CHIL

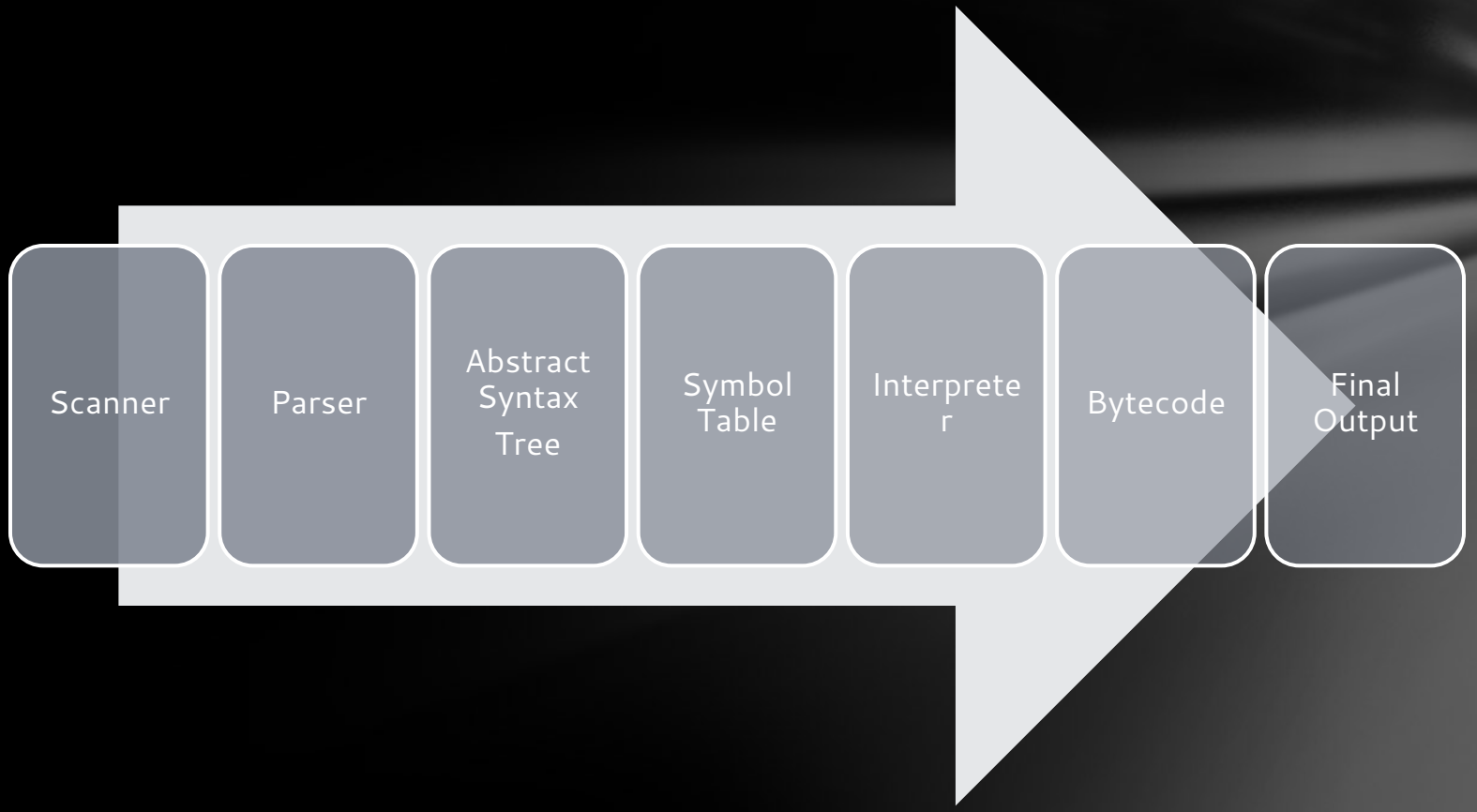
CSS HTML Integrated Language

Gil Chen--Zion - gc2466
Ami Kumar - ak3284
Annania Melaku - amm2324
Isaac White - iaw2105

Overview

- An Abstracted Mark-Up Language
- Bridges the gap between HTML and CSS
- Additional features
  - Looping
  - Simultaneous declarations
    - Structures & Styling
  - Complex elements

Language Structure



Details

- Static Scoping
- Strongly typed
- Global & Local Variable Declaration
- Features are self-scoping
- Sequential Ordered Analysis

Sample Code

functions.ch

```
fn testFunc(string param)
	el someElement = {
		contents: param
	}
	rtn someElement
endfn

el x = testFunc("element 1")
el y = testFunc("element 2")
el z = testFunc("element 3")

Page.add(x)
Page.add(y)
Page.add(z)
```

```
int totalValue = 20 + 22


el theAnswer = {
        contents: totalValue,
        style: ${
                css: "font-weight: bold; font-family:
arial; font-size: 2rem; color: white;
background-color: black; display: block; box-
sizing: border-box; padding: .5rem; border: 1px
dotted white; margin: 1rem;"
        }
}


Page.add(theAnswer)
```

arth1.ch

Lessons Learned

- Scheduling & Time Management
- Laying out the Language
- Ocaml
- Goals & Deadlines
- Testing & Debugging

What We Did

- Syntax Changes
  - \n
  - fn – endfn
  - while – endwhile
  - if – else – endif
  - styling and naming

- Added Binary Operations
  - *=, /=, ++, --
  - squared (^ ^)
  - factorial (!)
  - power (~)

- Refactor Test cases and Makefile

Sample Code

EXAMPLE 2: FIB

```
fn fib(x)
  if (x < 2)
       return 1
  endif
  return fib(x-1) + fib(x-2)
endfn

fn build()
  print(fib(0))
  print(fib(1))
  print(fib(2))
  print(fib(3))
  print(fib(4))
  print(fib(5))
endfn
```

EXAMPLE1

```
fn build()
       int b
       b += 10
       print(b)
       b -= 2
       print(b)
       b /= 4
       print(b)
       b *= 2
       print(b)
       b = b ^^
       print(b)
       b -= 14
       print(b)
       b = b ~ 3
       print(b)
       b = 3!
       print(b)
endfn
```