Notes - 1/11/2013

Thursday, January 10, 2013 7:57 PM

[In 41/2731 at 10:00 today.]

Agenda:

- Implementation update
  - Fidelity, pull model in progress
  - Generics prototyping starting in a few weeks

- Recap of remaining TypeScript v1.0 language topics
  - Generics [in spec]
    - Type parameters in scope in the entire declaration, even "backwards":
      - Foo<T extends Bar<U>, U>() {}
    - "var x: void; " is now legal.
    - var x: Array = [1,2,3]
      - Does this fully alleviate breaking changes related to switch to generics?
    - new Array<string>(3);
    - Naked type constraints?
      - Not currently, maybe could open this up more in the future.
    - Interesting issue for:
      - <T extends List<T>>(x: T, y: T): T[]
      - Substitution of bounds may not work naively
    - Ambiguity discussion
      - Ask Brian how common this is?
  - Modules and namespacing [in spec]
    - Technically a breaking change here because modules no longer define a type
  - Removal of brands and shift in meaning of privates [in spec]
    - Nice SelectableControl example - will be in spec
    - This enables some of the "type-only mixins" scenerios we've seen many asks for - worth validating what's possible/not here now.
  - Modules implementing interfaces
  - Enums changes
  - Overload on string constants
  - bool -> boolean
  - Readonly in type system
    - Specific questions like "are functions readonly"?
  - "export ="
    - Some open issues
  - 'this'-types
    - class C { foo(): C }
    - class D extends C { bar(): D }
    - d.foo().bar() // error - but shouldn't be
    - class Foo
    - Issue?
      - var x = this;
      - x = new C();
      - return this;

- Should we create a 0.9 branch and put the spec plus early prototyping in there?

- Other persistent feedback areas:
  - Union types
  - Generators
  - Mixin models

- Design backlog
  - Bringing back explicit “this” parameter type
  - … for arguments
  - String interpolation
  - Type-only mixins
  - Decorators
  - Static initialization (#74)
  - Thinking on how to approach async via generators (#38)
  - SkyDrive feedback – implications for future thinking?
- Others?