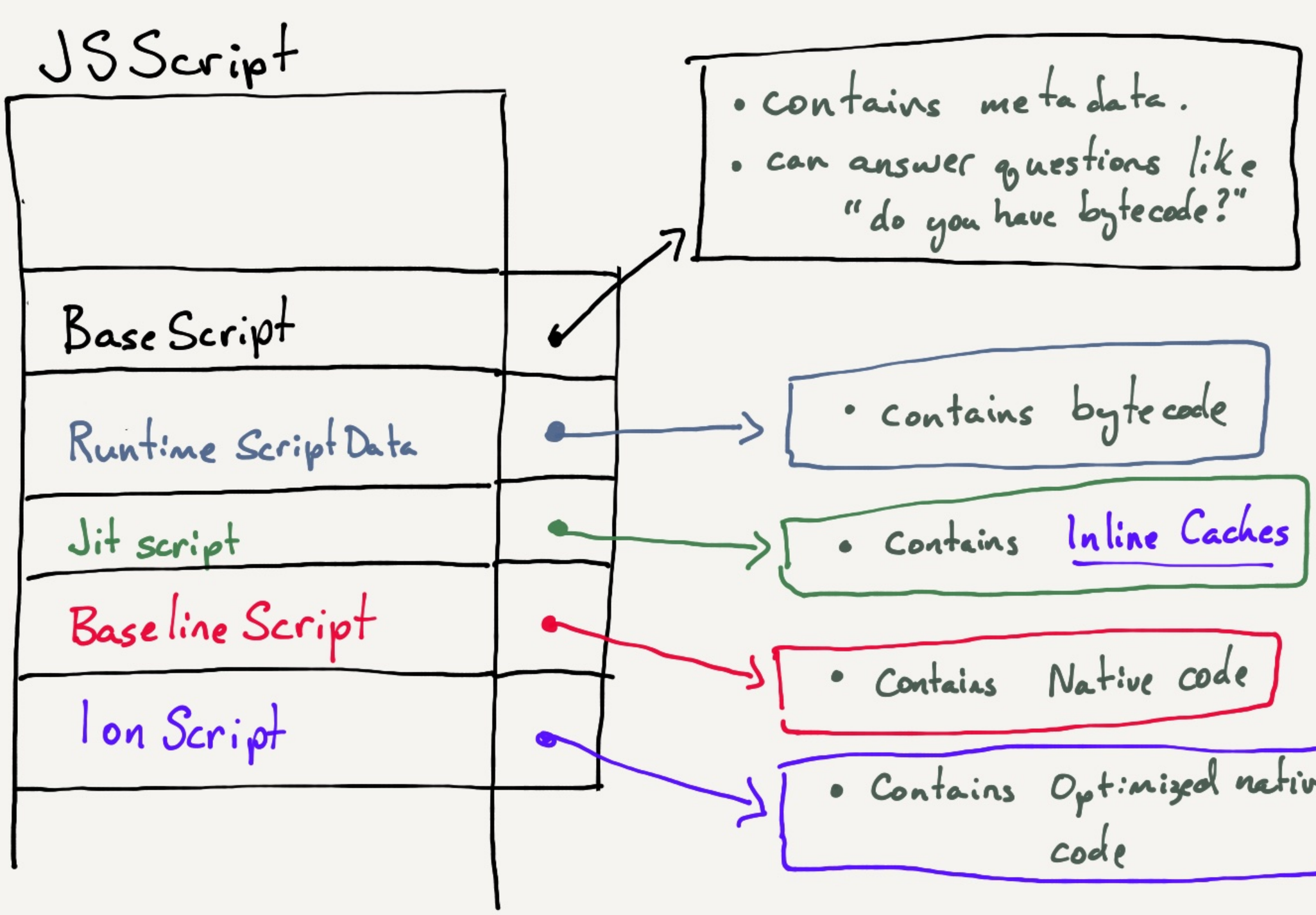

It is easier to go fast if we know what we are working with!

