Web Programming

Week 8

"They call it optional typing but you still have to type in your code."

Scott Davis on Groovy

JS Goodie

Today: Types in JS

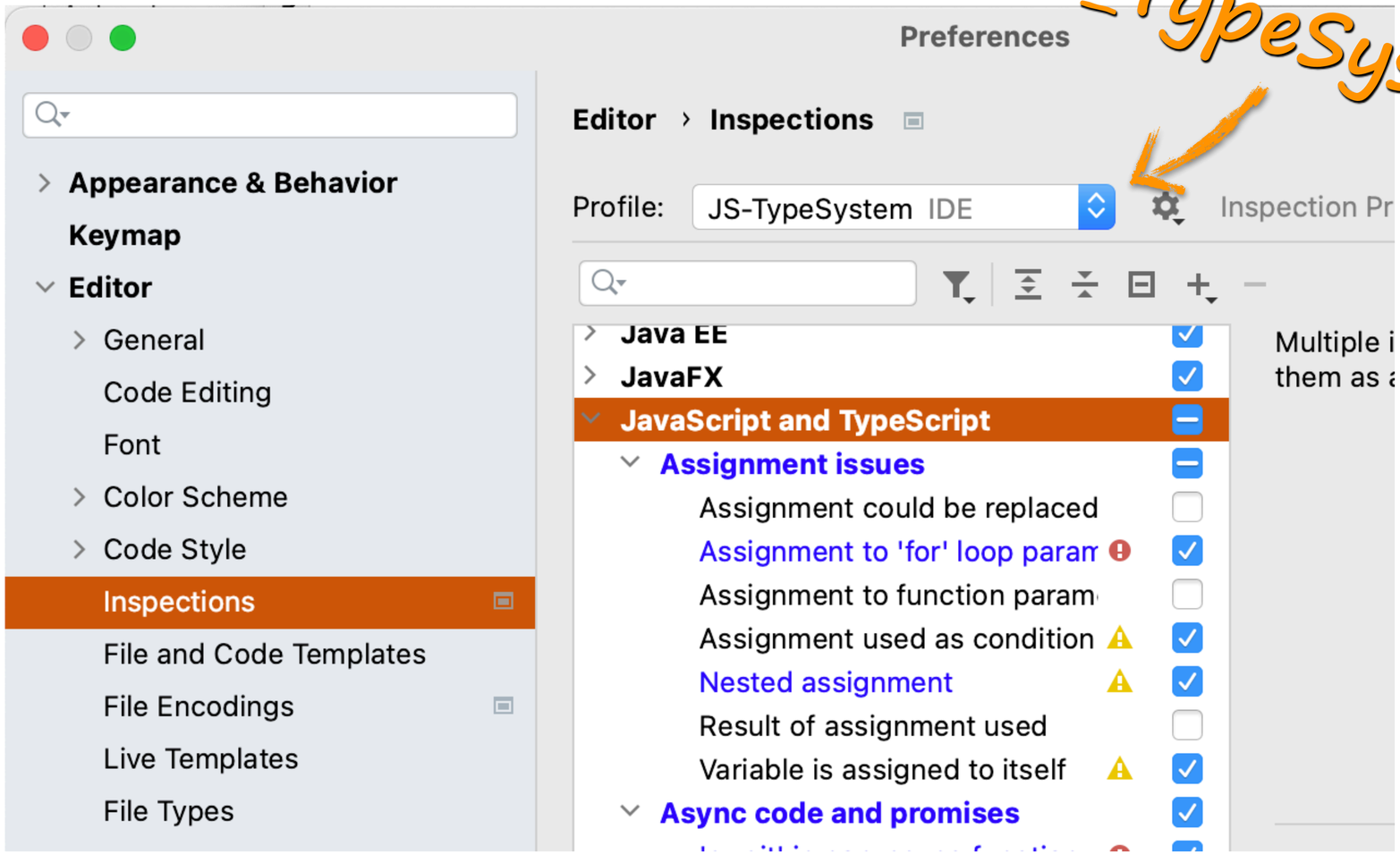
Define types in JsDoc

JsDoc is optional -> typing is optional

Not compiler enforced but IDE support

IDE Setup

JS_TypeSystem.xml



Overview

Unityped: built-in types

Functions, data structures, interfaces

Callbacks, generics, type casts

Built-in Types

Literal Constructor Operators

Boolean, Number, String, Object, BigInt, Symbol, Null, Undefined (typeof)

Object "classes": Function, Array, Date, RegExp, Error, Map, Set, JSON, ...

Functions (classical)

@function, @constructor, @callback

@param

@return

! non-null ? optional ... any number

Functions (curried)

```
@type {    (A) =>     (B) =>  C }
@type { (a:A) => (b:B) =>  C }
@type { (a:A<T>) => (B) =>  C }
@type { <T> (a:T) => (b:B) =>  T }
```

Data Structures I

Objects:
@property { Number } age - ...

Array:
{ Array<Number> }, { [Number] }

Data Structures II

Union type: { Number | String }

Intersection type: { Person & Cool }

String literal type: { "me" | "you" }

special notations...

Interfaces

@typedef

even the IDE show it like an Interface

Callbacks

@callback

a name for a function type that will be used in parameter position

Generics

@template T or <T> notation

similar to Java Generics but the processing (unification, reification) is unclear

Type Casts

@type

used locally to help the type checker

Best Practices I

Set up your IDE appropriately.

Follow conventions (type names capitalized, syntax variant, etc.)

Distinguish constructor from constructed object type.

Best Practices II

@example

{@link xxx}

Validate via test case.

Push only on clean inspection state.

Putting it all together

Have a look at Kolibri for a fully typed codebase with complex use cases.

Practical Work

JsDoc your codebase,
observe the inspection findings