Training TypeScript Module – the 'this' keyword

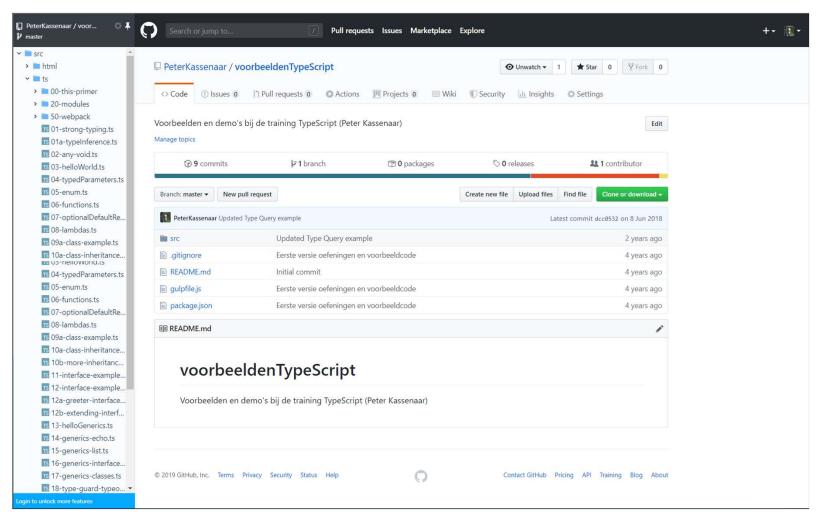




Peter Kassenaar

info@kassenaar.com

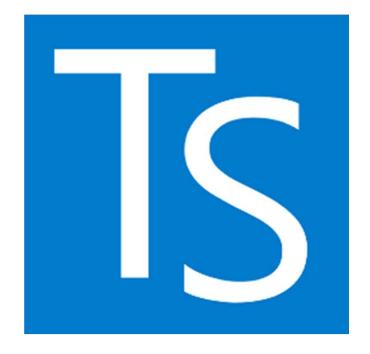
TypeScript Repo



https://github.com/PeterKassenaar/voorbeeldenTypeScript

Both ES6 and TypeScript know this keyword





But what exactly is this?

- One of the design flaws in JavaScript (20+ yrs ago)
- this has a different meaning, depending on the context
 - TypeScript flattens this behavior by providing a generic context
 - 'Lexical this'
 - Only inside classes!
- The question is: "who is calling this?
 - "the value of this is determined by how a function is called"

1. Invoking via function

```
//1. Via function
function myFunction() {
    console.log('Function :::', this);
}
myFunction();

// The window is invoking the function, so 'this'
points to Window object.
```

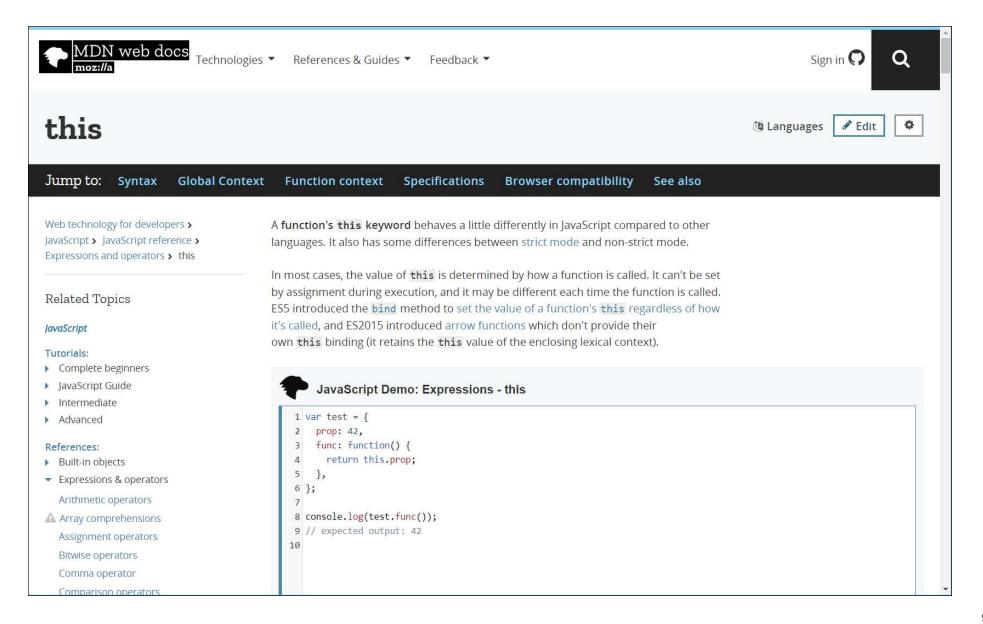
2. Via Object

```
// 2. Via Object
const myObj = {
    myMethod() {
        console.log('Object:::', this);
myObj.myMethod();
// The object myObj is invoking the function, so
'this' points to the object
```

3. Via TypeScript class

```
class MyClass {
    name: string = 'Peter';
    myMethod() {
        console.log('Class:::', this);
const myClass = new MyClass();
myClass.myMethod();
// The class is invoking the function, so 'this'
points to the class.
```

<u>https://developer.mozilla.org/en-</u> <u>US/docs/Web/JavaScript/Reference/Operators/this</u>



Workshop

- Create a TypeScript page with a plain old JavaScript function
- Compile it, run inside NodeJS or in HTML-page
- What does this point at?
- Create a function inside that function. For example a setTimeout()
 with a callback function.
- Do a console.log() inside the callback. What does this point at?
- Create a TypeScript class and copy the setTimeout function to the class. Where does this point at now?
- Optional: use the explicitely typed this inside a DOM-function.
 - See example 00-this-primer/00-this-primer.ts