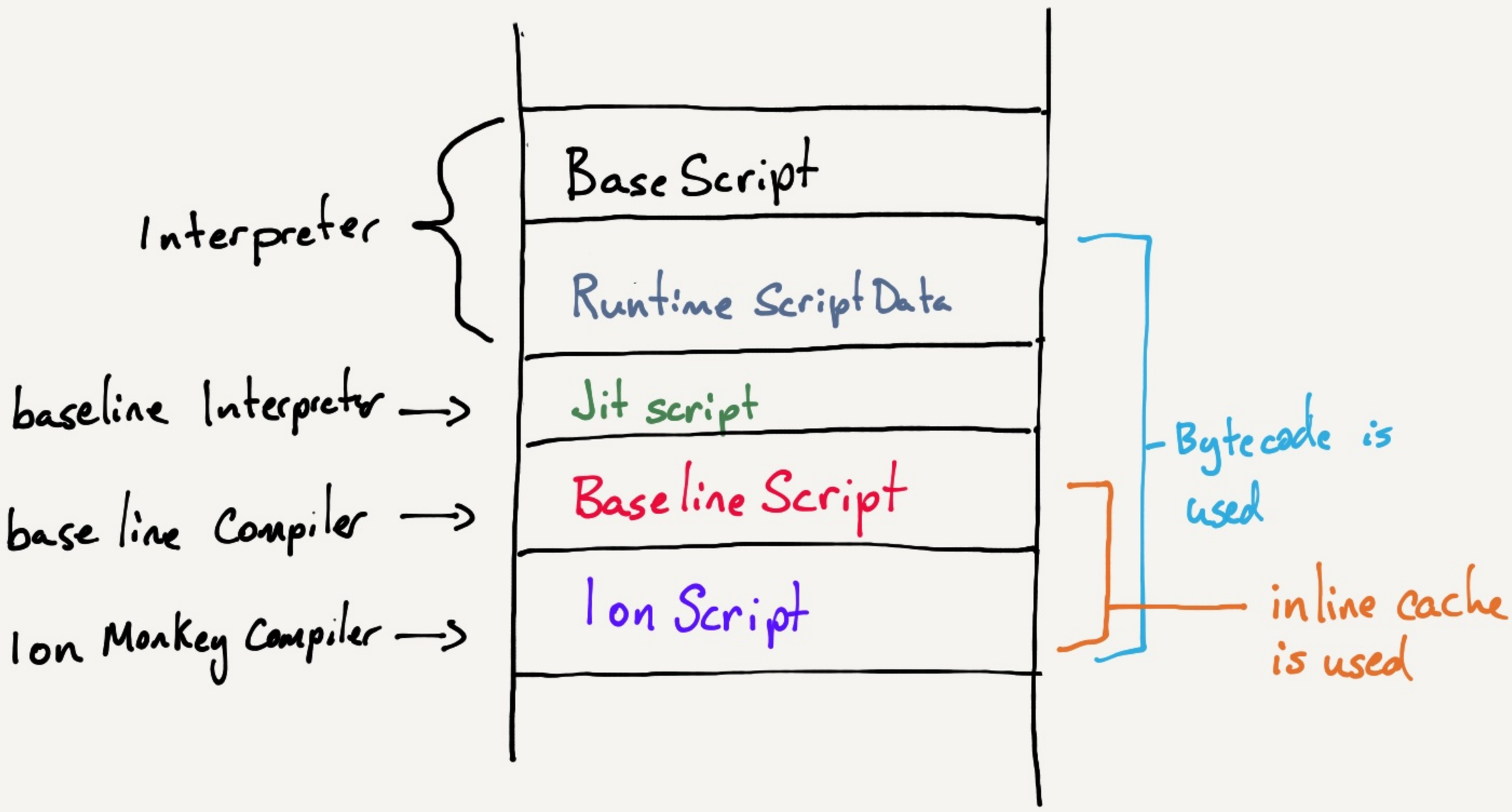


# What is a "Script" anyway?

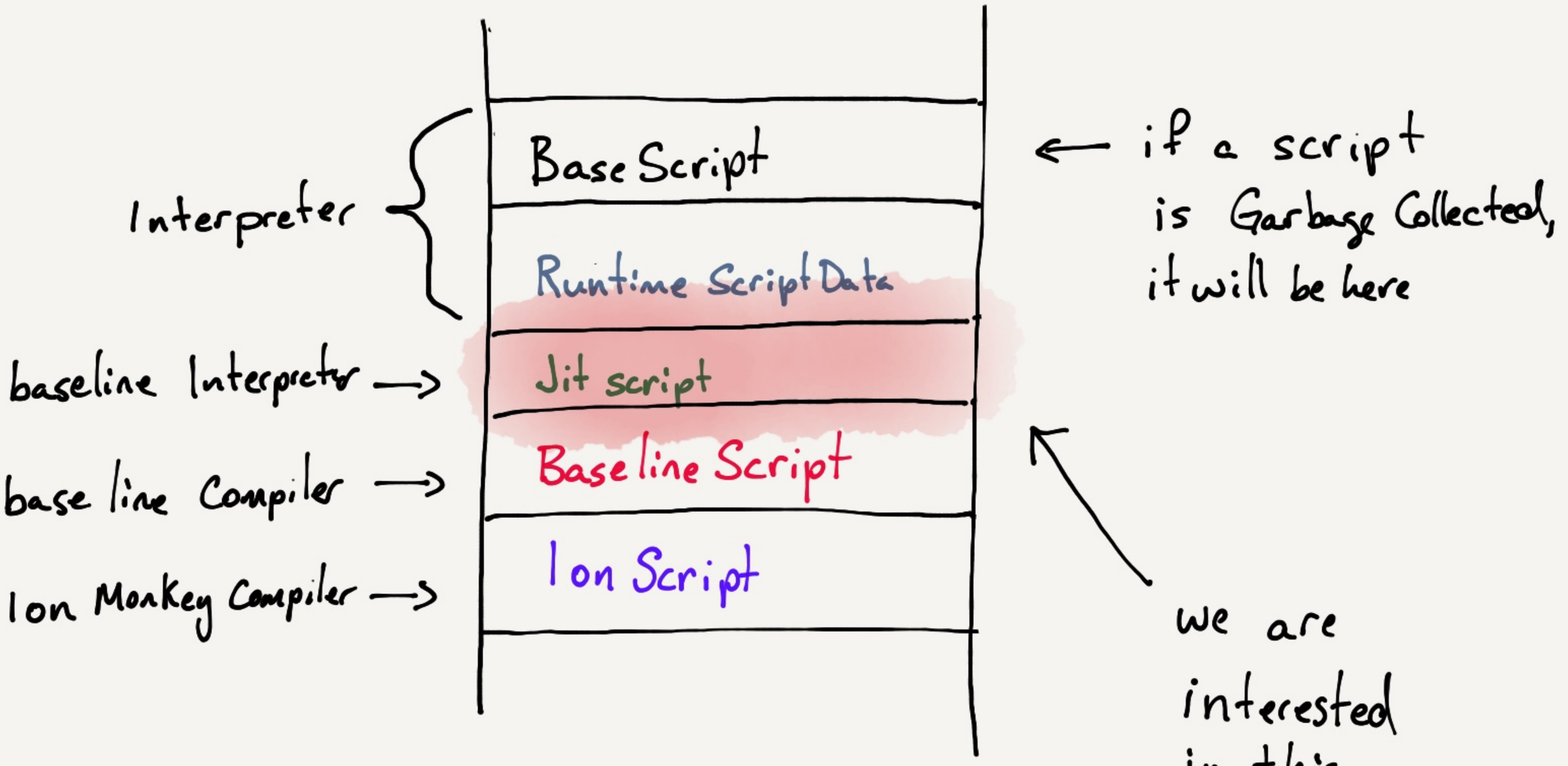



How are Scripts Represented?



