

Decorators

- Only works with class declarations
 - Can be applied to all members of a class
- Don not work with independent functions
- **"experimentalDecorators": true**
- **"emitDecoratorMetadata": true**
- decorators are functions
- starts with capital letter as industry convention
- number of arguments in decorator function depends on where it is going to be used
- In class, constructor in required argument in decorators
- Invoked only once, when class is defined during runtime and not when class is instantiated
- decorators are fired only once during lifecycle of application
- We can use decorators to add class prototype properties
- we can create class interface
- we get 3 arguments for **MethodDecorators**
 1. classPrototype: Object,
 2. methodName: string,
 3. descriptor: PropertyDescriptor,
- In static method decorators we get constructor, not class prototype
- Decorators Factories
- **ParameterDecorator**
 1. classPrototype:Object,
 2. methodName:string - name inside which parameter exists
 3. index:number - position of parameter in function
- **PropertyDecorator**
 1. classPrototype , is static class constructor
 2. name : string
- **AccessorDecorator**
 1. classPrototype : Object,
 2. accessorName: string,
 3. propertyDescriptor: PropertyDescriptor
- We can use decorators to modify class construction also
 - Decorator Factory + modifying class constructor values