



# codex

89

90

91

92



# **Guy Royse**

Engineering Manager  
ScriptDrop

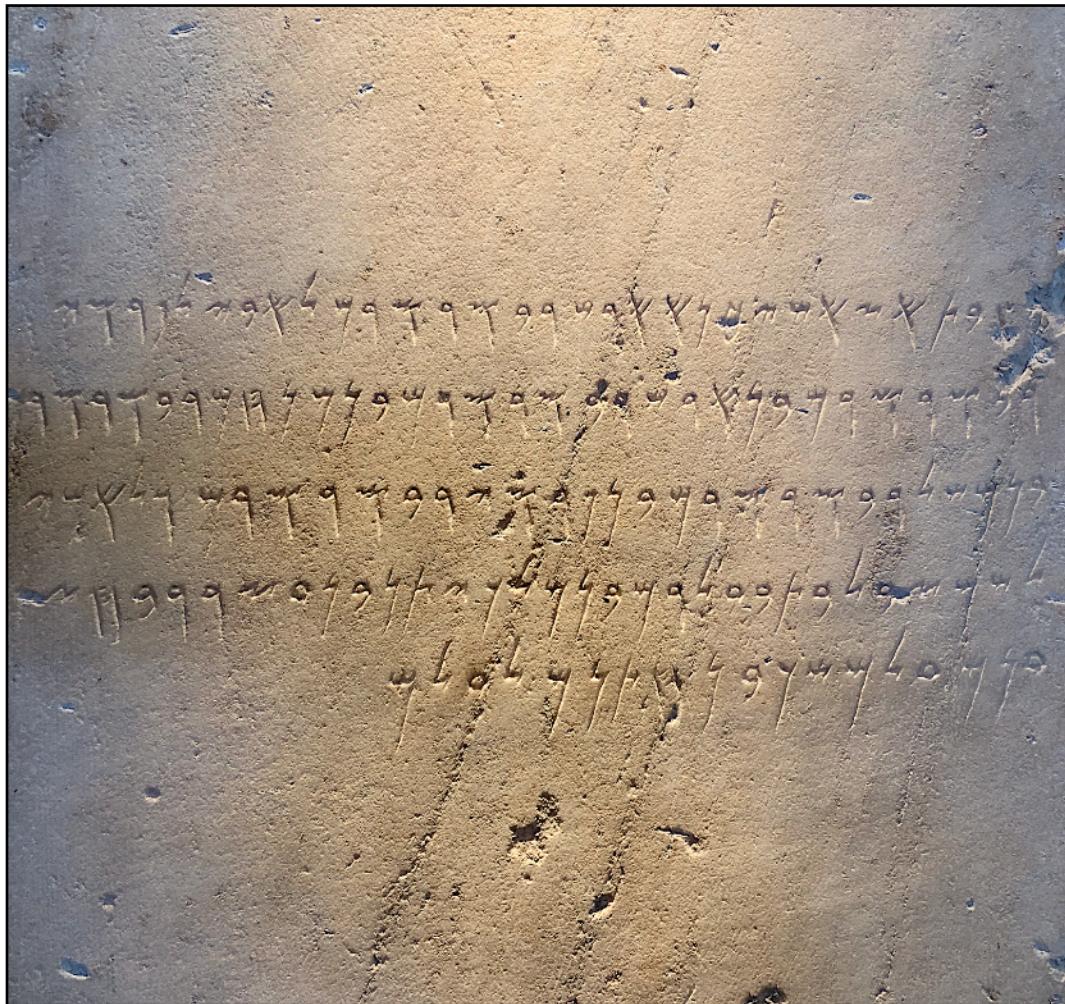
 guyroyse

 code.guy.dev

 guy.dev



# Linguistics & Programming



```
// Promise from setTimeout
const afterSomeTime = (time) => new Promise(resolve => {
  setTimeout(() => {
    resolve(true);
  }, time);
});
const callAfterSomeTime = (callback, time) => afterSomeTime(time).then(callback);

callAfterSomeTime() => console.log('Hello after 1500ms');

const getData = async (url) => fetch(url);

document
  .querySelector('#submit')
  .addEventListener('click', function() {
    const name = document.querySelector('#name').value;

    // send to backend
    const user = await fetch(`users?name=${name}`);
    const posts = await fetch(`posts?userId=${user.id}`);
    const comments = await fetch(`comments?post=${posts[0].id}`);
    //display comments on DOM
  });
}

Ln 24, Col 1 Spaces 2
```