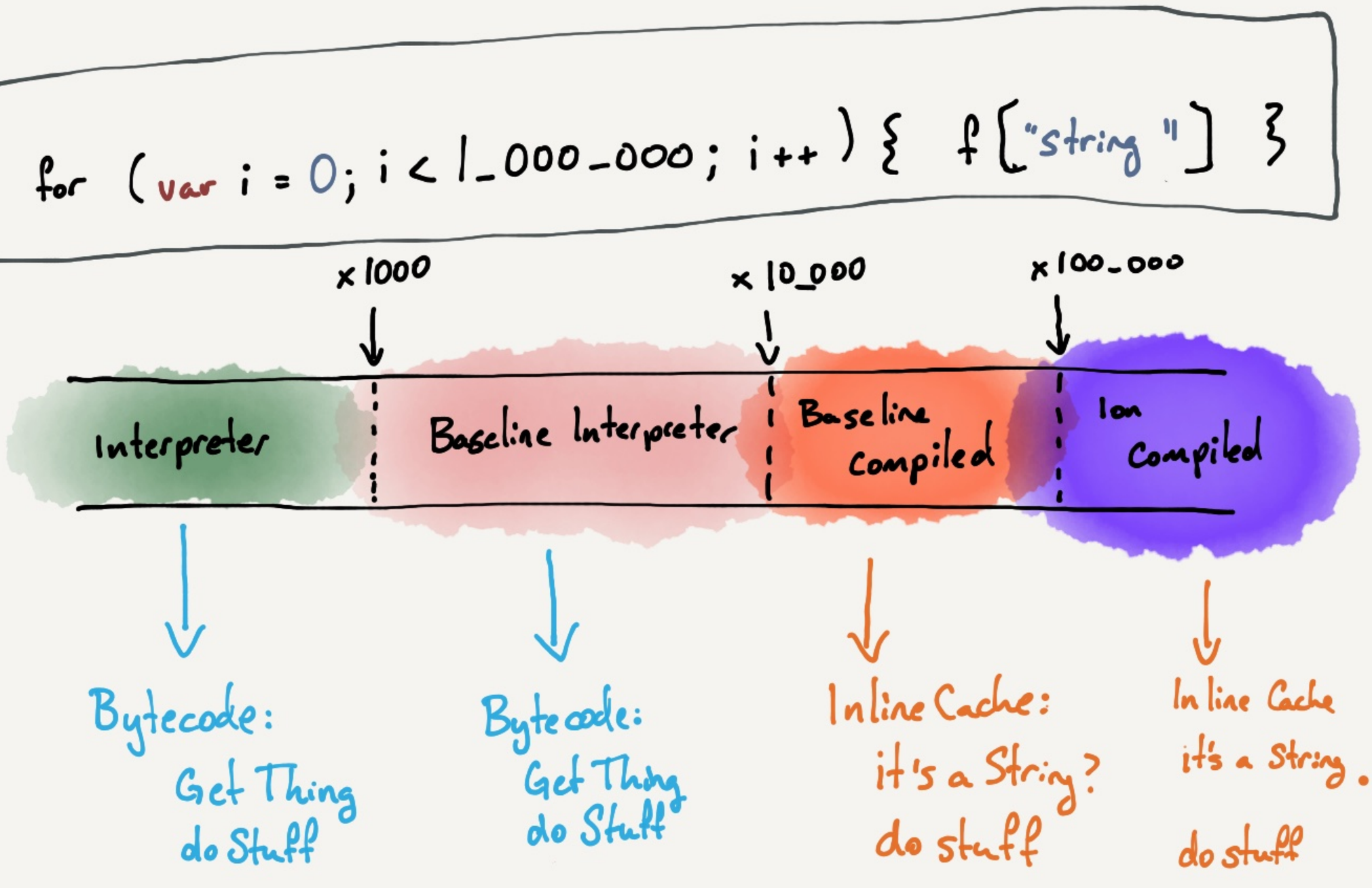



Inline caches are generic stubs of code that can be used to generate native code

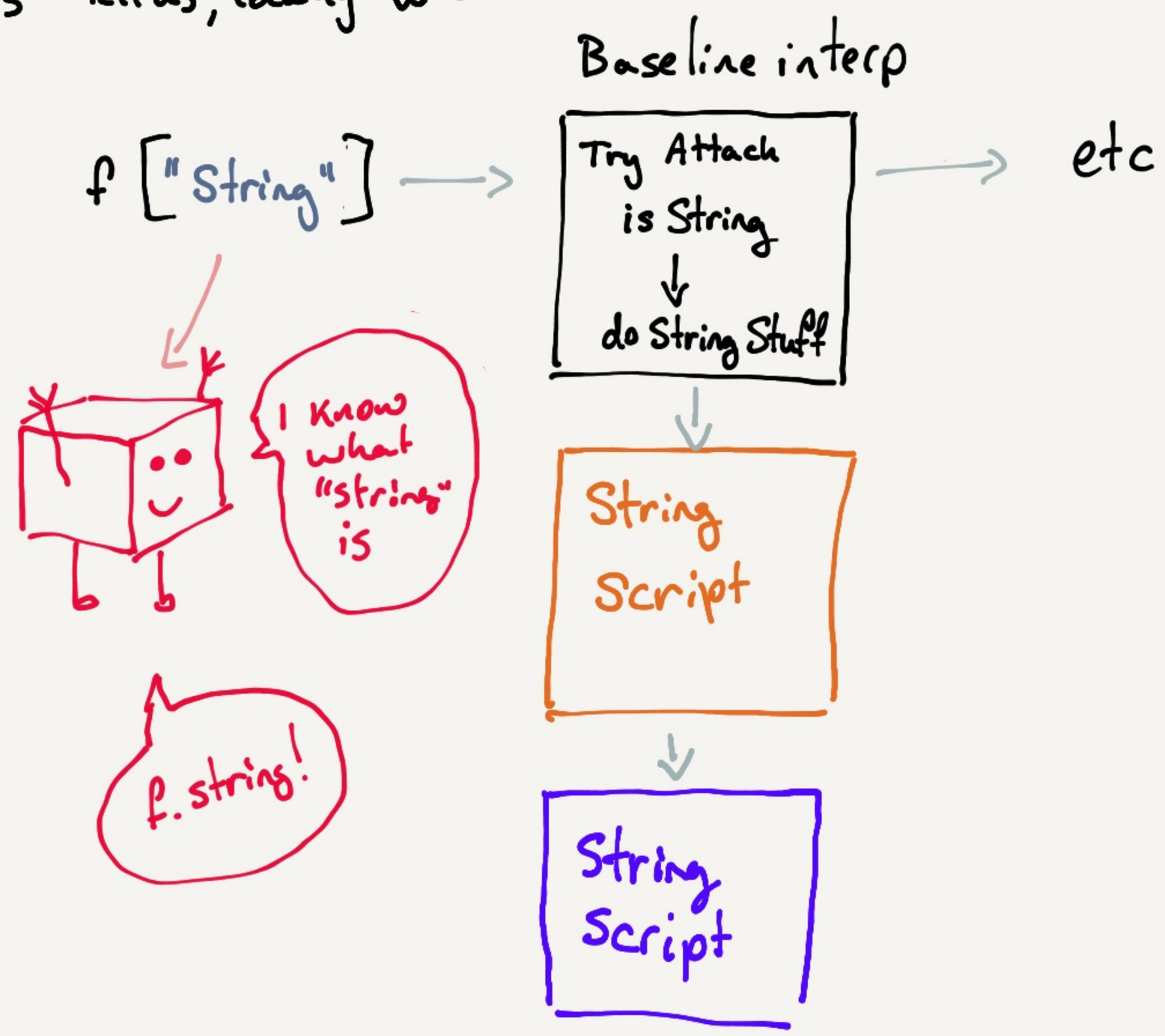



