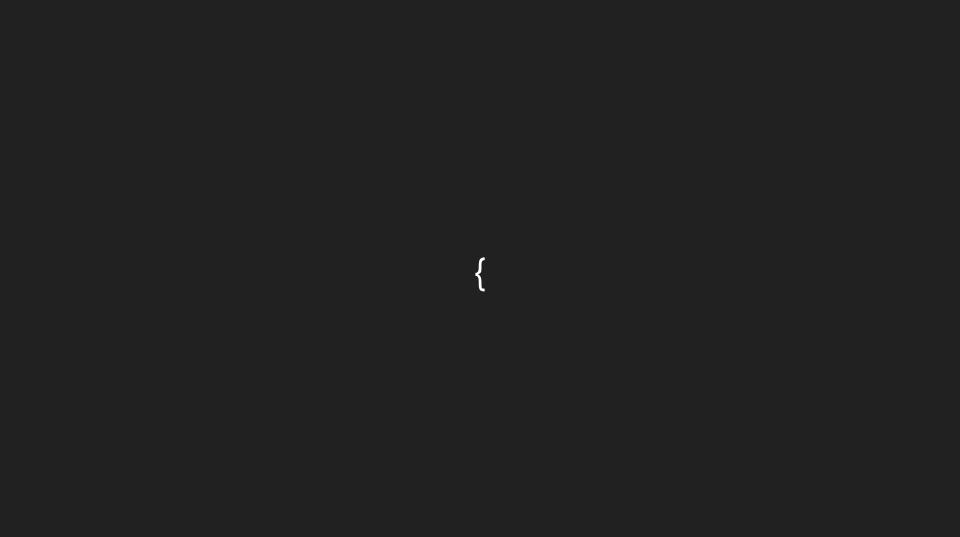
# ECMAScript design process

**ECMAScript Vision** 



How does TC39 evaluate proposals?

# Provide guidance on:

- Prioritization of usages
- The cost of growing the language
- Syntax vs APIs

What usages do we prioritize?

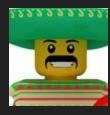
# Approach: Weighted User Personas

# Weighted User Personas: Examples



### Stanley Armstrong

- Builds toys for children
- Owner of world's largest rubber band ball



### Randy Ferguson

- Head of Application
  Security at FriendFace
- Builds static analysis tools for JS in free time



### Felecia Murray

- Lives on Mars in 2039
- Works on SPA for interplanetary event planning



## Sue Stephens

- Marketing manager for San Francisco startup
- Uses a CMS to maintain the corporate website

# The cost of growing the language

# Language growth

User-extensible features tame core language complexity

- Iterator protocol
- Value types

Others features arguably not so much

- Object.observe
- Most new builtins

# When do we allow ourselves to remove old features?

When are we willing to reject valuable features just because they grow the language?

## Mark has answered:

https://mail.mozilla.org/pipermail/es-discuss/2015-June/043307.html

When do we add syntax vs APIs?

# Syntax vs APIs: example criteria

- Composition
- Ergonomics
- Complexity of API
- Combinatorial syntax
- Frequency of use
- Domain specificity
- Static predictability
- Necessity

# Some specific design goals

# No hidden mutable state or IO in primordials

Goal:

Goal:

No undefined behavior

Goal:

Reject malformed programs

# Reject malformed programs

- const a = 0; a = 1;
- /a{4,2}/
- References to out-of-scope private fields

# 

# At least:

# Make new features enforce strict mode

# ldeally:

# Don't make new features available in sloppy mode

