

2025년 상반기 K-디지털 트레이닝

Vue.js 소개

[KB] IT's Your Life

- ✓ 개발 도구
- ✓ Node.js
- ✓ npm
- ✓ VS Code
- ✓ Vue 3 Snippets 확장팩

- ✓ VS Code의 Vue 3 Snippets 확장팩을 설치하세요.
- ✓ 크롬 Vue.js devtools 확장팩을 설치하세요.
- ✓ message 변수에 "Hello Vue3"를 지정하고, vue를 이용해 화면에 출력하세요.
콘솔에서 해당 메시지를 "첫 번째 Vue3 앱 입니다" 변경하여 화면에 출력하세요.

○ ex01.html

Hello Vue3!

첫 번째 Vue3 앱입니다

2025년 상반기 K-디지털 트레이닝

Vue.js를 위한 ES6

[KB] IT's Your Life

✓ 다음 코드의 실행 결과를 적으세요. 실제 결과를 확인하세요.

○ 02-02.js

```
let msg = "GLOBAL";
function outer() {
  let msg = "OUTER";
  console.log(msg);
  if (true) {
    let msg = "BLOCK";
    console.log(msg);
  }
}
outer()
```

✓ 다음 코드의 실행 결과를 적으세요. 실제 결과를 확인하세요.

○ 02-03.js

```
const p1 = { name: 'john', age: 20 };  
p1.age = 22;  
console.log(p1);  
  
p1 = { name: 'lee', age: 25 };
```

✓ 다음 출력결과가 나오도록 addContact() 함수를 완성하세요.

- 매개변수가 생략된 경우 기본값은 "없음"
- 출력할 문자열은 템플릿 문자열로 구성함

○ 02-05.js

```
function addContact( ) {
```

```
}
```

```
addContact("홍길동", "010-222-3331")
```

```
addContact("이몽룡", "010-222-3331", "02-3422-9900", "서울시");
```

```
name=홍길동, mobile=010-222-3331, home=없음, address=없음, email=없음
```

```
name=이몽룡, mobile=010-222-3331, home=02-3422-9900, address=서울시, email=없음
```

✓ 다음 결과가 나오도록 foodReport 함수를 작성하세요.

- 음식 목록은 가변 파라미터로 처리함
- 02-06.js

```
function foodReport( ) {  
  
}
```

```
foodReport("이몽룡", 20, "짜장면", "냉면", "불고기");  
foodReport("홍길동", 16, "초밥");
```

```
이몽룡, 20  
[ '짜장면', '냉면', '불고기' ]  
홍길동, 16  
[ '초밥' ]
```


✓ 배열의 구조 분해를 이용해 다음 변수들을 설정하세요.

○ a1, a2, a3, n, a, gender 변수

○ 02-07.js

```
let arr = [10,20,30,40];
```

```
console.log(a1, a2, a3);
```

```
let p1 = { name:"홍길동", age:20, gender:"M" };
```

```
console.log(n, a, gender);
```

```
10 20 30
```

```
홍길동 20 M
```

✓ 다음 출력이 되도록 각 함수의 매개변수를 선언하세요.

- addContact2() 함수의 빈칸을 채우세요.
- 02-08.js

```
function addContact1(          ) {  
    console.log(name,phone,email,age);  
}  
addContact1({ name : "이몽룡", phone : "010-3434-8989" })  
  
function addContact2(          ) {  
    if (!contact.email) contact.email = "이메일없음";  
    if (!contact.age) contact.age = 0;  
    let {_____} = contact;  
    console.log(name,phone,email,age);  
}  
addContact2({ name : "이몽룡", phone : "010-3434-8989" })  
  
function addContact3(name,phone,          ) {  
    console.log(name,phone,email,age);  
}  
addContact3("이몽룡","010-3434-8989")
```

```
이몽룡 010-3434-8989 이메일없음 0  
이몽룡 010-3434-8989 이메일없음 0  
이몽룡 010-3434-8989 이메일없음 0
```

✓ 다음 함수를 화살표 함수로 정의하여 호출하세요.

