Maxim Valcke

@_iwan_refmt

PieSync

Refmt

Practical Interpretation of Code Formatting

Code formatting is hot 🔥

- Gofmt
- Prettier (js, markdown, css…)
- Rustfmt
- Elm-format
- …

Code formatter?

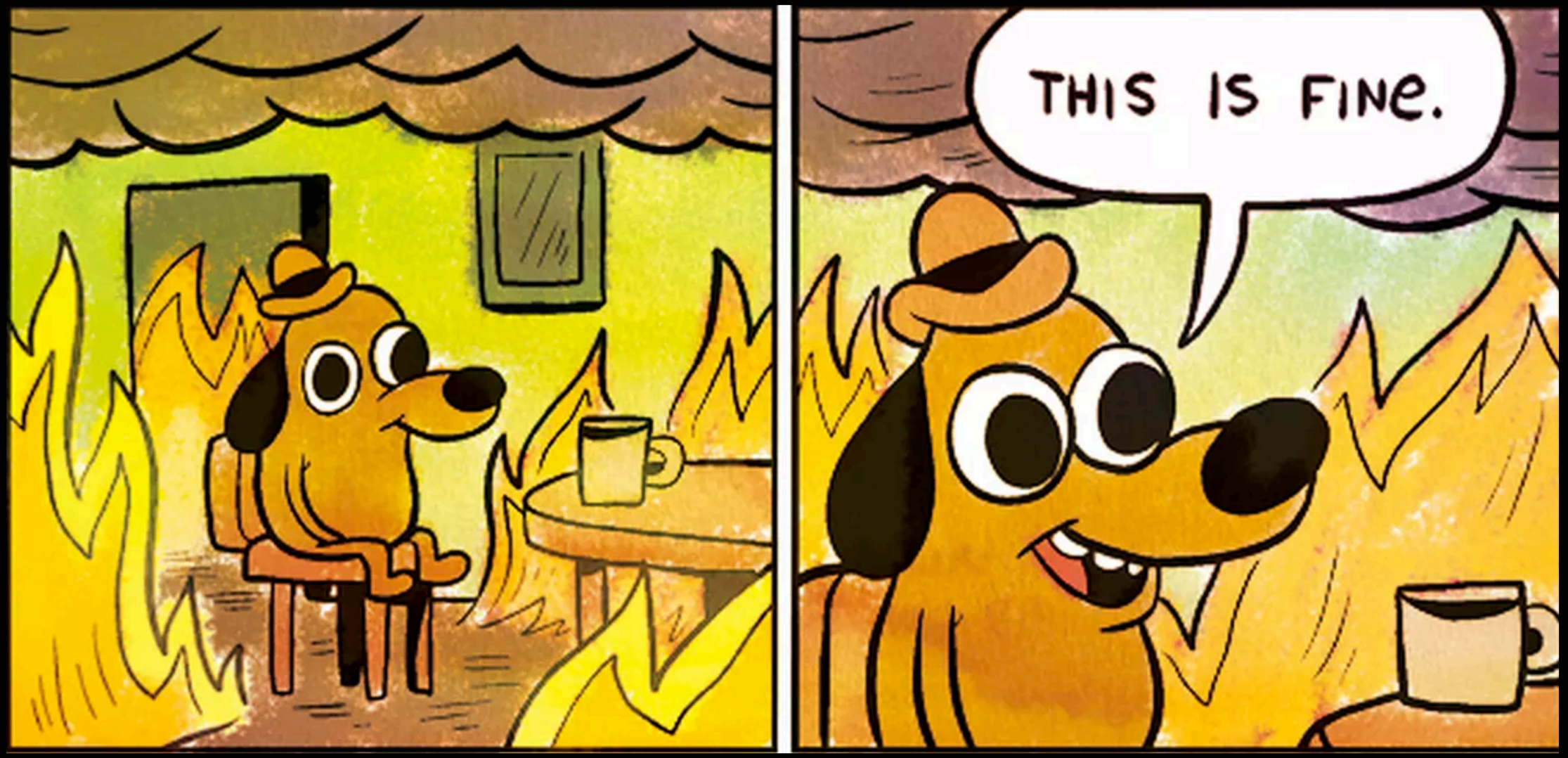
- Compiler?
- Linter?
- Code indenter?
- Magical shortcut in your editor?
- AI?