Inline caches are generic stubs of code that can be used to generate native code

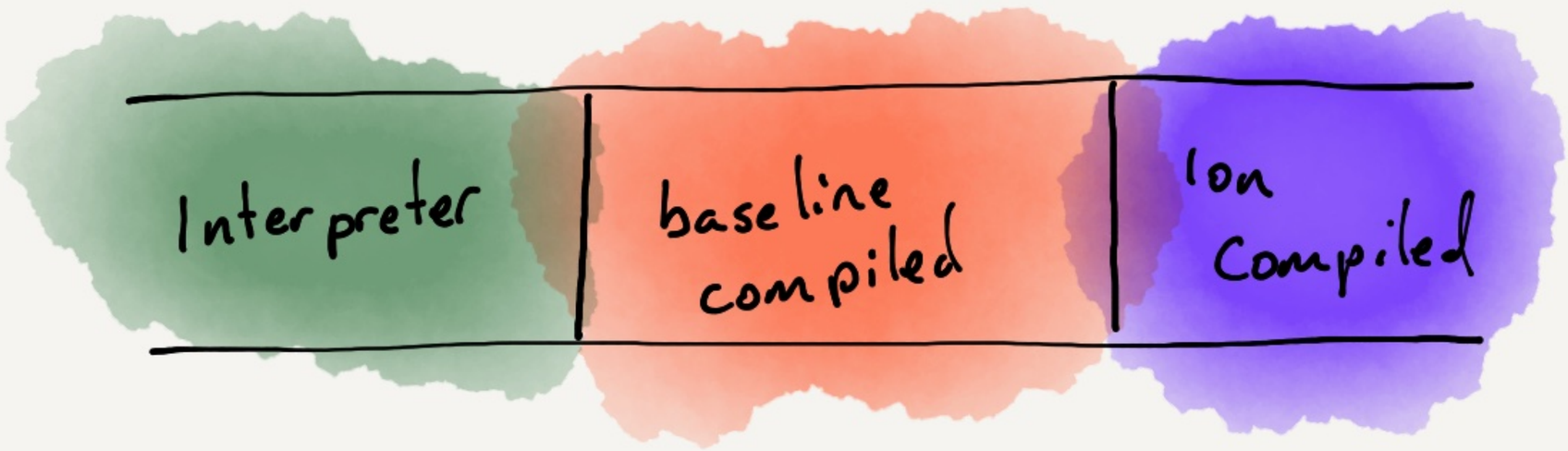
```
for (var i = 0; i < 1_000_000; i++) { f["string"] }
```




