

Notes - 4/19

Friday, April 19, 2013 9:42 AM

[In 41/4541 at 10:00 today.]

Agenda:

- Spec and implementation updates
 -
-
- Close on topics from mail threads:
 - Best common type - agreed to return to 0.8 implemented behaviour
 - Assume for now what's in mail.
 - Should "var x: Array" be allowed?
 - No!
 - Should function and class variables be read-only? Currently they are, but there is no such notion in the spec.
 - Putting on the backburner, current rules are practically useful but
 - Should shadowing of type parameters be an error?
 - No.
 - Should there be a betterness rule that prefers overloads with non-optional parameters over those with optional?
 - No.
 - [509678](#) and [509590](#): Should call, construct, and index signatures be allowed in classes?
 - Indexing is allowed.
 - Call and construct no.
 - Implemented softer rule for re-declaration
 - The type of all declarations with the same name in a scope must be identical
 - Don't special case any
 - Apply this to the global as well [implementation]
- Review of breaking changes from Monaco/TFS migrations from Paul
- Protected
 - Yesish?
- [No implicit 'any'](#).
- Modules codegen [\[509700\]](#)
- "-out" for external modules (and separating "-outdir")
- ES6 [Module Alignment Analysis](#) ([Web view](#))
- Other potentially open items for 1.0:
 - Recursive type check improvements
 - Extending 'declare var'
 - Remaining class alignment deltas:
 - [Class expressions](#)
 - Extends expressions
 - "extends null"

```

ib.d.ts  app.ts*  *  X
<global>

interface Accumulator {
    clear(): void;
    add(x: number): void;
    result(): number;
}

function makeAccumulator(): Accumulator {
    var sum = 0;
    return {
        clear: function () { sum = 0; },
        add: function (value) { sum += value; },
        (parameter) value: any { return sum; }
    }
}
  
```

```

ib.cts fappis' o
cglob&>
interface Accumulator {
  clear: void;
  add(x: number): void;
  result(): number;
  function makeAccumulatorQ:
  Accumulator {
    var sum = 0;
    return {
      clear: function () { sum = 0; },
      add: function (value) { sum +=
      value; },
      (parameter) value: any { return
      sum; }
    }
  }
}
  
```