Typen nach JavaScript tragen TypeScript

Johannes Dienst

Probleme von JavaScript

“JavaScript (kurz JS) ist eine Skriptsprache, die ursprünglich für dynamisches HTML in Webbrowsern entwickelt wurde[...]“

Quelle: Wikipedia

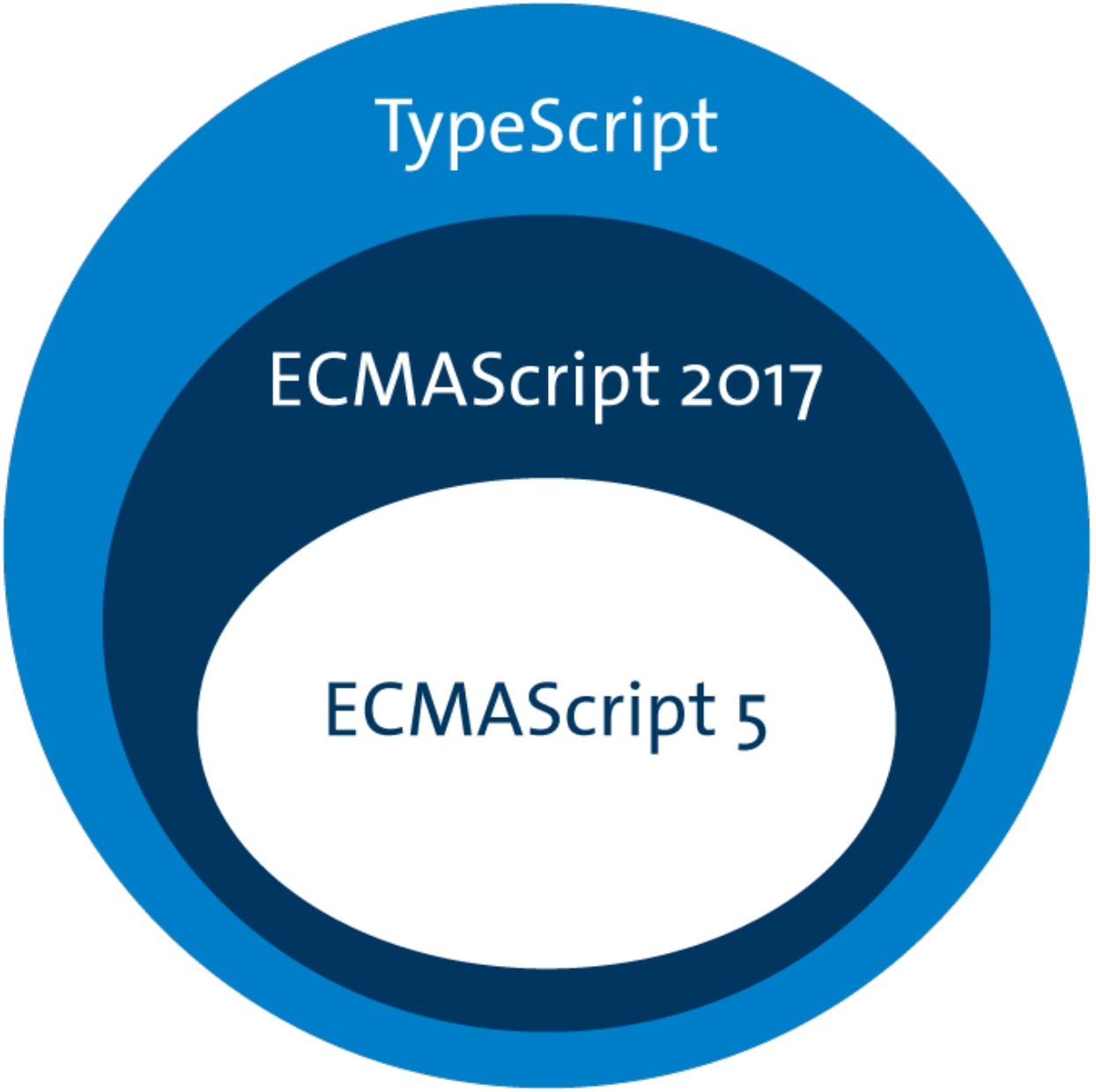
Probleme von JavaScript

Fehlendes (statisches) Typsystem

Wartung einer großen Codebasis

http://programmers.stackexchange.com/questions/221615/why-do-dynamic-languages-make-it-more-difficult-to-maintain-large-codebases

Was ist TypeScript



Warum TypeScript?

Warum TypeScript?

Typsystem - Basistypen

```
let isDone: boolean = false;

let decimal: number = 6;

let color: string = "blue";

let list: number[ ] = [1, 2, 3];

enum Color {Red, Green, Blue};
let c: Color = Color.Green;

let notSure: any = 4;
```

Typsystem - Interface

```
interface Robot {
   name: string
}

function printName(aRobot: Robot) {
   console.log(aRobot.name)
}

let myRobot =
    {name: 'Arnold', language: 'German'}

printName(myRobot)
```

Typsystem - Interface

```
interface Animal {
    name?: string
    color?: string
}

// Excess property
let myAnimal: Animal =
    {name: 'Bird', odor: 'Sweet'}
```

Typsystem - Interface

```
interface Human extends Animal {
    hugeBrain: boolean
}

let aHuman: Human =
    {name: 'Homo sapiens', hugeBrain: true}
```

