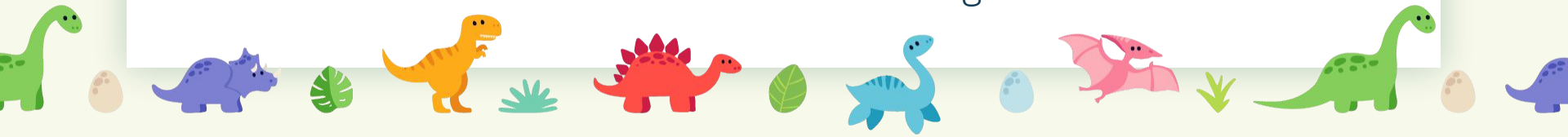
Dino

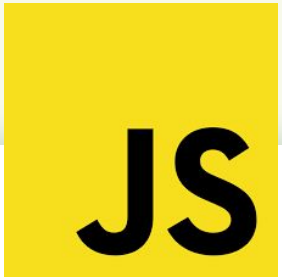
by Anthony M, Owen H, Evan Y, Brandon B

Dino Website: https://anthony29m.github.io

The Story of Dino

In the primordial landscape of programming languages, a language emerged with the thunderous roar of prehistoric beasts. Behold Dino, a programming language that transcends the ordinary, weaving a narrative of code that echoes the majesty of a time long before algorithms ruled the world. Every line of Dino code is an expedition into the untamed wilderness of creativity. This language channels the raw power of dinosaurs into its expressive syntax, turning each program into a thrilling adventure that leaves an indelible mark on the digital terrain.

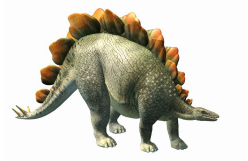




Elevator Pitch

Dino is a dynamically typed language that is a mix of JS and Python.

Simple enough that a dino could code...



Some Features/Focus

- Let and const for simplistic typing
- Less parenthesis, string concatenation
- Lots of for/while/function constraints, short/long return+if

Dino Logic

- Identifiers have to be declared before they are used.
- Arguments/Calls must match up with number of parameters + additional rules
- The primal commands of 'break' and 'continue' are restricted to loops
- Cannot have non-boolean value in a conditional or while loop.
- Has brackets where it makes sense, as we feel that is still necessary for clarity
- Silly emoji comments

Examples

```
quest concatenateStrings(str1, str2) {
  hatch str1 + " " + str2
}

🦕 Usage
letdino result = concatenateStrings("Hello", "Dino!")
rawr result
```

```
🦕 Dino
quest hungry(hunger) {
  if-rex hunger > 100 {
    hatch "full"
  } t-else {
    hatch "hungry"
  }
}

🦕 Usage
letdino str = hungry(50)
rawr str
```

```
quest sumOfSquares(n) {
  letdino sum = 0
  roar 1 to n {
    sum = i * i + sum
  }
  hatch sum
}

🦕 Usage
letdino SoS = sumOfSquares(3)
```

Thanks for listening