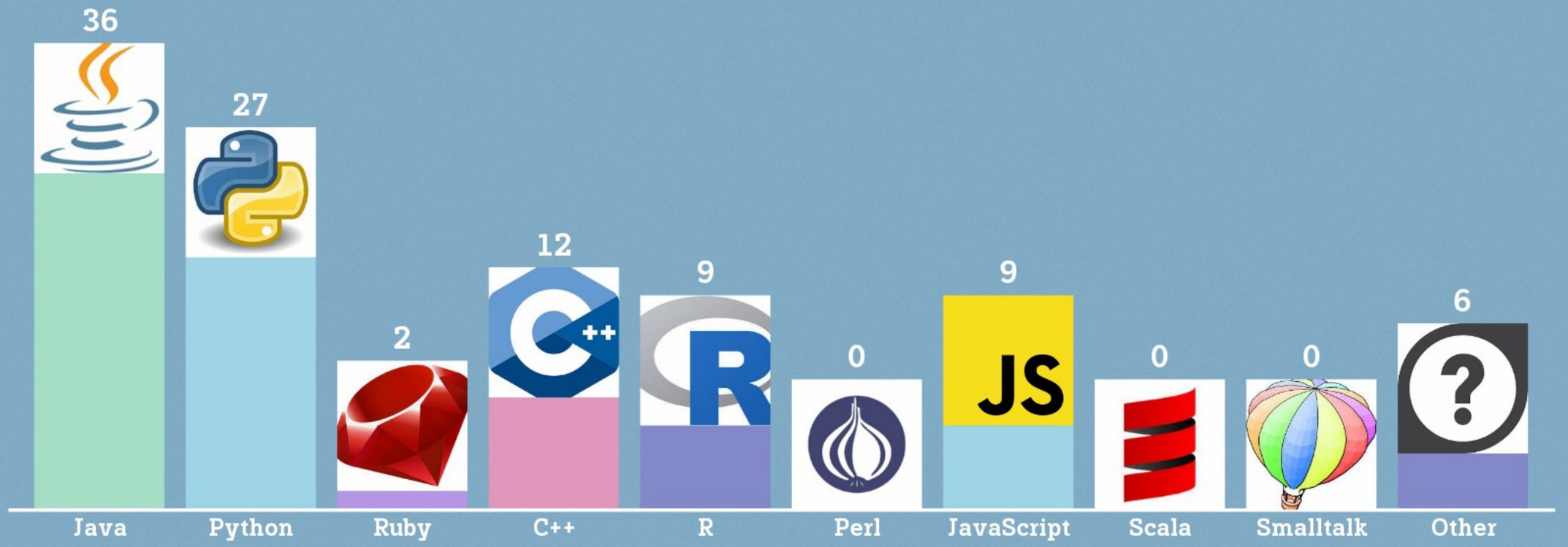
Ask me anything

Mentimeter

2 questions
