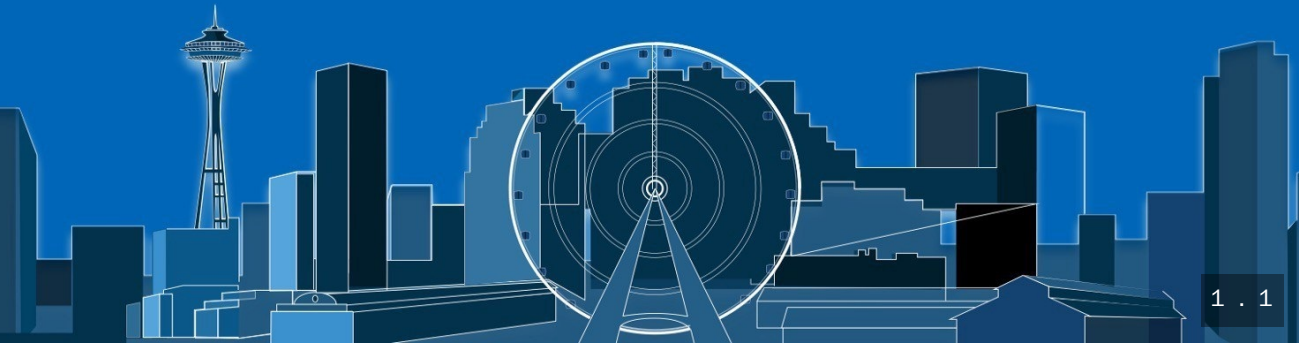


TypeScript

Starke Typen für JavaScript



Speaker notes

Timing:

Aktuelle Slide: 00:01:30

Slide Ende: 00:02:30

Typescript (TS)

Microsoft will in Frontend Entwicklung stark beteiligt sein.

Speaker notes

Timing:

Aktuelle Slide: 00:02:00

Slide Ende: 00:04:30

Typescript (TS)

Features

Timing:

Aktuelle Slide: 00:01:00

Slide Ende: 00:05:30

Und warum das Ganze?

'foo' validiert zu 'NaN'. Es wird kein Fehler geworfen.

Speaker notes

Timing:

Aktuelle Slide: 00:00:30

Slide Ende: 00:06:00

Bild Statistik

Speaker notes

Timing:

Aktuelle Slide: 00:01:00

Slide Ende: 00:07:00

Typescript installieren

Timing:

Aktuelle Slide: 00:02:00

Slide Ende: 00:09:00

Typescript File transpilieren

```
tsc -t es2022 demo3.ts
```

Timing:

Aktuelle Slide: 00:02:00

Slide Ende: 00:11:00

Duck-Typing

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

Timing:

Aktuelle Slide: 00:01:30

Slide Ende: 00:12:30

Typen

any: Stellt quasi keine Typisierung dar (Kann alles sein), any kann man alles zuweisen

unknown: Stellt das "Gegenteil" zu any dar. Alles kann unknown zugewiesen werden, unknown kann aber nur unknown und any zugewiesen werden.

never: Bezeichnet einen Wert, der nicht vorkommen kann.

void: Bezeichnet einen Wert ohne Daten. Kann nur undefined (oder null bei strictNullChecks=false) annehmen.

Timing:

Aktuelle Slide: 00:01:30

Slide Ende: 00:14:00

Typen Demo

any: Stellt quasi keine Typisierung dar (Kann alles sein), any kann man alles zuweisen

unknown: Stellt das "Gegenteil" zu any dar. Alles kan unknown zugewiesen werden, unknown kann aber nur unknown und any zugewiesen werden.

never: Bezeichnet einen Wert, der nicht vorkommen kann.

void: Bezeichnet einen Wert ohne Daten. Kann nur undefined (oder null bei strictNullChecks=false) annehmen.

Timing:

Aktuelle Slide: 00:01:30

Slide Ende: 00:15:30

Type Inference

Alle Typdefinitionen können entfernt werden

Speaker notes

Timing:

Aktuelle Slide: 00:01:00

Slide Ende: 00:16:30

Object Types

Type Inference bei Methoden Return

Timing:

Aktuelle Slide: 00:01:30

Slide Ende: 00:18:00

Type Aliases

```
type StringPropertyGetter = (obj: any) => string;
```

Timing:

Aktuelle Slide: 00:01:30

Slide Ende: 00:19:30

Property Modifiers

```
1 const point : Point = {x: 5, y: 10, color:  
  "red", name: "P1" };  
2 point.z; // Returns: undefined  
3 point.name = "P2"; // Error: Cannot assign to  
  'name' because it is a read-only Property
```

Timing:

Aktuelle Slide: 00:02:00

Slide Ende: 00:21:30

Union Types

```
1  if (typeof foo === "number") {  
2      foo.toExponential();  
3  } else {  
4      foo.length;  
5  }
```

Timing:

Aktuelle Slide: 00:01:30

Slide Ende: 00:23:00

Type Predicates

```
1 function isSquare(shape: Shape): shape is  
  Square {  
2     return typeof ((shape as Square).a) ===  
    'number';  
3 }
```


Timing:

Aktuelle Slide: 00:01:00

Slide Ende: 00:24:00

Intersection Types

```
1  const t: ColorfulCircle = {  
2      color: "red",  
3      radius: 4  
4  }
```

Timing:

Aktuelle Slide: 00:01:30

Slide Ende: 00:25:30

String/Number Literal Types

Timing:

Aktuelle Slide: 00:01:00

Slide Ende: 00:26:30

Nullable Types

```
1 notNullableString = null;  
2 nullableString = null;
```

Timing:

Aktuelle Slide: 00:01:00

Slide Ende: 00:27:30

Hier sollen alle Basis Funktionalitäten von Generics mit einer Funktion gezeigt werden.

Speaker notes

Timing:

Aktuelle Slide: 00:00:45

Slide Ende: 00:28:15

Timing:

Aktuelle Slide: 00:03:00

Slide Ende: 00:31:15

Mapped types

```
type OptionalPerson = {  
    [P in keyof Person]?: Person[P];  
}  
  
type StringPerson = {  
    [P in keyof Person]: string;  
}
```

Timing:

Aktuelle Slide: 00:01:30

Slide Ende: 00:32:45

Vorhandene Mapped Types

Speaker notes

Timing:

Aktuelle Slide: 00:02:00

Slide Ende: 00:34:45

Conditional Types

Conditional Types - Type Inferring

```
1 type Flatten<T> = Type extends Array<infer I> ?  
  I : T;  
2 type ReturnType<T extends () => any> = T  
  extends () => infer R ? R : never;
```

Speaker notes

No notes on this slide.

Timing:

Aktuelle Slide: 00:01:00

Slide Ende: 00:39:30

- Browser unterstützen TS nicht

Problem: --> TS müsste erst zu JS transpiliert werden. Microsoft hat die Intension JS um Typen zu erweitern, damit TS gar nicht mehr notwendig ist.

TypeScript

Starke Typen für JavaScript