**What if computers hadn't been  
developed by English speakers?**

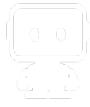


```
class FizzBuzz {  
    transform(number) {  
        if (number % 15 === 0) return 'fizzbuzz'  
        if (number % 5 === 0) return 'buzz'  
        if (number % 3 === 0) return 'fizz'  
        return number  
    }  
}  
  
let number = 12  
let fizzbuzz = new FizzBuzz()  
let result = fizzbuzz.transform(number)
```



```
class FizzBuzz {  
    transform(number) {  
        if (number % 15 === 0) return 'fizzbuzz'  
        if (number % 5 === 0) return 'buzz'  
        if (number % 3 === 0) return 'fizz'  
        return number  
    }  
}
```

```
let number = 12  
let fizzbuzz = new FizzBuzz()  
let result = fizzbuzz.transform(number)
```



```
class FizzBuzz {  
    transform(number) {  
        if (number % 15 === 0) return 'fizzbuzz'  
        if (number % 5 === 0) return 'buzz'  
        if (number % 3 === 0) return 'fizz'  
        return number  
    }  
}
```

```
let number = 12  
let fizzbuzz = new FizzBuzz()  
let result = fizzbuzz.transform(number)
```



The node writes a byte.



The node



subject

writes



verb

a byte



object



node



subject

write

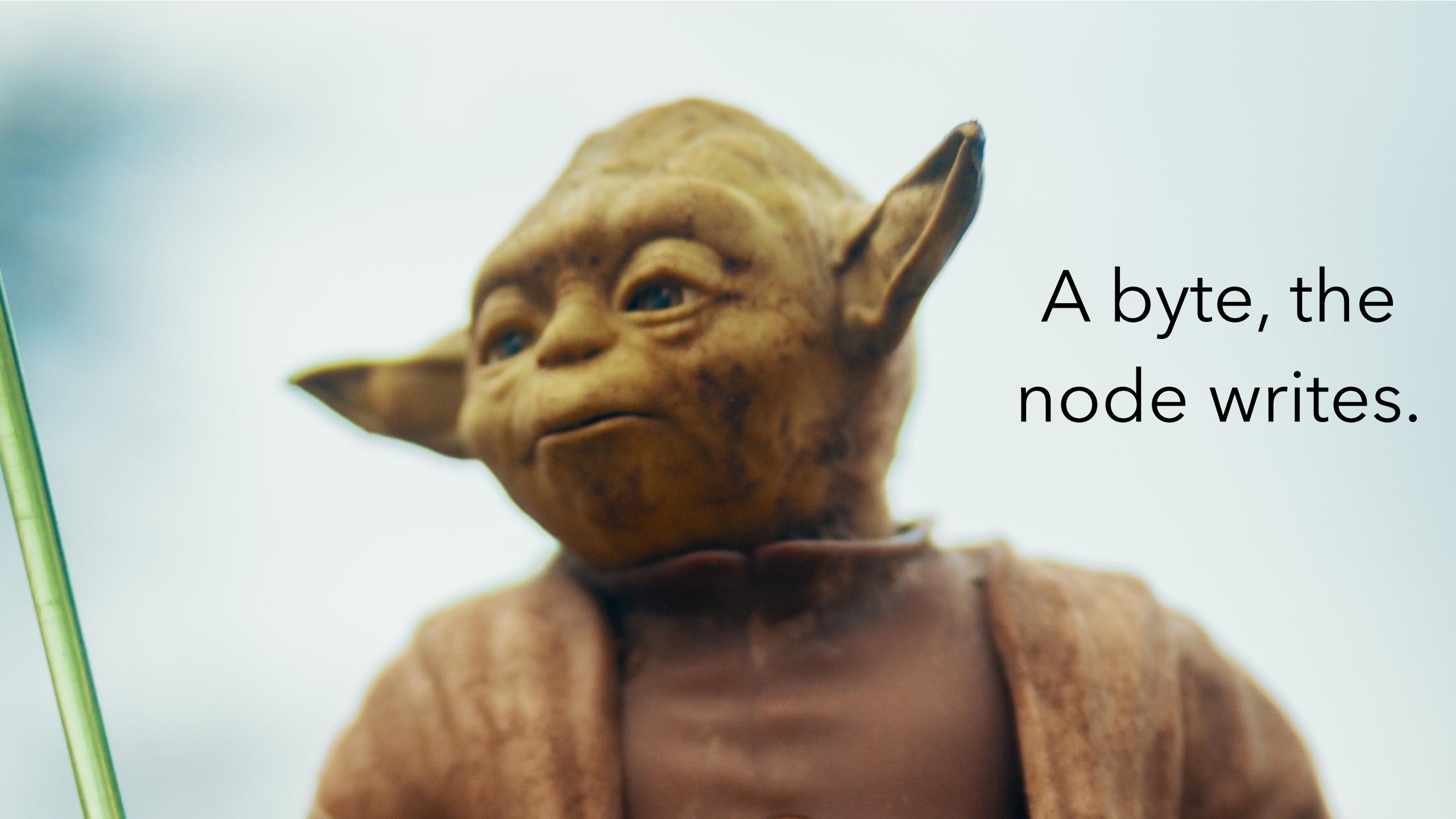


byte



object

node.write(byte)



A byte,  
the  
node writes.



byte



object

node

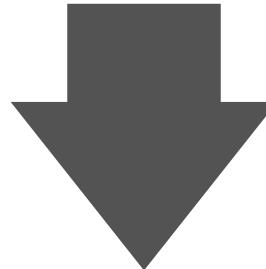


subject

write



verb



(byte) node.write

# Romanes Eunt Domus

# Nodus octetum scribit.

The node a byte it writes.



node



subject

byte

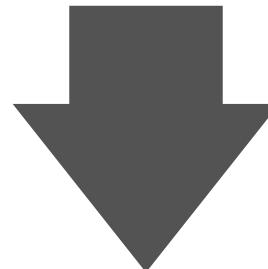


object

write



verb



node(byte).write

Nodus octetum scribit.

Nodus scribit octetum.

Octetum nodus scribit.

Octetum scribit nodus.

Scribit nodus octetum.

Scribit octetum nodus.

## Singular

### Nominative

nodus

### Genitive

nodi

### Dative

nodo

### Accusative

nodum

### Ablative

nodo

## Plural

nodi

nodorum

nodis

nodos

nodis

Singular**Nominative****octetus**Plural**octeti****Genitive****octeti****octetorum****Dative****octeto****octetis****Accusative****octetum****octetos****Ablative****octeto****octetis**