4 upvotes

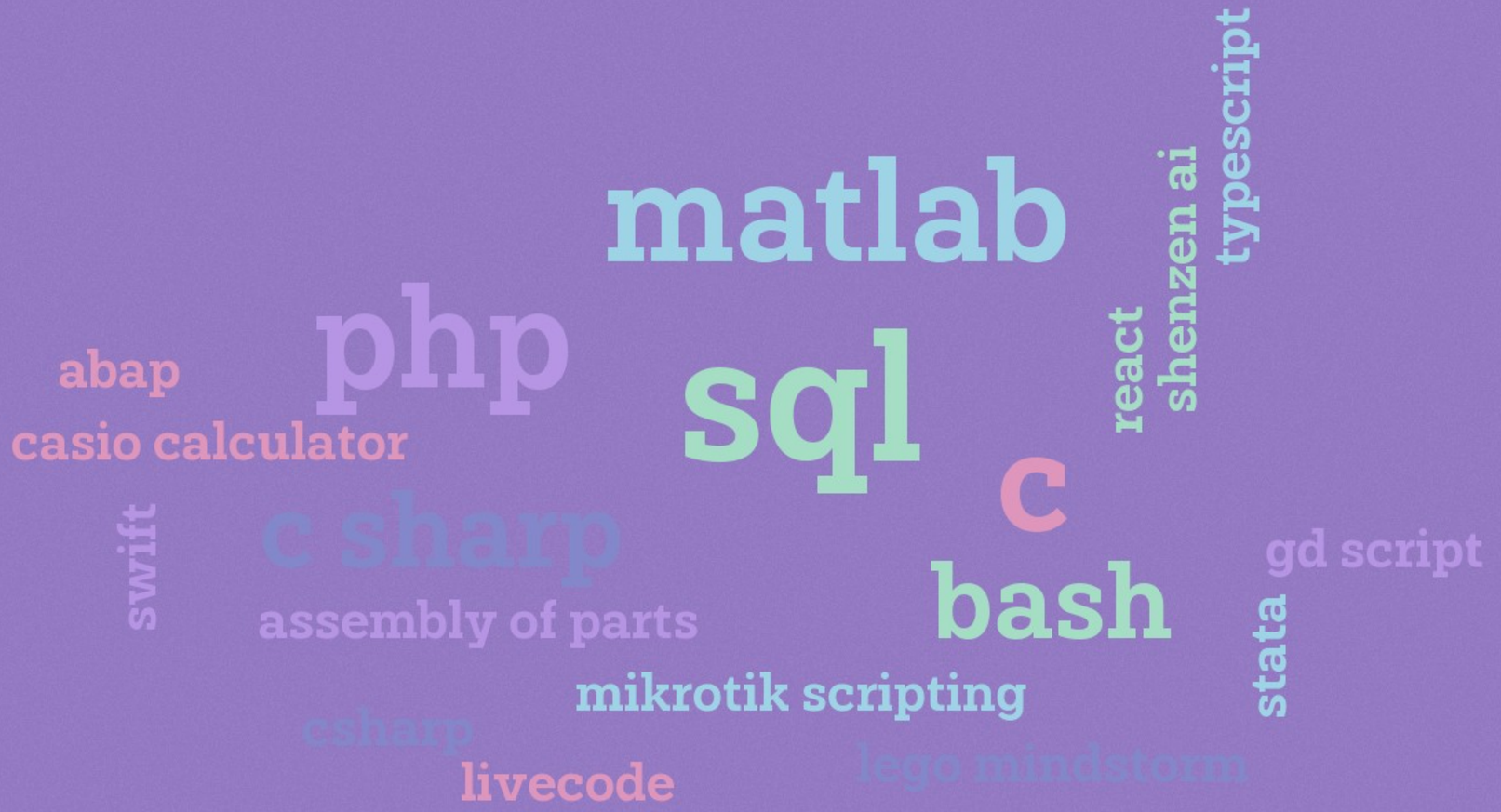
What languages have you programmed in?

