스타트업 개발자와 함께 공부하는 Node.js

01. 자바스크립트

강의 내용은 강사가 별도로 명시하지 않는 한 비공개로 간주합니다. 녹음이나 사진 촬영를 허락하지 않으며 콘텐츠를 블로그, SNS 등에 게시하거나 공개적으로 공유하지 마세요.

콘텐츠 공유 가능 여부에 대해 궁금한 점이 있는 경우 강사에게 문의하시기 바랍니다.







# 목차

- 1. 자바스크립트 기본
- 2. 조건문
- 3. 반복문
- 4. 함수
- 5. 객체
- 6. 예외처리



Tyle='color:orange;'>HTML fant man in the color:orange;'>HTML fant

- persisted propertie famely

SerrorMessage = text - 2000 % and a seriorMessage = text - 2000 % and a

- 200pm; > < todolistid = data location to the control of the cont

### □ 출력

```
1 console.log("Hello world");
2
3 console.log('hello world');
4
5 console.log(`hello ${"world"}`);
6
7
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_01.js]

#### □ 기본 자료형

```
1 //숫자 사칙연산
2 console.log(23);
3 \text{ console.log}(23 + 45);
4 console.log(23 * 45);
5 console.log(2 + 3 * 5);
 6 console.log(10 % 5);
7 console.log(10 / 5);
8 //문자자료형 및 이스케이프
9 console.log('안녕하세요')
10 console.log('this is "string"')
11 console.log("this is \"string\" ");
12 console.log("this is \n string");
13 console.log("this is \t string");
14 console.log("this is \\ string");
15 //문자열 더하기
16 console.log("hello" + " " + "javascript")
17 console.log("hello"[1]);
18 //템플릿 문자열
19 console.log(`52 + 45 = $\{52 + 45\}`);
20 //비교 연산자
21 console.log( 52 > 52 );
22 console.log( 45 == 52 );
23 console.log( 45 != 52 );
24 console.log( 52 >= 52 );
25 //논리 연산자
26 console.log( 52 >= 52 \&\& 52 > 52 );
27 console.log( 52 >= 52 || 52 > 52 );
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_02.js]

#### □ 변수

```
1 let pi;
 2 console.loog(pi);
 3 pi = 3.141592
   console.loog(pi);
 5
   let pi2 = 3.141592
   console.log(pi2);
 8
   let radius = 12;
10
   console.log(`넓이 : ${pi * radius * radius}`);
   console.log(`둘레 : ${pi * 2 * radius}`);
13
```

□ [C]-[nodejs]-[project]-[01]-[ch01\_03.js]

#### □ 복합 대입 연산자

```
1 \text{ radius } += 2
 2 console.log(radius);
 3
 4 let output = "hello";
 5 output += "world";
 6 console.log(output);
 8 let num = 0;
 9 num++;
10 console.log(`num++ : ${num}`);
11 num--;
   console.log(`num-- : ${num}`);
13
14
   console.log(`num++ : ${num++}`);
   console.log(`num-- : ${num--}`);
17
18 console.log(`++num : ${++num}`);
19 console.log(`--num : ${--num}`);
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_04.js]

#### □ 자료형 검사

```
1 console.log( typeof(10) );
   console.log( typeof("hello") );
 4
   console.log( typeof(true) );
 6
 7 console.log( typeof(function() {} ) );
 8
   console.log( typeof( {} ) );
10
  let beta
   console.log(typeof(beta));
13
```

□ [C]-[nodejs]-[project]-[01]-[ch01\_05.js]

#### □ 자료형 변환

```
1 console.log(String(52));
 2 console.log(typeof(52));
 3 console.log(typeof(String(52)));
   console.log(Number("45"));
   console.log(Number(true));
   console.log(Number("hello"));
 8
   console.log(isNaN(Number("hello")));
10
   console.log(Boolean(0));
   console.log(Boolean(NaN));
   console.log(Boolean(1));
   console.log(Boolean("Hello"));
15
   console log(25 + 125);
   console.log("25" + 125);
   console.log("25" - 125);
```

□ [C]-[nodejs]-[project]-[01]-[ch01\_06.js]

#### □ 일치 연산자

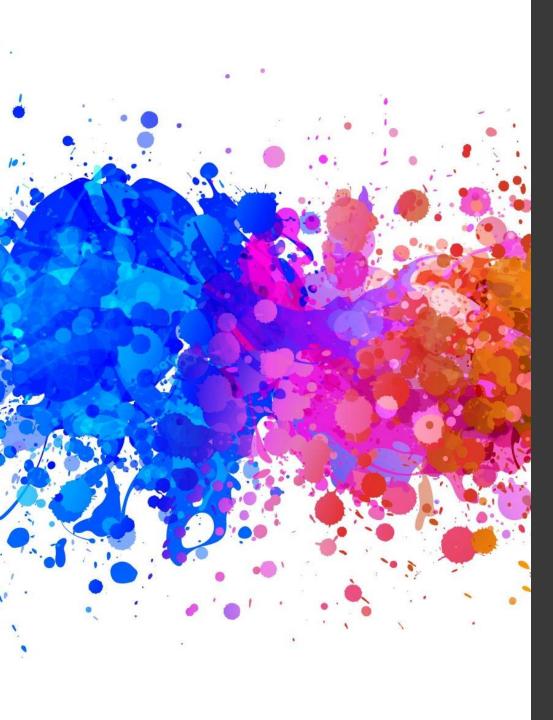
```
1 console.log(`50 == "50" : ${50 == "50" }`);
2 console.log(`50 === "50" : ${50 === "50" }`);
3
4 console.log(`0 == "" : ${0 == ""}`);
5 console.log(`0 === "" : ${0 === ""}`);
6
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_07.js]

