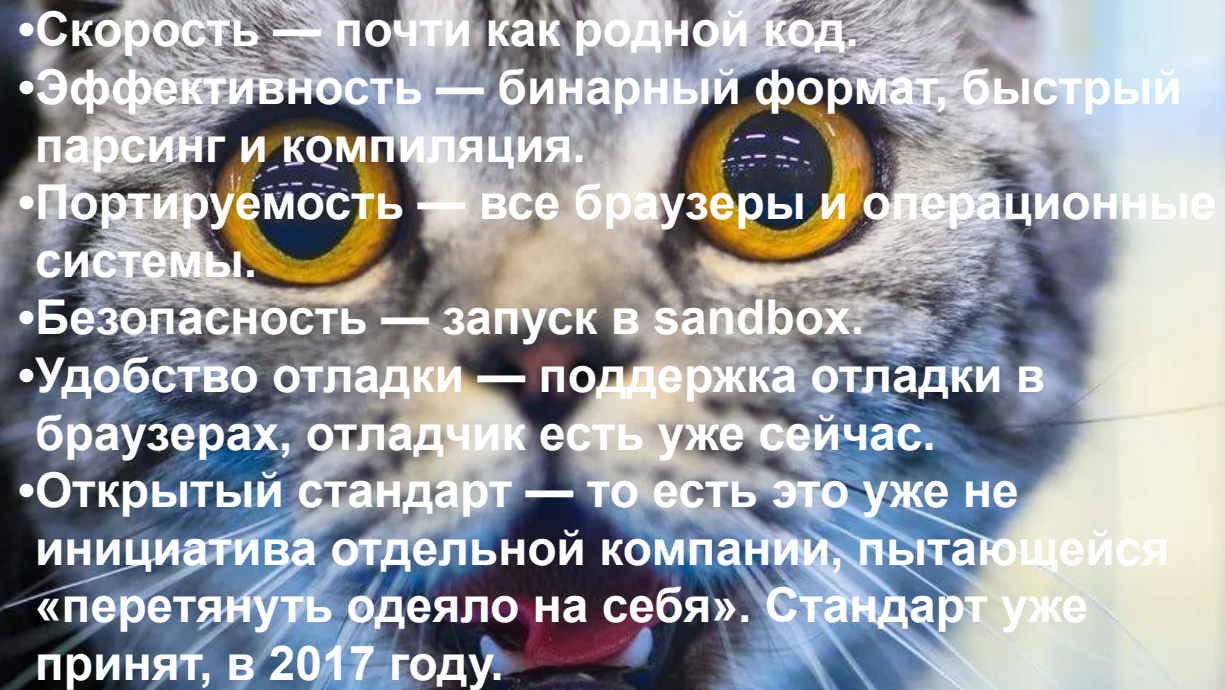
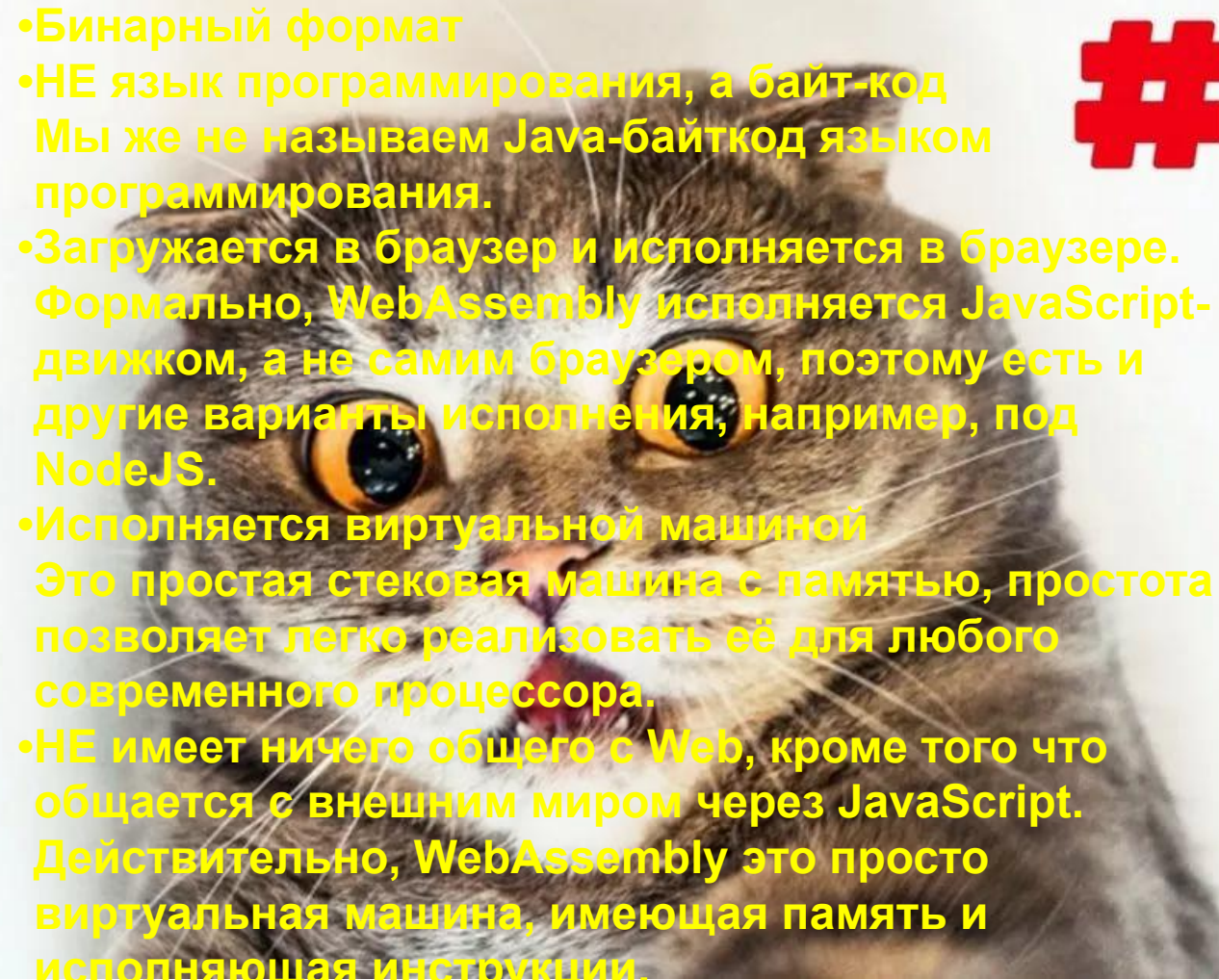