Code formatter


```
let
  add
  =
(a, b)    =>

    a
+
  b;
```

```
let add = (a, b) => a + b;
```

What if it’s just hype?



What if it’s just hype?

- Look at what the best in the world are doing
- Be very curious about the basics
- What is the desired effect? What is the actual effect?

Programming 101

Programming 101

- we create programs to direct processes
- Programming languages
  -> a way of getting a computer to perform things
  -> medium for expressing ideas
- control the complexity of large software systems
- solving problems

What I think I look like

Solving problems

What I actually look like

Whitespace
Indentation
Braces
Parens
…

Which things are worth fretting about?

What is the desired effect?

- **Write** distraction free
- Programs are made for humans to **read**
  (and only incidentally for machines to execute)
- **Automate** syntax related changes

What is the actual effect?

Refmt in production

Distraction free writing

```
/* TODO: other types of newlines */
let newline = "\n";

let

findNewlines =s=>{let arr: array(int) = [||]; let i = ref(0);while (i^ > (-1)){i :=
    Js
      .String
.indexOfFrom(
  s,
  i^ + 1


,


       newline);

if (i^ > (-1)) {ignore(
Js.Array.push
(i^, arr)); };
  };

arr; };
```

Distraction free writing

```
/* TODO: other types of newlines */
let newline = "\n";

let findNewlines = s => {
  let arr: array(int) = [|||];
  let i = ref(0);

  while (i^ > (-1)) {
    i := Js.String.indexOfFrom(s, i^ + 1, newline);

    if (i^ > (-1)) {
      ignore(Js.Array.push(i^, arr));
    };
  };

  arr;
};
```

Painless reading

```
compiler##hooks##afterEmit##tapAsync(
  "UnsafeAssets",
  (compilation, next) => {

  let c = config(compilation);

  compilation |> Plugin.buildAssets |> Js.Json.stringify |> Node.Fs.writeFileAsUtf8Sync(_, c##path);

  next();
  }
);
```

Painless reading

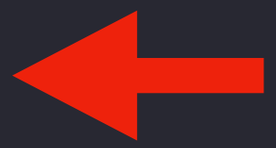
```
compiler##hooks##afterEmit##tapAsync("UnsafeAssets", (compilation, next) => {
  let c = config(compilation);

  compilation
  |> Plugin.buildAssets
  |> Js.Json.stringify
  |> Node.Fs.writeFileAsUtf8Sync(_, c##path);

  next();
});
```

Painless reading

```
compiler##hooks##afterEmit##tapAsync("UnsafeAssets", (compilation, next) => {
  let c = config(compilation);

  compilation
  |> Plugin.buildAssets
  |> Js.Json.stringify
  |> Node.Fs.writeFileAsUtf8Sync(_, c##path);

  next();
});
```

Painless reading

```
compiler##hooks##afterEmit##tapAsync("UnsafeAssets", (compilation, next) => {
  let c = config(compilation);

  compilation
  |> Plugin.buildAssets
  |> Js.Json.stringify
  |> Node.Fs.writeFileAsUtf8Sync(_, c##path);

  next();
});
```



Painless reading

```
compiler##hooks##afterEmit##tapAsync("UnsafeAssets", (compilation, next) => {
  let c = config(compilation);
  compilation
  |> Plugin.buildAssets
  |> Js.Json.stringify
  |> Node.Fs.writeFileAsUtf8Sync(_, c##path);
  next();
});
```

Boring maintenance

Automate arbitrary syntax changes

```
/* old syntax */
let add a b => a + b;

/* current syntax */
let add = (a, b) => a + b;
```

Maxim Valcke

@_iwan_refmt

PieSync

What’s next?