### □ 상수

```
1 const test = "변경 불가";
2 console.log(`before : ${test}`);
3 test = "hello"
4 console.log(`after : ${test}`);
5
6 const abc = "abc"
7 // const def;
8
```

□ [C]-[nodejs]-[project]-[01]-[ch01\_08.js]



Var self = this <html> data = dta | 1 <html
 data = dta | 1 <html> data = dta | 1 <html
 data = dta | 1 <html

#### □ 기본 조건문

☐ [C]-[nodejs]-[project]-[01]-[ch01\_09.js]

### □ 중첩 조건문

```
1 // 중첩 조건문
 2 let date = new Date();
 3 let hours = date.getHours();
  if(hours < 11) {</pre>
       console.log(`아침 먹을 시간 입니다`);
 7 }else {
       if(hours < 15) {
           console.log('점심 먹을 시간 입니다');
       }else{
10
           console.log(`저녁 먹을 시간 입니다`);
11
12
13 }
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_10.js]

#### □ switch 문

```
1 let input = 12;
   switch(input % 2) {
       case 0:
 4
           console.log(`even`);
           break;
       case 1:
           console.log(`odd`);
           break;
 9
10
11
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_11.js]

### □ 삼항 연산자

```
1 let test
2
3 test = typeof(test) != 'undefined' ? test: 'initial';
4 console.log(test);
5
6
7
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_12.js]



```
반복문
```

### □ 배열

```
1 // array
2 let arr = [5, 23, 'hello', true, 'world', -9]
3 console.log(arr)
4 console.log(arr[2]);
5
6
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_13.js]

#### ■ while

☐ [C]-[nodejs]-[project]-[01]-[ch01\_14.js]

#### ☐ for

☐ [C]-[nodejs]-[project]-[01]-[ch01\_15.js]

#### ☐ for in

```
1 // for in
2 let arr = [5, 23, 'hello', true, 'world', -9]
3 for(i in arr){
4 | console.log(` ${i} is ${arr[i]}`);
5 }
6
7
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_16.js]

#### for of

```
1 // for of
2 let arr = [5, 23, 'hello', true, 'world', -9]
3 for(e of arr){
4 | console.log(`${e}`);
5 }
6
7
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_17.js]

### □ 중첩 반복문

```
1 let output = '';
   for(let i=0;i<10;i++){</pre>
       for(let j=0;j< 10 - i;j++){
           output += ' ';
       for(let j=0;j< i + 1; j++){
           output += '*';
10
       output += '\n';
11
12
   console.log(output);
14
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_18.js]

#### □ break

```
1 let arr = [5, 23, 'hello', true, 'world', -9]
2
3 for(i in arr) {
4          if(typeof(arr[i]) == 'string'){
5               break;
6          }
7          console.log(arr[i]);
8 }
9
10
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_19.js]

#### continue

```
1 let arr = [5, 23, 'hello', true, 'world', -9]
2
3 for(i in arr) {
4         if(typeof(arr[i]) == 'string'){
5              console.log(arr[i]);
6               continue;
7         }
8 }
9
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_20.js]



#### □ 익명함수, 선언함수, 화살표함수

```
1 // 익명함수
 2 let add1 = function(x, y) {
       return x + y;
 6 // 선언 함수
 7 function add2(x, y) {
       return x + y;
 8
 9
10
11 // 화살표 함수
12 let add3 = (x, y) \Rightarrow x + y;
13
14 console.log(add1(1, 4));
15 console.log(add2(1, 4));
16 console.log(add3(1, 4));
17
```

□ [C]-[nodejs]-[project]-[01]-[ch01\_21.js]

### □ 콜백함수

☐ [C]-[nodejs]-[project]-[01]-[ch01\_22.js]

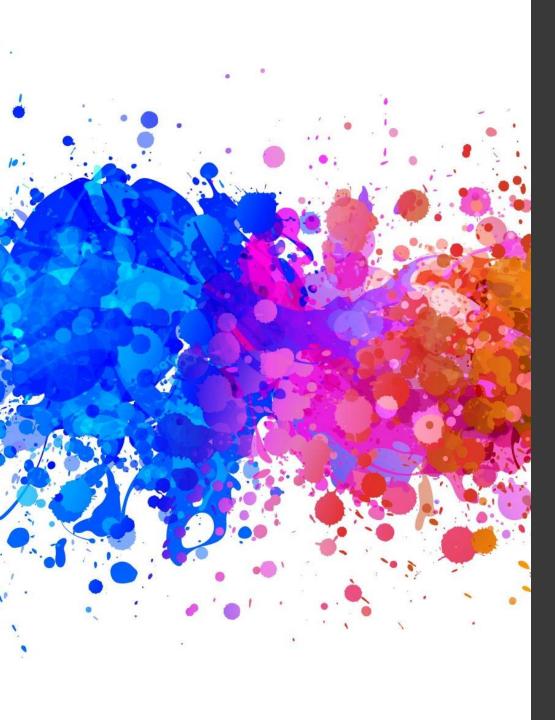
#### □ 콜백함수