**Вебасембли:
если бы я знал, что
дальше...**



WEBASSEMBLY

- 
- Скорость — почти как родной код.
 - Эффективность — бинарный формат, быстрый парсинг и компиляция.
 - Портируемость — все браузеры и операционные системы.
 - Безопасность — запуск в sandbox.
 - Удобство отладки — поддержка отладки в браузерах, отладчик есть уже сейчас.
 - Открытый стандарт — то есть это уже не инициатива отдельной компании, пытающейся «перетянуть одеяло на себя». Стандарт уже принят, в 2017 году.

- 
- Бинарный формат
 - НЕ язык программирования, а байт-код
Мы же не называем Java-байткод языком программирования.
 - Загружается в браузер и исполняется в браузере.
Формально, WebAssembly исполняется JavaScript-движком, а не самим браузером, поэтому есть и другие варианты исполнения, например, под NodeJS.
 - Исполняется виртуальной машиной
Это простая стековая машина с памятью, простота позволяет легко реализовать её для любого современного процессора.
 - НЕ имеет ничего общего с Web, кроме того что общается с внешним миром через JavaScript.
Действительно, WebAssembly это просто виртуальная машина, имеющая память и исполняющая инструкции.

10

Суть ТОКОВА:



Исходный код на C	«линейный ассемблерный байт-код»	бинарный код WASM
<pre data-bbox="461 486 983 758">int factorial(int n) { if (n == 0) return 1; else return n * factorial(n-1); }</pre>	<pre data-bbox="1009 486 1536 972">get_local 0 i64.eqz if i64 i64.const 1 else get_local 0 get_local 0 i64.const 1 i64.sub call 0 i64.mul end</pre>	<pre data-bbox="1561 486 2086 972">20 00 50 04 7e 42 01 05 20 00 20 00 42 01 7d 10 00 7e 0b</pre>



```
(module
  (type $FUNCSIG$dd (func (param f64) (result f64)))
  (import "global.Math" "exp" (func $exp (param f64) (result f64)))
  (memory 256 256)
  (export "memory" (memory 0))
  (func $doubleExp (param $0 f64) (result f64)
    (f64.mul
      (call $exp
        (get_local $0)
      )
      (f64.const 2)
    )
  )
  (export "doubleExp" (func $doubleExp))
)
```




Итог

И



Спасибо за
внимание!