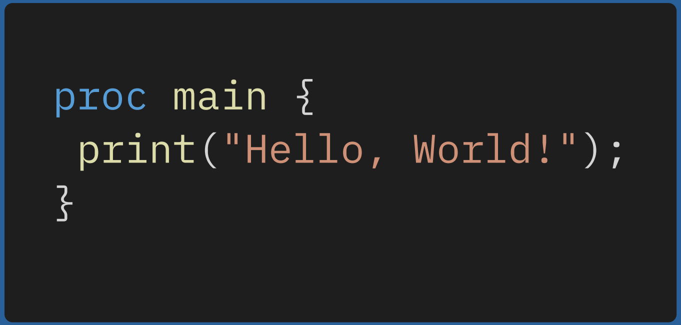
Mank Lang



```
proc main {
 print("Hello, World!");
}
```

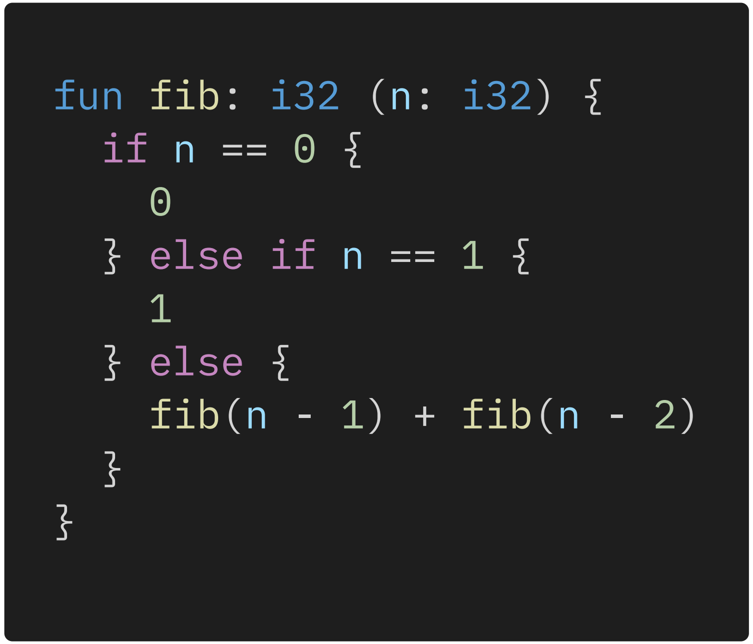
Ben Maxwell

What is Mank?

- Mank intends to be a fairly high level portable language
  - Modern design (avoiding the old baggage of C)
  - High level features: lambdas, closures, dynamic dispatch
  - Provide some extra safety guarantees & avoid undefined behaviour
- Mank compiles to LLVM
  - LLVM is an open source set of compiler tools
    - It does all the complex machine code generation
    - You just have to emit “LLVM IR” from your frontend

Current features

- Fun stuff
  - Lambdas and closures
  - Type inference
  - Destructuring assignments/declarations
  - More expression types (if/block expressions)
- Lots of boring stuff
  - Functions/procedures
  - Standard control flow and expressions
  - Variables and references
  - Structures and arrays



```
fun fib: i32 (n: i32) {
  if n == 0 {
    0
  } else if n == 1 {
    1
  } else {
    fib(n - 1) + fib(n - 2)
  }
}
```

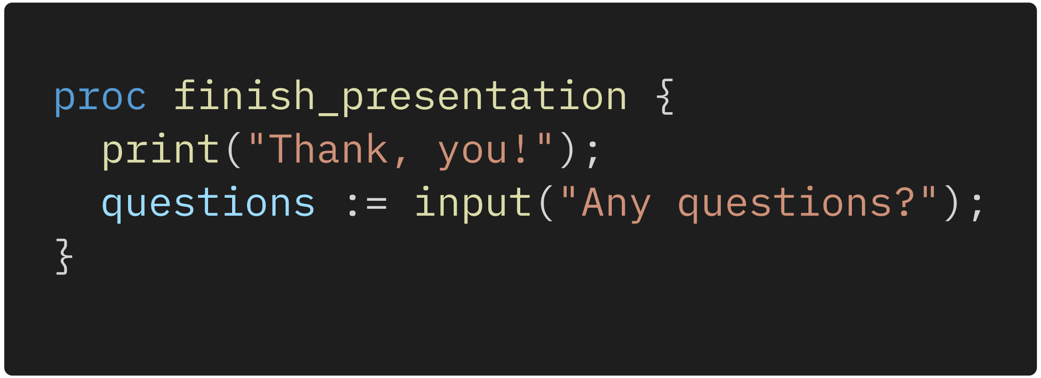
```
proc closure_example {
  make_adder := \x -> { \y -> {x + y}}
  add_10 := make_adder(10);
  fifteen := add_10(5);
}
```

```
proc swap(a: ref i32, b: ref i32) {
  (a, b) = (b, a);
}
```

```
fun fib_iter: i32 (n: i32) {
  a := 0;
  b := 1;
  for _: i32 in 0 .. (n - 1) {
    temp := b;
    b = a + b;
    a = temp;
  }
  b
}
```

Planned features

- Easy to use I/O and C FFI
- Runtime memory management
  - Likely with a garbage collector
- Generic data-types and functions
- Dynamic dispatch / runtime polymorphism
- Pattern matching
- More compile/runtime safety checks
  - Checking for integer under/overflows
  - Avoiding dangling references
  - Inserting bounds checking
  - Etc.



```
proc finish_presentation {
  print("Thank, you!");
  questions := input("Any questions?");
}
```