**TypeScript** is a typed superset of JavaScript” but it assumes you know what a “superset” is and what “typed” means. Instead, to keep things simple you can think of TypeScript as of “a layer on top” of JavaScript.

TypeScript is a layer because you can write TypeScript code in your editor. After a compilation, all that TypeScript stuff is gone and you’re left with plain, simple JavaScript.

If the idea of the compilation step confuses you keep in mind that JavaScript is already compiled and then interpreted. There is a JavaScript engine that reads and executes your code.

JavaScript engines are not able to read TypeScript code so any TypeScript file should go under the “pre-translation” process, called compilation.

Only after the first compilation step, you’re left with pure JavaScript code, ready to run in the browser. You’ll see later how the TypeScript compilation is done.

For now, let’s keep in mind that TypeScript is a special kind of JavaScript but it needs a “translator” before running in a browser.

**And in TypeScript Microsoft was tried to solve next design Goals:**

Extend JavaScript for writing large apps (superset of JavaScript),

Adds support for classes, interfaces & modules,

Development tooling support,

Compiled JavaScript runs in any browser (or Node.js),

Since JavaScript code is TypeScript code you can start off with JavaScript and just add some types here and there.

**And right now I show how to install TypeScript on yours laptop or PC:**

for Node.js

npm install -g typescript

Provides a command-line compiler.

tsc source.ts

TypeSctipr supported in next file Extensions

.ts is the extension for source files.

.d.ts is the extension for declaration files

**So, what about Declaration Source Files in TypeScript. I can tell you about some properties:**

Provide type definitions, separate from the corresponding source.

Analogous to header files in C/C++..

Can be used to describe the exported virtual TypeScript types of a JavaScript library or module when a third-party developer consumes it from TypeScript.

Gives type safety, intellisense and compile errors.

DOM and jQuery provided with TypeScript.

Write your own for any existing JavaScript library / code.

**And on of the most important thing in any programming language is types of values.**

All types are subtypes of a single top type called the Any type.

In TypeScript there are 5 Primitive Types:

number

bool

string

null

undefined

And, of course, Object Types such as

Class, module, interface and literal types.

Supports typed arrays: var reports: Employee[] = [];

**Although, TypeScript like a JS is supported classes**

In TypeScript methods are translated into JavaScript prototype chain - more memory efficient than using closures with anonymous functions.

public or private member accessibility.

Although parameter property declarations via constructor.

And Supports single-parent inheritance.

Derived classes make use of super calls to parent.

Let's considered code example. And we can see public member, using super calls to parent.

**Let's talk about TypeScript Modules. And we can disntinguish some properties such as:**

Analagous to .NET namespaces.

Prevents global variable naming collisions.

Closely aligned with those proposed for ECMAScript 6.

Supports code generation targeting CommonJS and AMD module systems.

Accessibility for internal and external modules.

Allows exposing a public API.

Modules code example:

Source File Dependencies:

Compiler automatically determines a source file's dependencies.

Uses reference comments and import declarations.

All references are analysed for their dependencies.

Visual Studio uses these references to load all associated source files from a single .ts file.

Reference comment

So, for inImport declaration:

import log = module("log");

log.message("hello");

**Source Map Support**

Alleviates the debugging issues that are raised by \*-to-JavaScript compilers and JavaScript minifiers.

You aren't debugging the code that you wrote.

Source maps fixes this; it works like magic!

A way to map combined / minified files back to their unbuilt state.

TypeScript team have done the hard work for us by providing a Source Map generator in the compiler.

Allows debugging and breakpoints from .ts files.

tsc -sourcemap example.ts

Requires either Chrome Canary or WebKit nightly.

Not yet supported by Visual Studio 2012.

... but available via Web Essentials 2012 plugin.

**And if somebody asked about self hosting, so, I can say that:**

Compiler is written in TypeScript, so can compile itself to JavaScript.

It can be hosted in any ECMAScript 3 compatible runtime.

It can host the compiler in a web browser by referencing typescript.js

**In summary we can distinguish some pluses and minuces in TypeScript. And, of cource, we begin with pluces:**

Why use TypeScript?

Support for ECMAScript 6 constructs (classes, modules) in today's ECMAScript 3 compatible browsers.

Open Source, under the Apache 2.0 license, and available to download on codeplex.

Tooling support, for type safety, inference and refactoring.

Static types and compilation helps catch mistakes & bugs earlier.

Structural interfaces & typing.

Simplicity of design, self hosting (less than 25k LoC).

Works well with existing projects.

Compiles to idiomatic JavaScript.

Anders Hejlsburg is involved.

Enables programming in the large.

But, unfortunately,there isn't a perfect language, so, typescript has following minuses:

Additional compilation step for JavaScript development.

Only at version 0.8

Bugs, for example implementing an interface ignores the access modifiers when compiling, but fails at runtime.

Limited availability of resources & libraries (can use any existing JavaScript, albeit with limited type saftey and tooling support).

No support for generics (yet, but is defined in spec).

All types are nullable.

Limited tooling support outside of Visual Studio 2012 or Monaco web editor.

Static typing does confer some very relevant value.

Typically it does so at the cost of flexibility, but this brings a lot of the benefits to JavaScript in an optional way, which is very powerful.