Inline Caches tell us, locally what to do

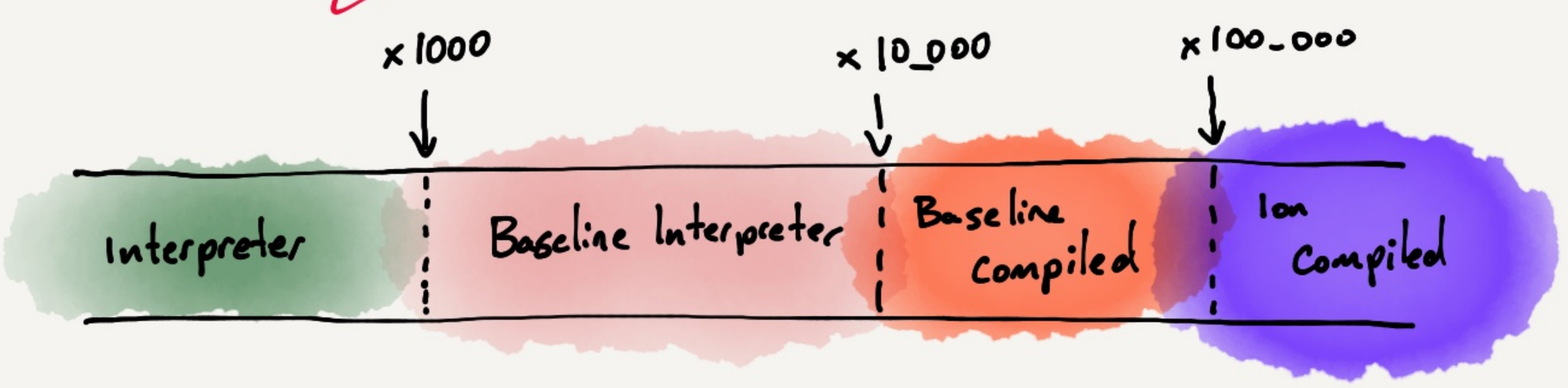


# Why not skip some Steps?



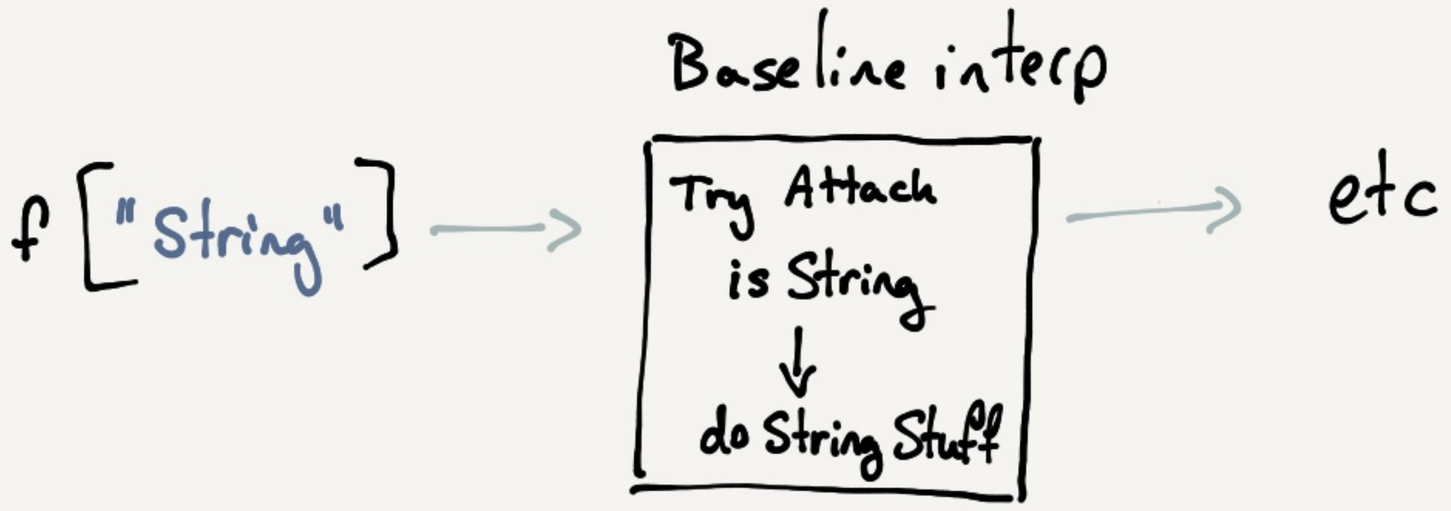

```
for (var i = 0; i < 1_000_000; i++) {
    if (i < 100_001) {
        f("string");
    }
    f(2);
}
```

