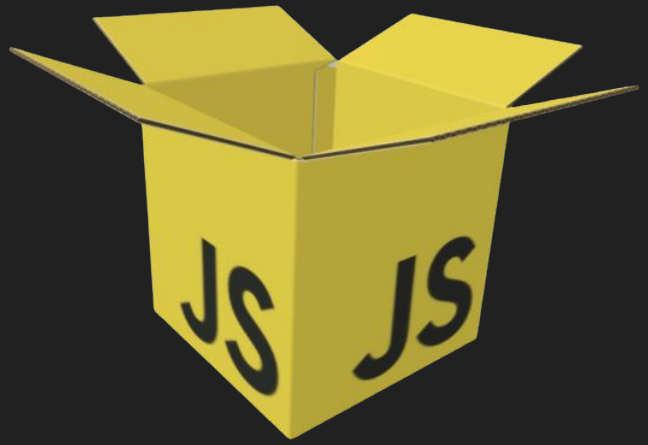
Record & Tuple

Stage 2 Update @ TC39 Tokyo 😢

Robin Ricard & Rick Button
Bloomberg

Advisor: Daniel Ehrenberg
Igalia

Box



Boxmaker, Boxmaker….

Boxes - a refresher

```
const object = {
    bar: () => "hello world",
};

const record = #{
    foo: Box(object),
};

assert("hello world" === record.foo.unbox().bar());
```

Boxes - viable!

- Previously, Boxes considered non-viable because they break membrane guarantees; membranes need to do linear work to perform a distortion on a Record/Tuple (if it contains a Box).
- Since this work already happens for objects via shadow target construction, it is acceptable for membranes to create new Records/Tuples and apply membrane to inner objects.
- This can be improved with a check for whether a Record/Tuple contains a Box, i.e. `Box.containsBox`, to eliminate extra work.
- Open questions around how boxes interact with Readonly Collections.

Boxes -

- Previou… …mbrane guarant… …rtion on a Record/…
- Since this work already happens for objects via shadow target construction, it is acceptable for membranes to create new Records/Tuples and apply membrane to inner objects.
- This can be improved with a check for whether a Record/Tuple contains a Box, i.e. `Box.containsBox`, to eliminate extra work.
- Open questions around how boxes interact with Readonly Collections.

Primitives vs Objects

Primitives vs Objects

- Currently, Records and Tuples are primitives, i.e. “not objects”.

- Desirable because primitives don’t have identity, they are observably equivalent to other primitive values with the same type and “contents”.

- Objects have identity; you can observe this via ==/===/Object.is, implemented in SameValueNonNumeric.

Identity-less objects?

- Instead, what if Records and Tuples were “identity-less” objects that:
  - Are compared by contents (like primitives)
  - Don’t have observable identity
  - Are always frozen, never have internal slots that change.
  - Can’t be used as keys in WeakMaps, entries in WeakSets, or targets of WeakRefs.
  - Can’t contain private fields.

- This means:
  - API stays the same, it is still a TypeError to put a “regular” object into a Record/Tuple.
  - No wrapper objects. Since Record and Tuple are already objects, they do not need wrapper objects.

Call for Participation: Monthly Record & Tuple Call

Discussion of Record and Tuple proposal and related work. We are interested in feedback from potential users, library authors, and implementers. Please attend if interested!

Sign up via Doodle, link will be posted in Reflector.

Started SpiderMonkey implementation work

Very early, much work to be done.
S/O to **Nicolò Ribaudo** for the awesome work! 🚀🚀🚀

https://phabricator.services.mozilla.com/D87232
https://phabricator.services.mozilla.com/D87272
https://phabricator.services.mozilla.com/D87586
https://phabricator.services.mozilla.com/D87737
https://phabricator.services.mozilla.com/D87762

Questions?