

TypeScript 2.{6 => 9} und die Zukunft

Roman Roelofsen - W11K GmbH - theCodeCampus

Twitter & GitHub: [romanroe](#)

Über mich

- Alpha Geek, Entwickler, Trainer
- W11K GmbH - *The Web Engineers*
- Individualsoftware
- theCodeCampus
- Schulungsanbieter Angular & TypeScript

Warum TypeScript?

JavaScript:

JavaScript

Duck Typing

- Wikipedia: "...bei dem der Typ eines Objektes nicht durch seine Klasse beschrieben wird, sondern durch das Vorhandensein bestimmter Methoden oder Attribute."
- „Wenn ich einen Vogel sehe, der wie eine Ente läuft, wie eine Ente schwimmt und wie eine Ente schnattert, dann nenne ich diesen Vogel eine Ente.“

JavaScript:

JavaScript:

DUCKOMENTA, <https://www.duckomenta.de/>

JavaScript - Typisierung

- Wann wird der Typ geprüft?
 - **statische** Typisierung:
`String s = new String();`
 - **dynamische** Typisierung:
`var s = "a"; s = 3;`

JavaScript - Typisierung

- Wie streng sind die Typen?
 - **schwache** Typisierung:
s = "Zahl " + 5;
 - **starke** Typisierung:
s = "Zahl " + str(5)

TypeScript

- Entwickelt von Microsoft
- Apache License
- **<http://www.typescriptlang.org>**
- **<https://github.com/Microsoft/TypeScript>**
- Compiler: TypeScript -> JavaScript
- Transpiler
- Keine API!
- Keine Runtime!

Hello TypeScript! JavaScript?

```
function hello(name) {  
    alert("Hello " + name + "!");  
}
```

Hello TypeScript!

```
function hello(name: any) {  
    alert("Hello " + name + "!");  
}
```

```
function hello(name: string) {  
    alert("Hello " + name + "!");  
}
```

Basic Types

```
var isDone: boolean = false;
var height: number = 6;
var name: string = "bob";
var list: number[] = [1, 2, 3];

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

Type Inferencing

```
var b1: boolean = false;  
var b2 = true;  
  
b2 = b1;
```

Strukturelle Typisierung

```
var person = {  
    firstname: "Max",  
    lastname: "Mustermann"  
};
```

```
interface Person {  
    firstname: string;  
    lastname: string;  
}
```

```
var person: Person = {  
    firstname: "Max",  
    lastname: "Mustermann"  
};
```


Strukturelle Typisierung

```
var person: Person = {  
    firstname: "Max",  
    lastname: "Mustermann"  
};  
  
function hello1(person: Person) {  
    alert("Hello " + person.firstname + "!");  
}  
  
function hello2(person: {firstname: string}) {  
    alert("Hello " + person.firstname + "!");  
}
```

Klassen

```
class Person {  
    firstname: string;  
    lastname: string;  
}  
  
var person = new Person();  
person.firstname = "Max";  
person.lastname = "Mustermann";
```

Klassen - Strukturelle Typisierung

```
class Person {  
    firstname: string;  
    lastname: string;  
}  
  
var person: Person = {  
    firstname: "Max",  
    lastname: "Mustermann"  
};
```

Klassen - Strukturelle Typisierung

- Achtung!

```
class Person {
  firstname: string;
  lastname: string;
}

var person: Person = {
  firstname: "Max",
  lastname: "Mustermann"
};

console.log(person instanceof Person); // false
```

TypeScript Advanced

DUCKOMENTA, <https://www.duckomenta.de/>

Code: Constructor Modifier

Code: Intersection Types/Union Types/Type Guards

Code: Type aliases

Code: Nullable types

Code: Literal types

Code: polymorphic this

Code: this parameter

Code: Index types

Code: Mapped types

Code: Decorator

Code: Conditional Types

Ausblick

DUCKOMENTA, <https://www.duckomenta.de/>

unkown top type

type-safe Gegenstück zu `any`

```
let zahl: any = 1;  
let text: string = zahl;
```

Jetzt mit `unkown`

```
let zahl: unkown = 1;  
let text: string = zahl; // COMPILER FEHLER!
```

Generic rest parameters

```
declare function bind<T, U extends any[], V>(f: (x: T, ...args: U) => V, x: T): (...args: U) => V;

declare function f3(x: number, y: string, z: boolean): void;

const f2 = bind(f3, 42); // (y: string, z: boolean) => void
const f1 = bind(f2, "hello"); // (z: boolean) => void
const f0 = bind(f1, true); // () => void

f3(42, "hello", true);
f2("hello", true);
f1(true);
f0();
```

Viel Spaß beim Ausprobieren!

Roman Roelofsen - W11K GmbH - theCodeCampus

@romanroe