

Rollup comparisons

Discuss in your teams the rollup L2s that you have experience with.

Polygon zkEVM

Optimism -> Kwenta (has lower fees ;))

Arbitrum (L2) → GMX (Gambling not trading xD)

zkSync

<https://l2beat.com/>

What factors are important to you when choosing a rollup?, in particular discuss the criteria we covered in the lesson.

Cheap Fees

Infrastructure (Swap/Bridges/ Interoperability)

Centralized/Decentralized? Manipulation Resistant

Tx Speed (execution speed + finality)

Rust practice

Fizz buzz program

Create a project called bootcamp using Cargo.

The main function should print a welcome message.

Write a 'fizz buzz' function that will be called from your main function.

The function should have a loop counting up to 301.

If the count is divisible by 3, print "fizz".

If the count is divisible by 5, print "buzz".

If the count is divisible by 3 and 5, print "fizz buzz".

At the end, print the number of times "fizz buzz" occurred.

```
fn main() {  
    println!("Welcome to the Fizz Buzz program!");  
  
    let mut fizz_buzz_count = 0;  
  
    // The function is now defined outside of 'main'  
    for i in 0..=301 {  
        fizz_buzz_count += fizz_buzz(i);  
    }  
}
```

```
println!("'Fizz Buzz' occurred {} times.", fizz_buzz_count);
}

// 'fizz_buzz' function is defined here, outside of 'main'
// It returns 1 if the current number is a 'fizz buzz', otherwise 0
fn fizz_buzz(i: u32) -> u32 {
    let fizz = i % 3 == 0;
    let buzz = i % 5 == 0;

    match (fizz, buzz) {
        (true, true) => {
            println!("fizz buzz");
            1 // Return 1 for 'fizz buzz'
        },
        (true, false) => {
            println!("fizz");
            0 // Return 0 for other cases
        },
        (false, true) => {
            println!("buzz");
            0 // Return 0 for other cases
        },
        _ => 0, // Return 0 for other cases
    }
}
```