Polymorphism

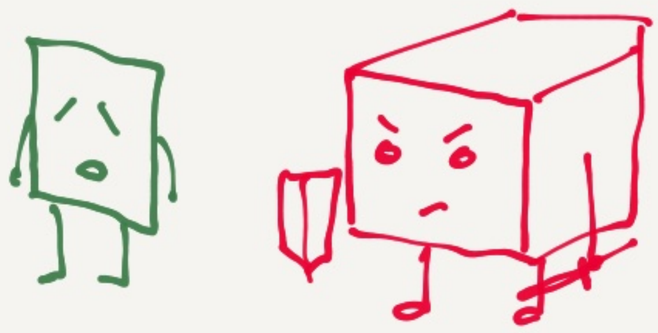


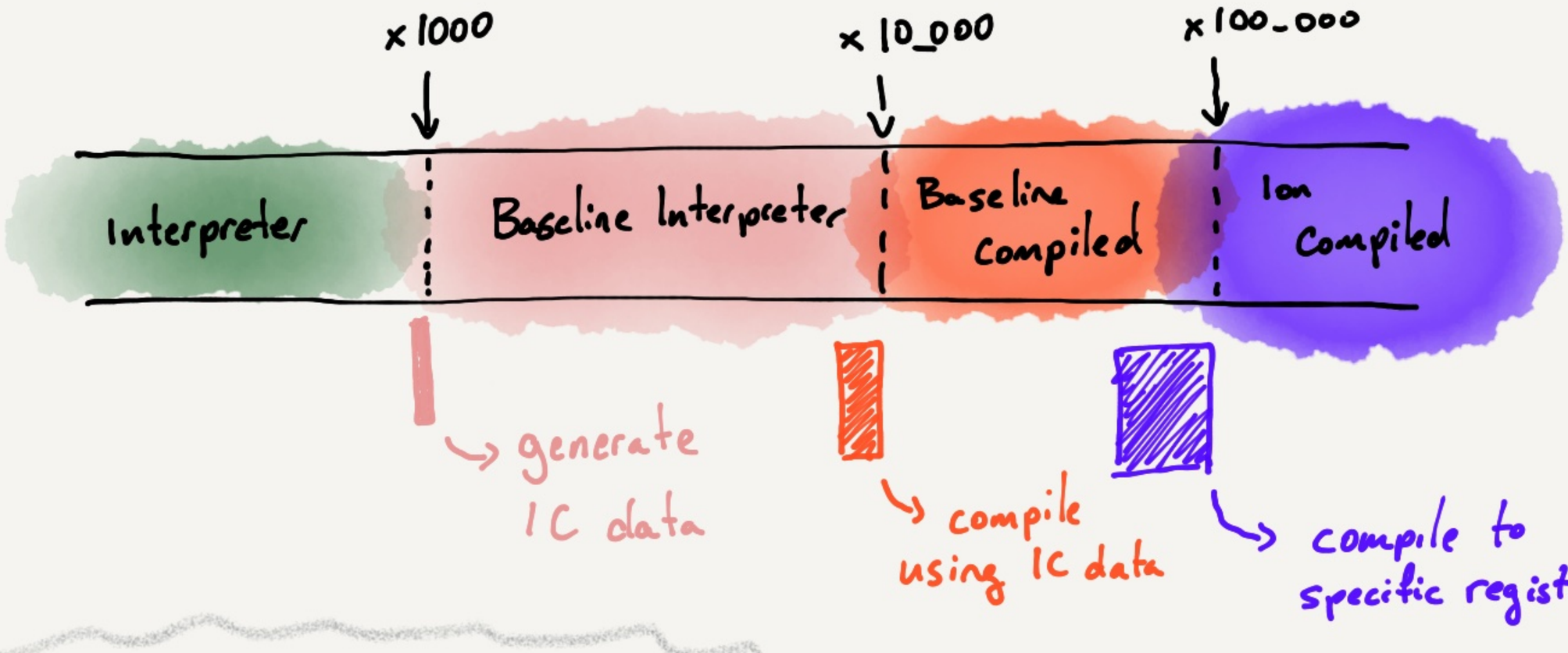
it will be ok. We all have our assumptions challenged sometimes

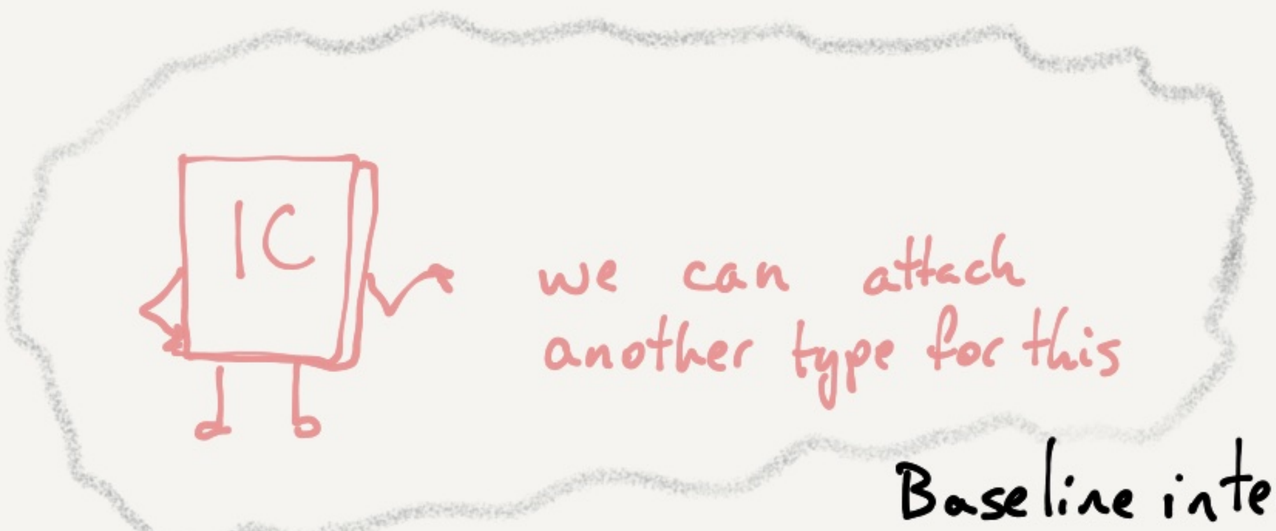
```
for (var i = 0; i < 1_000_000; i++) {
    if (i<100_001) {
        f["string "] i
    }
    f [2];
}
```



