Extending null

Gus Caplan / OpenJS Foundation

History

- ES2015 intended to allow `extends null`, meaning a class which does not inherit from `Object.prototype`
- Various discussion, commits, reverts, etc. since then
- It is still the consensus of the committee that this should be supported
- Class fields do not currently work with `extends null`

```
class X extends null {}

const x = new X();

assert(x.hasOwnProperty === undefined);
```

Moving Forward

- Do we want to fix this or be done with it?
- https://github.com/tc39/ecma262/pull/1321

`super()` Problems

Conflicting positions previously expressed by delegates:

- `super()`, when syntactically allowed (ClassHeritage present), should always be valid at runtime (shouldn't throw)
- `super()` should throw in classes which extend null

```
let Foo = null;

class X extends Foo {}
```

- As proposed, allows (but does not require) `super()` and preserves `super() === this`

Mutable Prototype Behavior

```
class Foo extends null {}
Object.setPrototypeOf(Foo, Function.prototype);
Object.setPrototypeOf(Foo.prototype, Object.prototype); // as proposed, `new Foo().hasOwnProperty` exists

class Bar {}
Object.setPrototypeOf(Bar, null);
Object.setPrototypeOf(Bar.prototype, null); // as proposed, `new Bar().hasOwnProperty` does not exist

class Baz {}
class Qux extends Baz {}
Object.setPrototypeOf(Qux, null);
Object.setPrototypeOf(Qux.prototype, null); // as proposed, `new Qux()` throws
```

Discussion