node (byte) .write

node .write (byte)

(byte) node .write

(byte) .write node

.write node (byte)

.write (byte) node



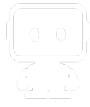
```
class FizzBuzz {  
    transform(number) {  
        if (number % 15 === 0) return 'fizzbuzz'  
        if (number % 5 === 0) return 'buzz'  
        if (number % 3 === 0) return 'fizz'  
        return number  
    }  
}
```

```
let number = 12  
let fizzbuzz = new FizzBuzz()  
let result = fizzbuzz.transform(number)
```



```
class FizzusBuzzus {  
    transforma(numerum) {  
        if (numerum % 15 === 0) return 'fizzusbuzzus'  
        if (numerum % 5 === 0) return 'buzzus'  
        if (numerum % 3 === 0) return 'fizzus'  
        return numerum  
    }  
}
```

```
let numerum = 12  
let fizzusbuzzus = new FizzusBuzzus()  
let exitui = fizzusbuzzus.transforma(numerum)
```



```
class FizzusBuzzus {  
    transforma(numerum) {  
        if (numerum % XV === 0) return 'fizzusbuzzus'  
        if (numerum % V === 0) return 'buzzus'  
        if (numerum % III === 0) return 'fizzus'  
        return numerum  
    }  
}
```

```
let numerum = XII  
let fizzusbuzzus = new FizzusBuzzus()  
let exitui = fizzusbuzzus.transforma(numerum)
```



```
genus FizzusBuzzus {  
    transforma(numerum) {  
        si (numerum % XV == 0)  
        si (numerum % V == 0)  
        si (numerum % III == 0)  
            volvitur numerum  
    }  
}
```

volvitur 'fizzusbuzzus'  
volvitur 'buzzus'  
volvitur 'fizzus'

```
sit numerum = XII  
sit fizzusbuzzus = novat FizzusBuzzus()  
sit exitui = fizzusbuzzus.transforma(numerum)
```



```
genus FizzusBuzzus {  
    (numerum) transforma {  
        si (numerum % XV == 0) 'fizzusbuzzus' volvitur  
        si (numerum % V == 0) 'buzzus' volvitur  
        si (numerum % III == 0) 'fizzus' volvitur  
        numerum volvitur  
    }  
}
```

numerum sit = XII

fizzusbuzzus sit = FizzusBuzzus() novat

exitui sit = fizzusbuzzus (numerum) .transforma



# Romani Ite Domum