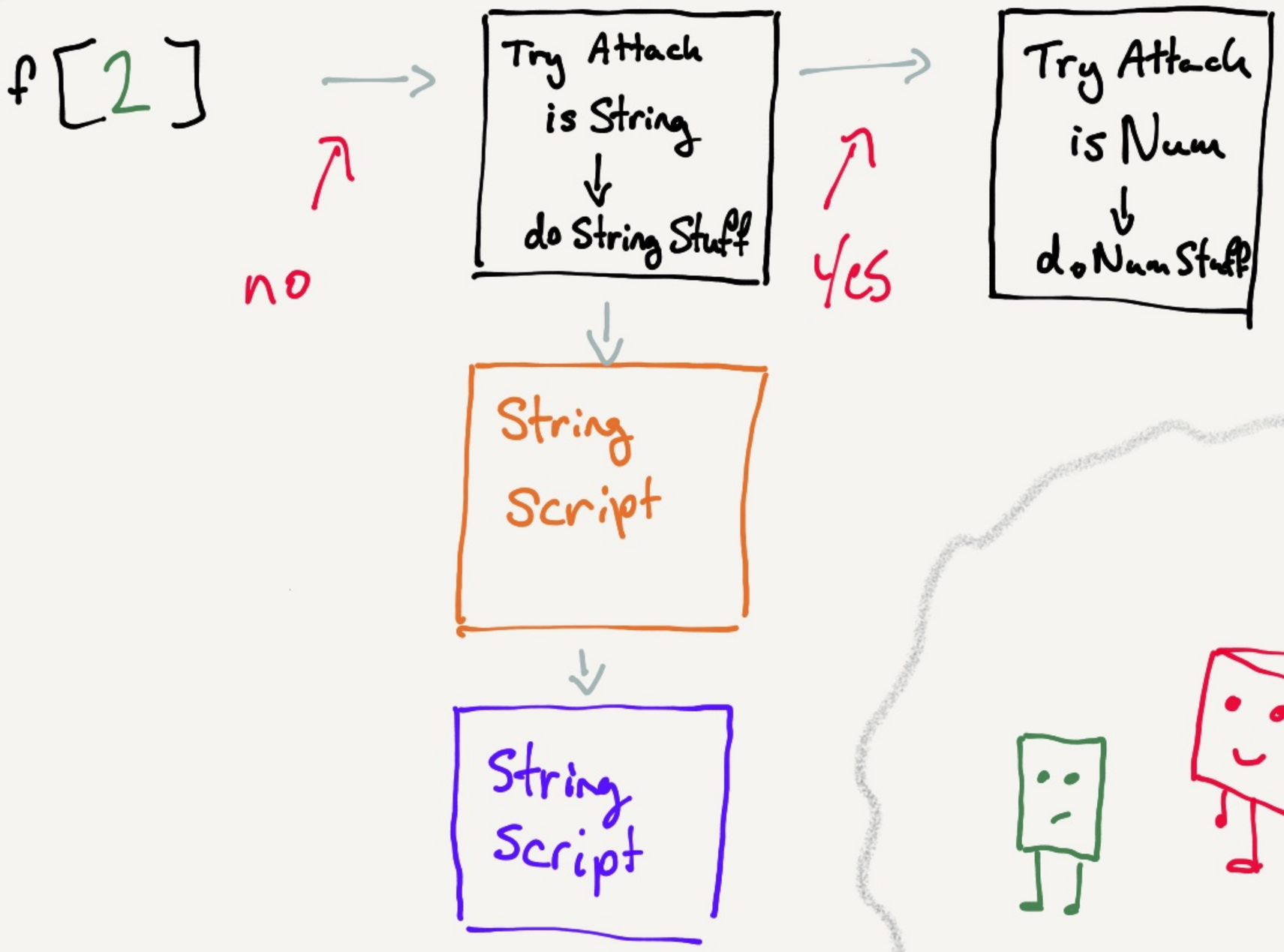

I have an idea...

