WebAssembly

Travis Banken and Ben Schroeder

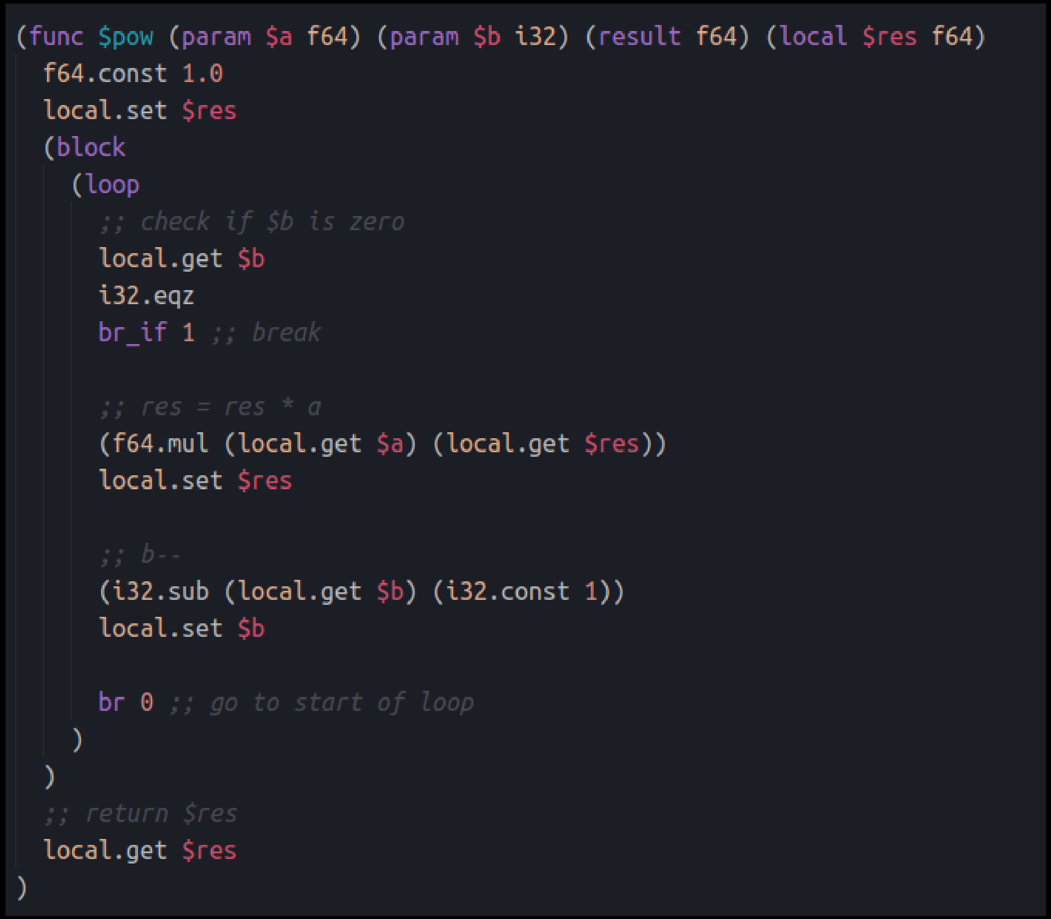
What is WebAssembly

- Its not assembly
- Its not only applicable to the web
- Runs on a stack machine
- Target for compilers of other languages (JavaScript, C, C++, Rust…)
- Pushes portability
- WebAssembly Text Format (WAST) Allows programming directly in WebAssembly

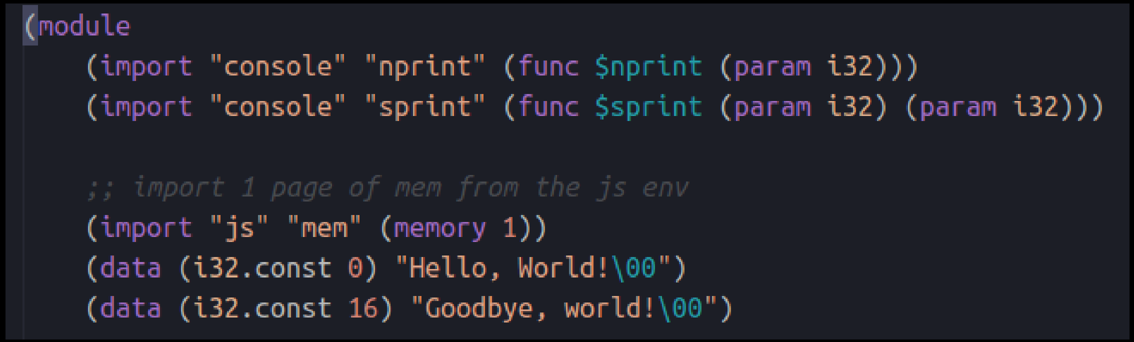
Language Features of Wast

- S-Expressions!
- Stack-Based programming
- Branching and Loops
- 4-Types: i32, i64, f32, f64

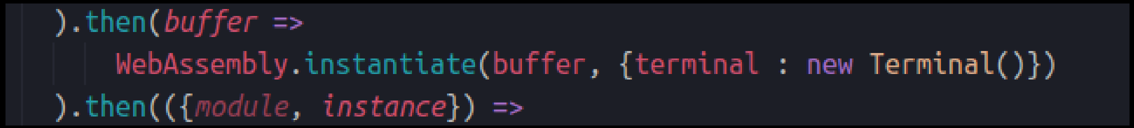
Examples of Wast



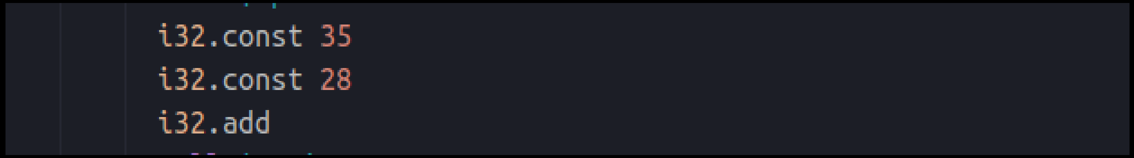
```
(func $pow (param $a f64) (param $b i32) (result f64) (local $res f64)
  f64.const 1.0
  local.set $res
  (block
    (loop
      ;; check if $b is zero
      local.get $b
      i32.eqz
      br_if 1 ;; break

      ;; res = res * a
      (f64.mul (local.get $a) (local.get $res))
      local.set $res

      ;; b--
      (i32.sub (local.get $b) (i32.const 1))
      local.set $b

      br 0 ;; go to start of loop
    )
  )
  ;; return $res
  local.get $res
)
```



```
(module
    (import "console" "nprint" (func $nprint (param i32)))
    (import "console" "sprint" (func $sprint (param i32) (param i32)))

    ;; import 1 page of mem from the js env
    (import "js" "mem" (memory 1))
    (data (i32.const 0) "Hello, World!\00")
    (data (i32.const 16) "Goodbye, world!\00")
```



```
  ).then(buffer =>
    WebAssembly.instantiate(buffer, {terminal : new Terminal()})
  ).then(({module, instance}) =>
```



```
        i32.const 35
        i32.const 28
        i32.add
```

Progam: 2d Graphing