○ 02-09.js

```
const test1 = function(a,b) {  
  return a+b;  
}
```

```
const test2 =
```

```
const test3 =
```

```
console.log(test1(3,4));  
console.log(test2(3,4));  
console.log(test3(3,4));
```

```
7  
7  
7
```

✓ 다음 코드의 실행 결과를 적으세요. 실제 실행결과를 확인하세요.

○ 02-13.js

```
let obj = { result:0 };

obj.add = function(x,y) {
  function inner() {
    this.result = x+y;
  }
  inner();
}
obj.add(3,4)

console.log(obj)
console.log(result)
```

✓ ES6 객체 리터럴 정의로 obj 변수를 설정하세요.

○ 02-16.js

```
let name = "홍길동";  
let age = 20;  
let email = "gdhong@test.com";
```

```
let obj = _____;  
console.log(obj);
```

```
{ name: '홍길동', age: 20, email: 'gdhong@test.com' }
```

✓ 다음 코드에서 discount 메서드를 ES6 규칙으로 정의하세요.

○ 02-17.js

```
let p1 = {
  name : "아이패드",
  price : 200000,
  quantity : 2,
  order : function() { // 기존 방법
    if (!this.amount) {
      this.amount = this.quantity * this.price;
    }
    console.log("주문금액 : " + this.amount);
  },
  discount : { // ES6의 메서드 선언
    if (rate > 0 && rate < 0.8) {
      this.amount = (1-rate) * this.price * this.quantity;
    }
    console.log((100*rate) + "% 할인된 금액으로 구매합니다.");
  }
}
p1.discount(0.2);
p1.order();
```

20% 할인된 금액으로 구매합니다.
주문금액 : 320000

✓ 다음과 같이 출력되도록 아래 코드에서 출력할 문자열을 리터럴 템플릿으로 정의하세요.

○ 02-18.js

```
const d1 = new Date();  
let name = "홍길동";  
let r1 = _____;  
console.log(r1);  
  
let product = "갤럭시S7";  
let price = 199000;  
let str = _____;  
console.log(str);
```

홍길동 님에게 Mon Jan 15 2024에 연락했다.
갤럭시S7의 가격은
199000원 입니다.

✓ 다음과 같은 모듈을 정의하였다. 모듈을 임포트하여 사용해 보세요.

○ modules/02-19-module.js

```
let base = 100;  
const add = (x) => base+x;  
const multiply = (x) => base*x;
```

```
export { add, multiply };
```

○ 02-20-main.js

```
_____ // 02-19-module 임포트  
  
console.log(add(4));
```


✓ 다음과 같은 모듈을 정의하였다. 모듈을 임포트하여 사용해 보세요.

○ modules/02-19-module.js

```
let base = 100;  
const add = (x) => base+x;  
const multiply = (x) => base*x;
```

```
export { add, multiply };
```

○ add와 multiply 각각을 직접 export하여 위와 동일한 모듈이 되도록 작성하세요

✓ 다음 모듈을 정의하고 사용하는 쪽에서 임포트 하는 코드를 완성하세요.

○ modules/02-19-module.js

```
let base = 100;  
const add = (x) => base+x;  
const multiply = (x) => base*x;  
const getBase = ()=>base;  
  
export { add, multiply };  
export default getBase;
```

○ 02-20-main.js

```
_____  
;  
  
console.log(add(4));  
console.log(getBase());
```

✓ 다음과 같은 모듈을 정의하였다. 모듈을 임포트하여 사용해 보세요.

○ modules/02-20-module.js

```
let base = 100;  
const add = (x) => base+x;  
const multiply = (x) => base*x;  
const getBase = ()=>base;  
  
export { add, multiply };  
export default getBase;
```

○ 위 모듈에서 add, multiply 각각을 직접 expor하고, getBase는 디폴트로 바로 export 하도록 다시 정의해 보세요.