



NIKE CROSS NATIONALS

New Generation Track and Field
1.3K views • 42 minutes ago



Kanye West x Gav Interview | Hasanabi reacts

Hasan Compilations
31K views • 3 hours ago



Portugal vs. Switzerland Highlights - FIFA World Cup 2022

TSN
338K views • 5 hours ago



GUARDIANS OF THE GALAXY Volume 3 Trailer BREAKDOWN: Easter Eggs and...

ScreenCrush
543K views • 5 days ago



Why You Shouldn't Nest Your Code

CodeAesthetic
113K views • 10 hours ago



Evil in Andor | The Banality of Evil

The Canvas
133K views • 9 days ago



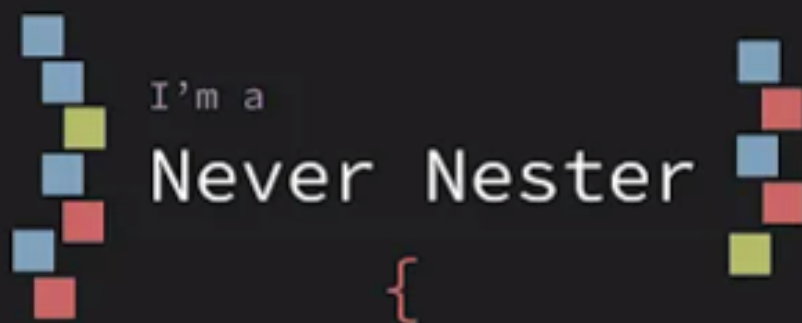
I Thought It Couldn't Get Any CRAZIER... (Massive Record)

Total Running Productions
12K views • 4 hours ago



The Truth About Jenna Ortega & this Wednesday Backlash, AI Made This...

Philip DeFranco
302K views • 3 hours ago



8:30



Why You Shouldn't Nest Your Code

CodeAesthetic

113K views • 10 hours ago



```
int calculate(int bottom, int top)
{
    if (top > bottom)
    {
        int sum = 0;

        for (int number = bottom; number <= top; number++)
        {
            if (number % 2 == 0)
            {
                sum += number;
            }
        }

        return sum;
    }
    else
    {
        return 0;
    }
}
```



```
int calculate(int bottom, int top)
{
    if (top > bottom)
    {
        int sum = 0;

        for (int number = bottom; number <= top; number++)
        {
            if (number % 2 == 0)
            {
                sum += number;
            }
        }

        return sum;
    }
    else
    {
        return 0;
    }
}
```



```
int calculate(int bottom, int top) {  
    if (top > bottom) {  
        int sum = 0;  
        for (int number = bottom; number <= top; number++) {  
            if (number % 2 == 0) {  
                sum += number;  
            }  
        }  
        return sum;  
    } else {  
        return 0;  
    }  
}
```



```
int calculate(int bottom, int top) {  
    if (top > bottom) {  
        int sum = 0;  
        for (int number : std::views::iota(bottom, top + 1)) {  
            if (number % 2 == 0) {  
                sum += number;  
            }  
        }  
        return sum;  
    } else {  
        return 0;  
    }  
}
```




```
int calculate(int bottom, int top) {  
    if (top > bottom) {  
        int sum = 0;  
        auto even = [](auto e) { return e % 2 == 0; };  
        for (int number : std::views::iota(bottom, top + 1)  
              | std::views::filter(even)) {  
            sum += number;  
        }  
        return sum;  
    } else {  
        return 0;  
    }  
}
```



```
int calculate(int bottom, int top) {  
    if (top > bottom) {  
        auto even = [](auto e) { return e % 2 == 0; };  
        auto evens = std::views::iota(bottom, top + 1)  
                    | std::views::filter(even);  
        return std::accumulate(evens.begin(), evens.end(), 0);  
    } else {  
        return 0;  
    }  
}
```



```
int calculate(int bottom, int top) {  
    if (top > bottom) {  
        auto even = [](auto e) { return e % 2 == 0; };  
        return ranges::accumulate(  
            std::views::iota(bottom, top + 1) |  
            std::views::filter(even), 0);  
    } else {  
        return 0;  
    }  
}
```



```
namespace rv = std::views;

int calculate(int bottom, int top) {
    if (top > bottom) {
        auto even = [](auto e) { return e % 2 == 0; };
        return ranges::accumulate(
            std::views::iota(bottom, top + 1) |
            std::views::filter(even), 0);
    } else {
        return 0;
    }
}
```



```
namespace rv = std::views;

int calculate(int bottom, int top) {
    if (top > bottom) {
        auto even = [](auto e) { return e % 2 == 0; };
        return ranges::accumulate(
            rv::iota(bottom, top + 1) |
            rv::filter(even), 0);
    } else {
        return 0;
    }
}
```



```
namespace rv = std::views;

int calculate(int bottom, int top) {
    return top <= bottom ? 0 :
        ranges::accumulate(
            rv::iota(bottom, top + 1) |
            rv::filter([](auto e) { return e % 2 == 0; }), 0);
}
```



```
fn calculate(bottom: i32, top: i32) -> i32 {  
    (bottom..=top).filter(|e| e % 2 == 0).sum()  
}
```



```
fn calculate(bottom: i32, top: i32) -> i32 {  
    (bottom..=top)  
        .filter(|e| e % 2 == 0)  
        .sum()  
}
```




```
int calculate(int bottom, int top)
{
    if (top > bottom)
    {
        int sum = 0;

        for (int number = bottom; number <= top; number++)
        {
            if (number % 2 == 0)
            {
                sum += number;
            }
        }

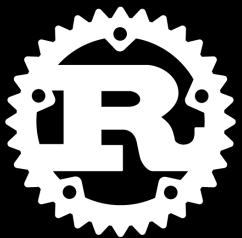
        return sum;
    }
    else
    {
        return 0;
    }
}
```



```
fn calculate(bottom: i32, top: i32) -> i32 {  
    (bottom..=top)  
        .filter(|e| e % 2 == 0)  
        .sum()  
}
```



```
calculate :: Int -> Int -> Int  
calculate bottom top = sum $ filter even [bottom..top]
```



```
fn calculate(bottom: i32, top: i32) -> i32 {  
    (bottom..=top).filter(|e| e % 2 == 0).sum()  
}
```



```
calculate :: Int -> Int -> Int  
calculate bottom top = sum $ filter even [bottom..top]
```

Assembly Instructions Generated (-O2 v -O3 with GCC 12.2)