```
1 // parseInt
2 console.log(`parseInt("52") => ${parseInt("52")}`);
3 console.log(`parseInt("3.14") => ${parseInt("3.14")}`);
4 console.log(`parseInt("1209\text{8"}) => ${parseInt("1209\text{8"})}`);
5 // parseFloat
6 console.log(`parseFloat("51.128") => ${parseFloat("51.128")}`);
7 console.log(`parseFloat("3.14") => ${parseFloat("3.14")}`);
8 console.log(`parseFloat("1209.2\text{8"}) => ${parseFloat("1209.2\text{8"})}`);
9
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_23.js]

### □ 시간 함수

☐ [C]-[nodejs]-[project]-[01]-[ch01\_24.js]



# 객체

#### □ 객체

```
1 // 객체
 2 let personInfo = {
       'name': 'lee',
      'age': 15,
     'address': 'seoul',
       'hobby': ['fishing','rc','star']
   console.log(personInfo['name']);
   console.log(personInfo['address']);
   console.log(personInfo['hobby']);
   console.log(personInfo.age);
13
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_25.js]

### □ 객체와 for문

☐ [C]-[nodejs]-[project]-[01]-[ch01\_26.js]

#### □ 객체와 메서드

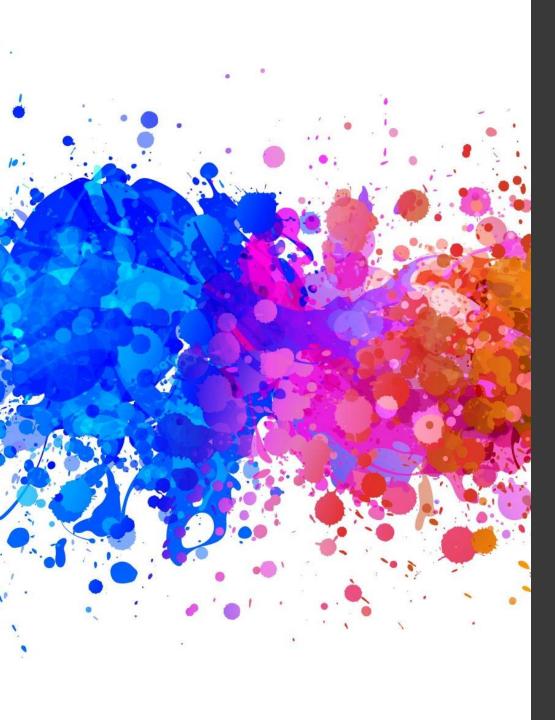
```
1 // 객체와 메서드
2 let personInfo = {
       name: 'lee',
       age: 15,
       address: 'seoul',
       hobby: ['fishing','rc','star'],
       addAge: function() {
           this.age = this.age + 1;
10
11
12 console.log(`before age : ${personInfo.age}`);
  personInfo.addAge();
14 console.log(`after age : ${personInfo.age}`);
15
16
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_27.js]

#### □ 클래스

```
1 // 클래스
 2 class PersonInfo {
 3
       constructor(name, age, address, hobby) {
 4
           this.name = name;
           this.age = age;
           this.address = address;
 6
 7
           this.hobby = hobby
 8
 9
       addAge(age) {
10
11
           this.age = this.age + age;
12
13
14
       getAge() {
15
           return this.age;
16
17 }
18
19 let p = new PersonInfo('lee', 12, 'seoul', ['rc', 'star']);
20 console.log(`before age : ${p.getAge()}`);
21 p.addAge(10);
22 console.log(`after age : ${p.getAge()}`);
23
```

☐ [C]-[nodejs]-[project]-[01]-[ch01\_28.js]



# 예외처리

### □ 예외처리

☐ [C]-[nodejs]-[project]-[01]-[ch01\_29.js]

### □ 예외처리

```
1 try{
       const err = new Error('This is Error');
       err.name = 'My Error';
       err.message = 'My Error Message'
       throw err;
 7 }catch(e) {
       console.log(`exception occured =>
           name:${e.name}, message: ${e.message}`);
10 }finally {
       console.log(`finally called`);
11
12 }
13
14
```

□ [C]-[nodejs]-[project]-[01]-[ch01\_30.js]

