

INTRODUCING ERLANG

INSTALLING ERLANG

<https://www.erlang-solutions.com/downloads/download-erlang-otp>

What's so great about it?

- Great concurrency model - easy to understand, easy to reason about
- Good at slicing and dicing network protocols
- "For free" SMP scaling (up to about 8 cores)
- Fault tolerance mechanisms help avoid "defensive coding"
- Replace/upgrade running code without downtime

What sucks about it?

- String handling is stupid - represented as a list of integers in Latin-1 range 0 - 255 - hard to distinguish between legit list of integers and a string
- Library support isn't that great.
- *A bit* of a learning curve.

Erlang types

- Integers (example: 1)
- Floats (example: 3.14)
- Binary (example: <<205, 72, 29, 1, 0, 92>>)
 - Binary strings <<"Hello!">>
- Tuples (example: {key, "Value"})
- Lists (example: [1, "Hi!", 42.1])
- Atoms (example: 'error')

- A 'process' is the unit of statefulness - there are no objects in Erlang.
- Variables are immutable (once bound, they can never be rebound to a new value in the same lexical scope)
- The looping construct is recursion (no while, for, until, etc)
- Logic branches are accomplished by pattern matching not if-then-else statements. (Erlang has an 'if' keyword but should be avoided if possible.)

IMPLEMENT FIZZBUZZ

Generate a list of integers from 1 to 100. For each number:

- If it's divisible by 3 output "fizz"
- If it's divisible by 5 output "buzz"
- If it's divisible by 3 and 5 output "fizzbuzz"

Here it is in 46 characters:

```
print+(Fizz)[$_%3].(Buzz)[$_%5]||$_ for 1..100
```

```
12Fizz4BuzzFizz78FizzBuzz11Fizz1314FizzBuzz1617Fizz19Buzz  
Fizz2223FizzBuzz26Fizz2829FizzBuzz3132Fizz34BuzzFizz3738  
FizzBuzz41Fizz4344FizzBuzz4647Fizz49BuzzFizz5253FizzBuzz  
56Fizz5859FizzBuzz6162Fizz64BuzzFizz6768FizzBuzz71Fizz73  
74FizzBuzz7677Fizz79BuzzFizz8283FizzBuzz86Fizz8889Fizz  
Buzz9192Fizz94BuzzFizz9798FizzBuzz
```