Typsystem - Klassen

```
class SpecialRobot implements Robot {
   name: string

   constructor(name: string, private color: string) {
      this.name = name;
   }

   setColor(color: string) {
      this.color = color;
   }

   getColor() {
      return this.color;
   }
}
```

Typsystem - Klassen

```
let mySpecial =
    new SpecialRobot('T1000', 'metal');

printName(mySpecial);

console.log(mySpecial.getColor())
```

Funktionstypen

```
function printName(aRobot: Robot) {
   console.log(aRobot.name)
}

let printFunc: (Robot) => void;
printFunc = printName;

// Error
let printFunc2: (Robot) => boolean;
printFunc2 = printName;
```

Funktion - Goodies

```
function funcDefault(p1="World", p2?: number) {
   console.log('Hello ' + p1 + ' ' + p2);
}

// Hello Buenos dias  42
funcDefault("Buenos dias ", 42)

// Hello World undefined
funcDefault()
```

Funktion - Goodies

```
// Rest parameter
function buildName(firstName: string,
    ...restOfName: string[]) {
      return firstName + " " +
             restOfName.join(" ");
}

// T800 T1000 TX
let robotNames =
    buildName("T800", "T1000", "TX");
```

Typsystem - Eigenschaften

```
// optional
let myRobot =
    {name: 'Arnold', language: 'German'};

// structural
function printName(aRobot: Robot) {
   console.log(aRobot.name)
}

printName(myRobot)
```

Typsystem - Eigenschaften

```
// local
let bigRobot =
    new SpecialRobot(name='Big', 'blue');
bigRobot = 42 // error

// contextual
window.onmousedown = function(mouseEvent) {
    console.log(mouseEvent.buton);  //<- Error
};
```

Typsystem – Generics

```
// Generic function
function identity<T>(arg: T): T {
    return arg;
}


function loggingIdentity<T>(arg: T): T {
    // Error: T doesn't have .length
    console.log(arg.length);
    return arg;
}
```

Typsystem – Generics

```
// Generic constraint
interface Lengthwise {
    length: number;
}

function loggingIdentity<T extends Lengthwise>
    (arg: T): T {
    console.log(arg.length);  // No error
    return arg;
}
```

Typsystem – Generics

```
// Generic class
class GenericNumber<T> {
    zeroValue: T;
    add: (x: T, y: T) => T;
}

let myGenericNumber =
    new GenericNumber<number>();

myGenericNumber.zeroValue = 0;

myGenericNumber.add =
    function(x, y) { return x + y; };
```

Union Types

```
interface C3PO {
    move();
    talk();
}

interface R2D2 {
    move();
    whistle();
}

function getRobot(): C3PO | R2D2 {
    // ...
}
```

Union Types

```
let robot = getRobot();

robot.move(); // okay

robot.talk(); // error
```

Type Aliases

```
type Name = string;
type NameResolver = () => string;
type NameOrResolver = Name | NameResolver;

function getName(n: NameOrResolver): Name {
    if (typeof n === 'string') {
        return n;
    }
    else
    {
        return n();
    }
}
```

String Literal Types

```
type Easing = "ease-in" | "ease-out" |
    "ease-in-out";

class UIElement {
    animate(dx: number, dy: number,
        easing: Easing) {
        ...
    }
}

let button = new UIElement();
button.animate(0, 0, "ease-in");
button.animate(0, 0, "uneasy"); // error
```

Polymorphic this types

```
class BasicCalculator {

    public constructor(
        protected value: number = 0) { }

    public currentValue(): number {
        return this.value;
    }

    public multiply(operand: number): this {
        this.value *= operand;
        return this;
    }
}
```

Polymorphic this types

```
let v = new BasicCalculator(2)
            .multiply(5)
            .currentValue();
```

http://work.tinou.com/2009/07/wtf-is-fbounded-polymorphism.html

Interessantes

```
class Person {
    talk() {alert("Hello")}
}

class ConsoleLogger {
    log() { console.log("World!") }
}

class Mixin implements Person, ConsoleLogger {
    talk: () => void;
    log: () => void;
}
```

Interessantes

```
applyMixins(Mixin, [Person, ConsoleLogger]);

let mix = new Mixin();
mix.talk();
```

Wenig Sinnvolles

```
function isRobot(robot: C3PO | R2D2):
    robot is R2D2 {
    return (<R2D2>robot).hologram !==
        undefined;
}

// "number", "string", "boolean", "symbol"
if (typeof padding === "number") {
    return ...
}

if (c3poInst instanceof Robot) {
    …
}
```

Praxisrelevanz?

- Hybrid Types
- Interfaces erweitern Klassen
- Index-Typen

Ökosystem

Home | Guides | Directory | Repository | TSD | NuGet

DefinitelyTyped

The repository for high quality TypeScript type definitions

Usage

Include a line like this:

```
/// <reference path="jquery/jquery.d.ts" />
```

Get the definitions

GitHub repository

NuGet package manager

TypeScript Definition manager

Contributing

See the contribution guide

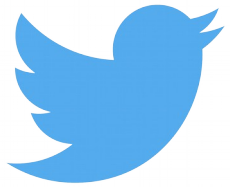
News

Add a badge to your library

TypeScript directory restructured

Probieren Sie TypeScript!

- Statisches Typsystem
- Lebendiges Ökosystem
- ES20XX Features
- Umstieg schmerzfrei



JohannesDienst

jdienst@multamedio.de

JohannesDienst.net