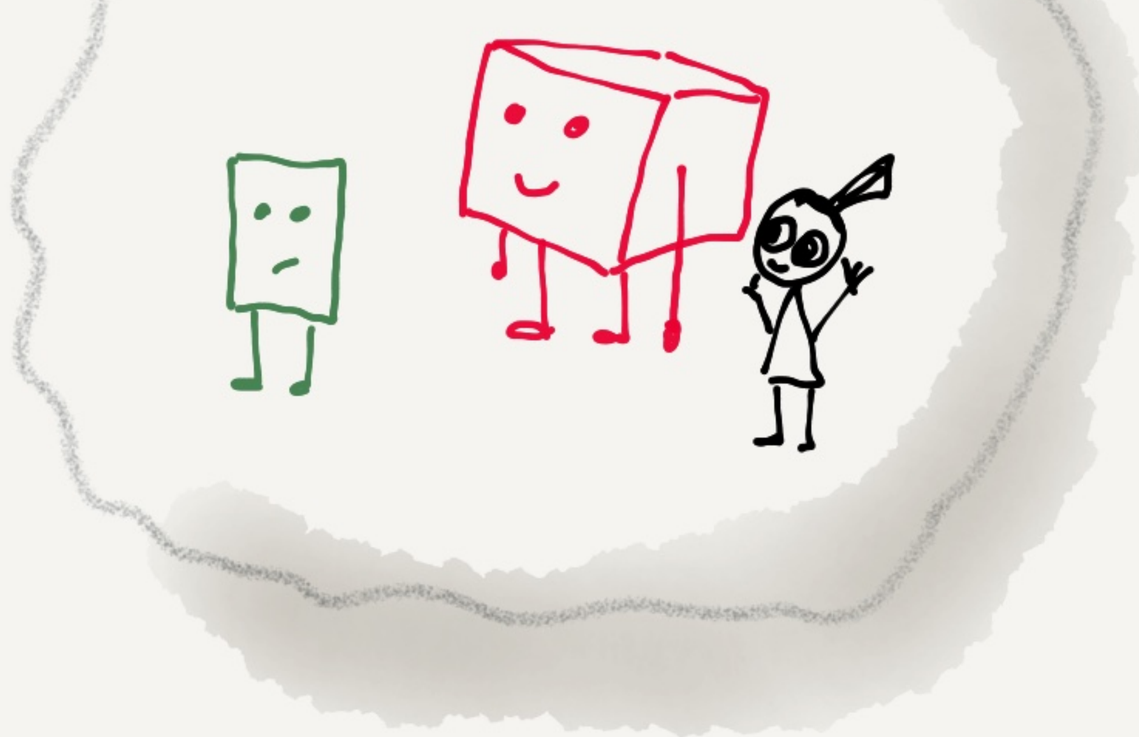




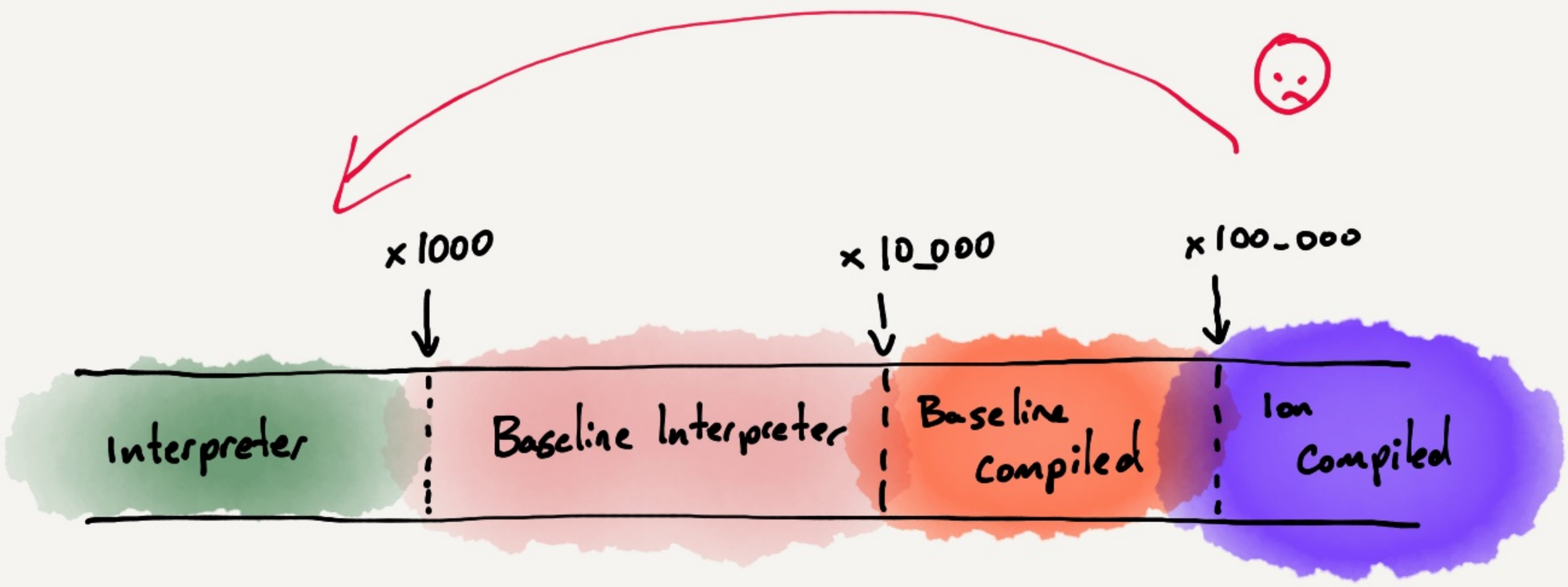




Baseline interp



```
for (var i = 0; i < 1_000_000; i++) {
    if (i<100_001) {
        f("string ");
    }
    f(2);
}
```




With IC