

Where do carrots grow?

`Object.assign()` and classes


```
class Carrot {
  name = "carrot"
  grow() {
    return `${this.name}s grow in the ground`
  }
}
```

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()  
  
console.log(carrot.grow())
```

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()  
  
console.log(carrot.grow())
```

carrots grow in the ground


```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```



```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()  
  
console.log(carrot.grow())
```

```
class Apple {
  name = "apple"
  grow() {
    return `${this.name}s grow on trees`
  }
}
```

```
class Carrot {
  name = "carrot"
  grow() {
    return `${this.name}s grow in the ground`
  }
}
```

```
const apple = new Apple()
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow in the ground

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow in the ground

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```


```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow in the ground

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

enumerable



```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow in the ground

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

enumerable

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow in the ground

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```



```
class Apple {  
  name = "apple"  
  grow = () => {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow = () => {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

```
class Apple {  
  name = "apple"  
  grow = () => {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow = () => {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow on trees

```
class Apple {  
  name = "apple"  
  grow = () => {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow = () => {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

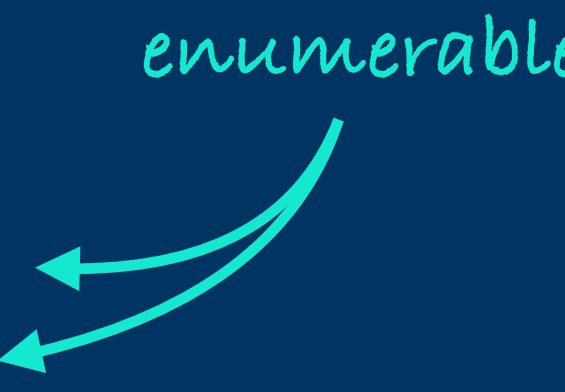
```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow on trees

```
class Apple {  
  name = "apple"  
  grow = () => {  
    return `${this.name}s grow on trees`  
  }  
}
```

enumerable



```
class Carrot {  
  name = "carrot"  
  grow = () => {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow on trees

```
class Apple {  
  name = "apple"  
  grow = () => {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow = () => {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow = () => {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow = () => {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow in the ground

```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow = () => {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow in the ground


```
class Apple {  
  name = "apple"  
  grow() {  
    return `${this.name}s grow on trees`  
  }  
}
```

enumerable

```
class Carrot {  
  name = "carrot"  
  grow = () => {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow in the ground

```
class Apple {
  name = "apple"
  grow = () => {
    return `${this.name}s grow on trees`
  }
}
```

```
class Carrot {
  name = "carrot"
  grow() {
    return `${this.name}s grow in the ground`
  }
}
```

```
const apple = new Apple()
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

```
class Apple {  
  name = "apple"  
  grow = () => {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow on trees

```
class Apple {  
  name = "apple"  
  grow = () => {  
    return `${this.name}s grow on trees`  
  }  
}
```

```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

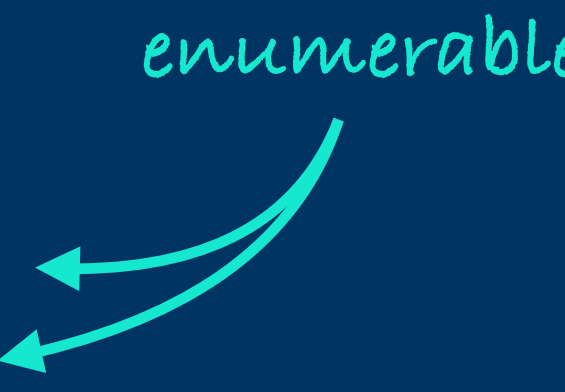
```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow on trees

```
class Apple {  
  name = "apple"  
  grow = () => {  
    return `${this.name}s grow on trees`  
  }  
}
```

enumerable



```
class Carrot {  
  name = "carrot"  
  grow() {  
    return `${this.name}s grow in the ground`  
  }  
}
```

```
const apple = new Apple()  
const carrot = new Carrot()
```

```
Object.assign(carrot, apple)
```

```
console.log(carrot.grow())
```

apples grow on trees

```
Apple GdocBase
Carrot GdocAuthor
grow() _enrichSubclassContent()
```

The target of the assign is irrelevant

- if fn is **enumerable**, it will always take over
- if fn is non enumerable, it will never be used*

*except in inheritance scenarios

When in doubt, use class methods

- override behaviour is less surprising
- better performance