CeylonWeb Notes

* Component based.
* No html templates, only code.
* Model objects need not inherit or implement anything special. After events have been processed each bound value is compared to last known value to detect change.
* Components are composed to build a template.
* A template can be turned into a component.
* A template can be instantiated to produce a DOM fragment.
* Template instantiation is a simple clone of a DOM fragment and a iteration over the cloned nodes to attach behavior (such as binding text, attributes and properties to model). Cloning should be a lot faster compared to building a DOM fragment from scratch.
* Model is bound to view in a type safe manner.
* Components and templates are typed through a generic type parameter. A template gets the same type argument as the component that created it. If the template is turned into a component, the resulting component gets the same type argument as the original component. This type can for example be used to enforce html flow and paragraphing content rules.

Issues

* A lot of generics are needed to get it type safe.
* Potential performance issues if a page has too many bindings. Interceptors might enable a more efficient implementation.
* A lot of JavaScript code is generated…

Features that would be nice:

* Interceptors
* #749 “cross-references within named argument lists”.
* Lower bound type constraints (given T abstracts …).
* Not needing a dynamic block to assign a dynamic to another dynamic.