

프론트엔드 개발자를 위한 Vue.js 시작하기

Vue.js </> 시작하기

ES2015 – OOP



한국기술교육대학교
온라인평생교육원

학습내용

- Class와 생성자
- Property

학습목표

- Class 선언과 생성자, 상속에 대해 설명할 수 있다.
- Property 선언과 setter, getter 이용 방법에 대해 설명할 수 있다.

Class와 생성자

Class와 생성자

1 Class 선언

- ES5에서는 함수로 클래스 정의함

```
var User = (function(){
  function User(name){
    this.name=name
  }
  User.prototype.sayHello = function(){
    console.log(`sayHello...${this.name}`)
  }
  return User
})();
```

Class와 생성자

1 Class 선언

- ES5에서는 함수로 클래스 정의함

```
var obj1=new User('kkang')
var obj2=new User('kim')
obj1.sayHello()//sayHello...kkang
obj2.sayHello()//sayHello...kim
```

Class와 생성자

Class와 생성자

1 Class 선언

- ES2015에서는 class 예약어로 클래스 선언함

```
class User {
  constructor(name){
    this.name=name
  }
  sayHello(){
    console.log(`sayHello...${this.name}`)
  }
}
```

Class와 생성자

1 Class 선언

- ES2015에서는 class 예약어로 클래스 선언함

```
var obj1=new User('kkang')
var obj2=new User('kim')
obj1.sayHello()//sayHello...kkang
obj2.sayHello()//sayHello...kim
```

Class와 생성자

Class와 생성자

2 생성자

생성자

- 객체 생성 시점에 호출되는 함수
- constructor 이름으로 선언되는 함수
- 하나의 클래스 내에 하나의 constructor 함수만 선언 가능함

Class와 생성자

2 생성자

- 생략 가능하며 생략하면 매개변수 없는 constructor가 자동 추가됨

```
class User{}  
  
class User1{  
  constructor({})  
}
```

Class와 생성자

Class와 생성자

2 생성자

- 생략 가능하며 생략하면 매개변수 없는 constructor가 자동 추가됨

```
class User2{
  constructor(name){}
}

var obj1=new User()
var obj2=new User1()
var obj3=new User2('kkang')
```

Class와 생성자

3 상속

- extends 예약어로 상속 관계 명시
- super 예약어로 상위 클래스의 생성자, 멤버 지칭

```
class Shape {
  constructor(id, x, y) {
  }
}
class Rectangle extends Shape {
  constructor(id, x, y, width, height) {
    super(id, x, y)
  }
}
var obj = new Rectangle(1, 0, 0, 100, 100)
```

Property

Property

1 Property

Property

- 클래스의 멤버 변수
- 클래스 선언영역에서 선언

Property

1 Property

- 클래스 내에서 멤버 변수 접근은 this 예약어를 이용함

```
class User {  
  name  
  address='seoul'  
  constructor(arg1, arg2){  
    this.name = arg1  
  }  
  sayHello(){  
    console.log(`${this.name}, ${this.address}`)//hello, seoul  
  }  
}
```

Property

Property

1 Property

- 클래스 내에서 멤버 변수 접근은 this 예약어를 이용함

```
var user = new User("hello", "a@a.com")
user.sayHello()
```

Property

1 Property

- 클래스 선언영역이 아닌 생성자 내부 혹은 함수 내부에서도 클래스 멤버 변수 선언이 가능함

```
class User {
  constructor(arg1, arg2){
    this.name = arg1
    this.email = arg2
  }
  someFun() {
    this.age = 30
  }
  sayHello(){
    console.log(`${this.name}, ${this.email}, ${this.age}`)
    //hello, a@a.com, 30
  }
}
```


Property

Property

1 Property

- 클래스 선언영역이 아닌 생성자 내부 혹은 함수 내부에서도 클래스 멤버 변수 선언이 가능함

```
var user = new User("hello", "a@a.com")
user.someFun()
user.sayHello()
```

Property

2 Getter/Setter

- Getter, Setter는 get과 set 예약어로 선언되는 함수
- 외부에서는 Property로 이용

```
class Product {
  constructor(arg){
    this._name = arg
  }
  get name() { return this._name }
  set name(arg) { this._name = arg }

  get price() { return this._price }
  set price(arg) { return this._price }
}
```

Property

Property

2 Getter/Setter

- Getter, Setter는 get과 set 예약어로 선언되는 함수
- 외부에서는 Property로 이용

```
var prod = new Product('book1')
prod.name = 'book2'
prod.price = 200

console.log(`${prod.name}, ${prod.price}`)
```

Property

2 Getter/Setter

get 예약어로
선언되는 함수

매개변수가 **없어야 함**

set 예약어로
선언되는 함수

매개변수가 **하나**만 선언되어야 함

Property

Property

2 Getter/Setter

- set, get 선언되는 함수를 외부에서 함수로 이용할 수는 없음

```
class Product {
  get price() { return this._price }
  set price(arg) { this._price = arg }

  get code(arg) { return this._code } //error
  set code(arg1, arg2) { this._code = arg1 } //error
}

var prod = new Product('book1')
prod.price('100') //error
prod.price = 200
```

Property

3 static 함수

static 함수

- static 예약어로 추가되는 함수
- 클래스 명으로 접근해야 함(객체로 접근 불가)

Property

Property

3 static 함수

- static 함수에서 Object member 접근하면 Undefined임

```
class User2 {
  data = 10
  static fun1() {
    console.log(`data : ${this.data}`)//undefined
  }
  fun2() {
    console.log(`data : ${this.data}`)
  }
}
```