Mentimeter



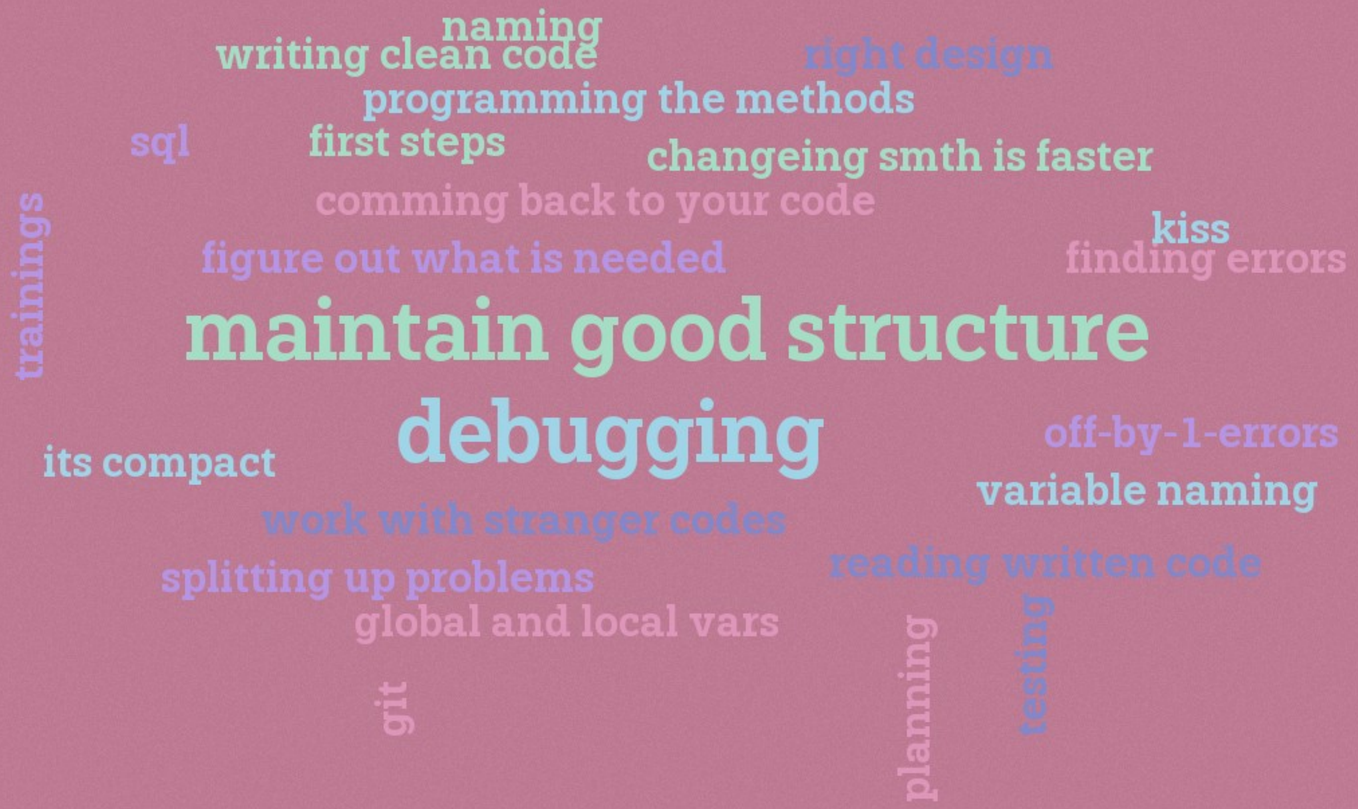
37

Any languages not mentioned so far?



24

What is the hardest part about programming?



Procedural

```
double size() {
    double total = 0;
    for (Shape shape : shapes) {
        switch (shape.kind()) {
        case SQUARE:
            Square square = (Square) shape;
            total += square.width * square.width;
            break;
        case RECTANGLE:
            …
        case CIRCLE:
            …
            break;
        }
    }
    return total;
}
```

Object-oriented

```
double size() {
    double total = 0;
    for (Shape shape : shapes) {
        total += shape.size();
    }
    return total;
}
```

```
public class Square extends Shape {
...
    public double size() {
        return width*width;
    }
}
```

What are the pros and cons of the procedural and OO designs?

Mentimeter

better structure

OO: faster to channge, you need to change it once and not in every Methode

OO: easily add further shapes

OO con: Fragmentation

OO: reusability

OO con: can be more time consuming to programm

OO --> easier to change things in an object, without having to think about the rest of the code

Procedural: everything together, i.e switch caseOO: you can split the programs and test faster certain methods

procedural -> everything is in one place, you dont forget to implement something

What are the pros and cons of the procedural and OO designs?

Mentimeter

OO: access modifiers

Proc: You can change it individual

OO: Overloading

OO pro: easier to add new methods

Oo Its compact, changeing goes faster

Procedural: compact codeblock

oo: can cause strange errors

a procedural approach may be closer to one's intuitive way of solving a given problem

proc: easy program flow

23

What are the pros and cons of the procedural and OO designs?

Mentimeter

OO: the object can become many many methodes, and can become a God Objects

procedural: easy to get some quick results for a very specific problem

Because it schows, that the refactoring was not done well

because it probably could be broken down into more smaller methods

refactoring was not done well

23

Why are long methods a bad code smell?

Mentimeter

They are incomprehensible.

hard for others to understand

more difficult to debug

Prone to errors

Easily loose the overview.

readability

More than 1 task

more errors during coding

it takes more time to understand/debug

21

Why are long methods a bad code smell?

Mentimeter

probably take on more tasks than they should

Slower?

hard to edit without breaking the whole thing

likely code duplicates

the have a big aera of effect

More opportunities for mistakes

They're probably doing multiple steps.

Because they could be broken up in easier to understand subtasks.

harder to reuse

21

Why are long methods a bad code smell?

Mentimeter

use more space

refacotring was not done well

whats up-front design?

21

Mentimeter

Last chance for questions