

Exploring Dynamism

Allison Randal
*Parrot Foundation &
O'Reilly Media, Inc.*

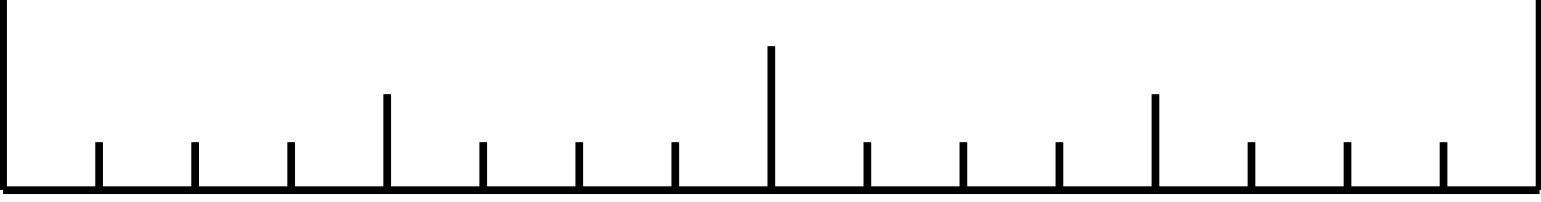
Exploration

- Thought exercise
- Defining dynamism
- Reexamine theory
- Linguistic schism
- Particle vs. wave

static

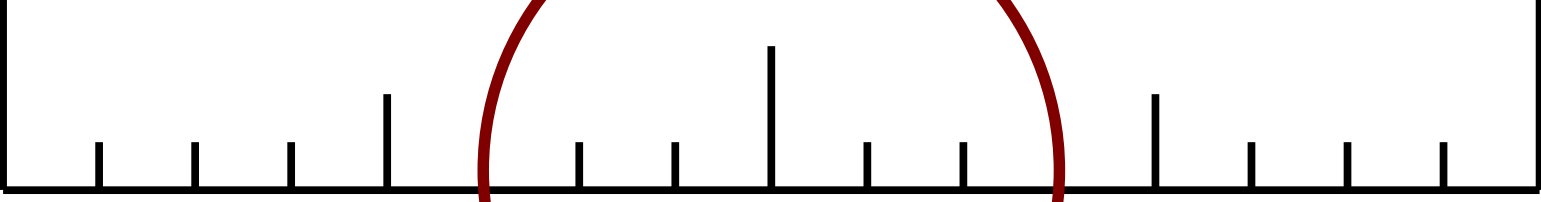
dynamic

static



dynamic

static



dynamic

Dynamic Typing

- “Dynamically Typed Languages”
- Red herring
- Least interesting
- Strong, strict, manifest
- Pre/post conditions

Dynamic Dispatch

- Code selection
- Symbolic lookup
- Optional args
- Symbolic args
- Aggregating params, dissecting args
- Multiple dispatch
- PIC

Introspection

- Source information
- Annotations
- Runtime internals
- Meta-model

Dynamic Compilation

- Eval
- Closures, continuations, coroutines
- Higher-order functions
- JIT
- Interactive
- Reflection API
- File-based
- Self-modification

Dynamic Loading

- Extension
- Linking
- Name binding
- Mixins/traits/roles
- Inheritance/type hierarchy
- Unloading

Conclusions

- Both important
- Tighter control vs greater abstraction
- Productivity vs performance
- Problem set
- New/evolving
- Hybrid
- Work ahead

Workshop

- Other characteristics?
- Languages
- Challenges, preconceptions
- Theory misfits
- Possible futures

Questions?

- Further Reading
 - “On the Revival of Dynamic Languages.” O. Nierstrasz, A. Bergel, M. Denker, S. Ducasse, M. Gälli, and R. Wuyt.
 - “Static Typing Where Possible, Dynamic Typing When Needed: The End of the Cold War Between Programming Languages.” E. Meijer, P. Drayton.
 - “Securing Web Applications with Static and Dynamic Information Flow Tracking.” M. S. Lam, M. Martin, B. Livshits, and J. Whaley.