

# Decorators 進捗

---

TypeScript / ECMAScript 7

# 自己紹介

---

azu

---

@azu\_re

---

Web scratch, JSer.info



# TypeScript進捗

- [Decorators · Issue #2249 · Microsoft/TypeScript](#)
  - 仕様 Issue
  - 仕様 : [jonathandturner/brainstorming](#)
- [Minimal implementation of Decorators by rbuckton · Pull Request #2399 · Microsoft/TypeScript](#)
  - Origin: [Value decorators by rbuckton · Pull Request #2386 · Microsoft/TypeScript](#)

# ECMAScript 7(next?)

- [wycats/javascript-decorators](#)
- by Yehuda Katz
- [agendas/03.md at master · tc39/agendas](#)で提案予定

# ES7 Decorators

```
class Person {  
  @nonenumerable  
  get kidCount() { return this.children.length; }  
}
```

```
function nonenumerable(target, name, descriptor) {  
  descriptor.enumerable = false;  
  return descriptor;  
}
```

```
@nonenumerable  
get kidCount() { return this.children.length; }
```

=> 以下のように定義がnonenumerable関数でラップされる

```
descriptor = nonenumerable(Person.prototype, 'kidCount', descriptor);  
  
function nonenumerable(target, name, descriptor) {  
  descriptor.enumerable = false; // non emu  
  return descriptor;  
}
```

# ES7 Decorator is suger

— Decorator は ただの糖衣構文

```
@F("color")
class Foo {
}
// ===== //
var Foo = (function () {
  class Foo {
  }

  Foo = F("color")(Foo) || Foo;
  return Foo;
})();
```

## @ confusing zenparsing/es-private-fields?

```
class Point {  
  @x;  
  @y;  
  constructor(x = 0, y = 0) {  
    @x = +x;  
    @y = +y;  
  }  
  get x() { return @x }  
  set x(value) { @x = +value }  
  get y() { return @y }  
  set y(value) { @y = +value }  
  toString() { return `Point<${ this.@x },${ this.@y }>` }  
}
```



- [zenparsing/es-private-fields](#)
  - class内に@var = 42でprivate変数
  - coffeescriptの@に近い書き味(privateになるけど)
- [wycats/javascript-decorators](#)
  - @func class外/class内 どちらでも使える
- ;の有無しかという違いがある！！
- => [45th meeting of Ecma TC39](#) で話し合うとの事！

# TypeScriptとES7

- Decoratorsの糖衣構文的な仕組みは同じっぽい?
- C.3 Type Serialization: とか方に関する記述もあるので仕様はTypeScriptのヤツのほうが大きい

# Babel

- [Decorators · Issue #974 · babel/babel](#)
- それぞれの仕様提案してる人が集まるので観測に便利