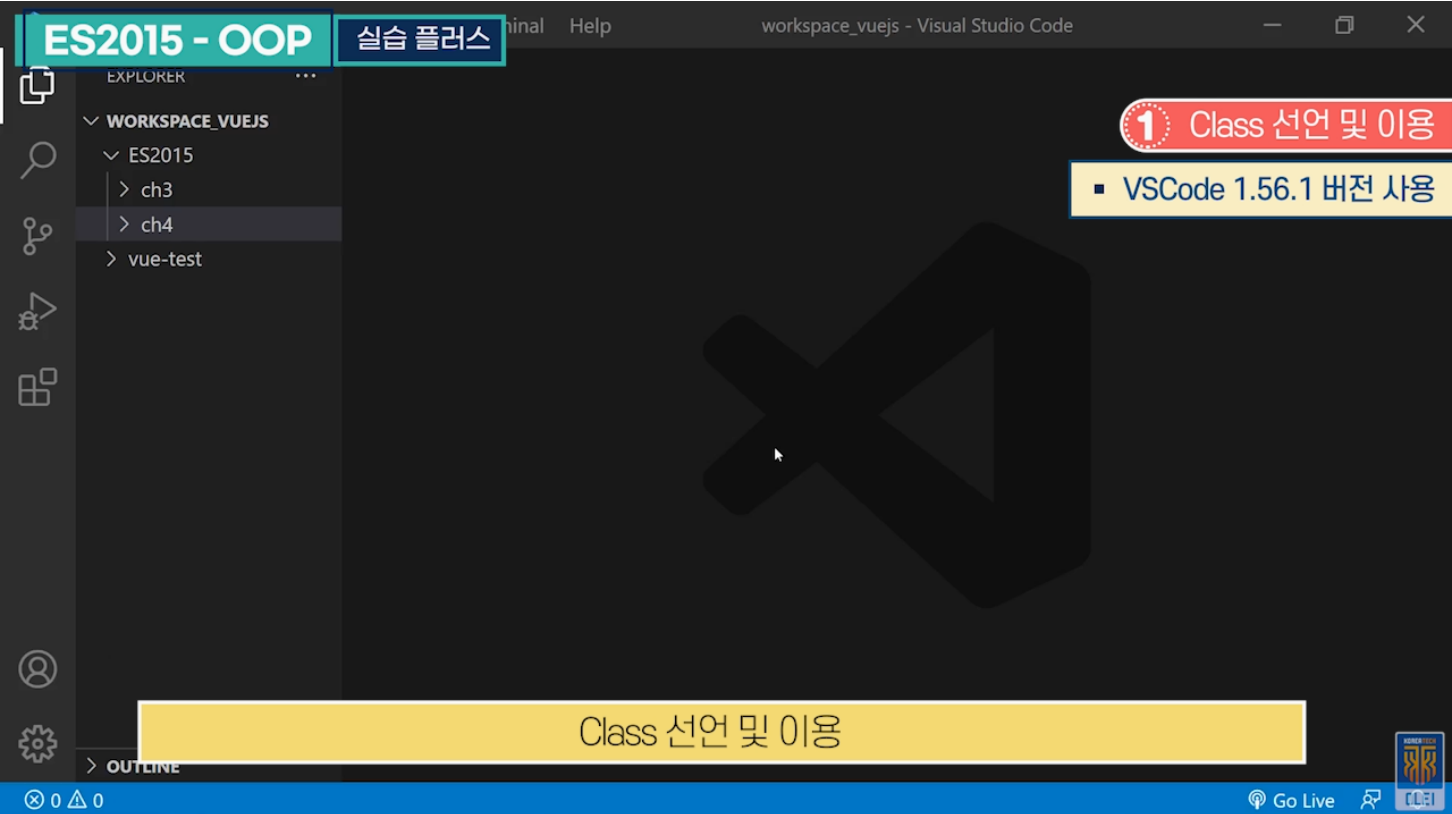
Property

3 static 함수

- static 함수에서 Object member 접근하면 Undefined임

```
User2.fun1()
// User2.fun2()//error
var user2 = new User2()
// user2.fun1()//error
user2.fun2()
```

실습 플러스



실습단계
ES2015 → ch5 서브 폴더 생성
서브 폴더 생성 후 자바스크립트 파일 생성
Customer Class 내에 멤버변수 name, address 선언
매개변수 두 개를 받는 constructor 선언
받은 매개변수로 멤버변수 초기화
멤버변수를 Class 영역에서 선언하지 않아도 됨
someFun 함수 선언
일반 함수에서도 멤버변수 선언 가능
console 창에 멤버변수를 출력하는 함수 선언
Getter/Setter 정의
객체 생성, 매개변수 전달
생성한 객체의 함수 호출
테스트용 index.html 파일 생성
Script Tag로 자바스크립트 파일을 지정함

실습 플러스

실습단계
Live 서버로 실행
console에서 데이터 확인 가능
Class 상속 테스트
ch5 서브 폴더에 inheritance.js 파일 생성 후 Class 선언
생성자를 만들고 멤버변수 초기화
함수 선언, console에 출력 정의
만들어진 Class를 상속받아 다른 Class 생성
상위 Class 생성자 호출
상위 Class 함수 오버라이드
Class 이용을 위해 객체 생성
방금 작성한 자바스크립트 파일을 추가함