

01 자바스크립트 기본 용어

- 표현식, 문장, 프로그램 : 값을 만들어 내는 간단한 코드를 표현식이라고 하며, 표현식이 모여 문장, 문장이 모여 프로그램이 됩니다.
- 키워드 : 자바스크립트를 처음 만들 때 정한 특별한 의미가 있는 단어입니다.
- 식별자 : 이름을 붙일 때 사용하는 단어입니다.
- 주석 : 프로그램의 진행에 전혀 영향을 주지 않고 프로그램 코드를 설명하는 목적으로 사용하는 코드입니다.

02 기본적인 출력을 할 때는 console.log() 메소드를 사용합니다.

03 기본 자료형 : 자바스크립트에는 다음의 기본 자료형이 있습니다.

- 숫자 : 52, 273, 52.273과 같은 숫자를 나타냅니다. 기본적인 사칙 연산자와 나머지 연산자를 사용할 수 있습니다.
- 문자열 : 문자로 구성된 것을 의미합니다. 문자열 연결 연산자를 사용할 수 있습니다.
- 불 : 참(true) 또는 거짓(false)을 나타냅니다. 불을 생성할 때는 비교 연산자를 사용하며, 불끼리는 논리 연산자를 사용할 수 있습니다.

04 변수와 상수

- 변수 : 값을 저장할 목적으로 사용하며, 이후에 변경할 수 있는 값입니다. 변수를 선언할 때는 let 키워드를 사용합니다.

```
let 식별자 = 자료
```

- 상수 : 값을 저장할 목적으로 사용하며, 이후에 변경할 수 없는 값입니다. 상수를 선언할 때는 const 키워드를 사용합니다.

```
const 식별자 = 자료
```

05 자료형 변환

- 숫자로 변환할 때는 `Number()` 함수를 사용합니다. 숫자로 변환할 수 없는 값을 강제로 숫자로 변환하면 `NaN`(Not a Number)이 됩니다.
- 문자열로 변환할 때는 `String()` 함수를 사용합니다.
- 불로 변환할 때는 `Boolean()` 함수를 사용합니다.
- 불로 변환할 때 `false`로 변환되는 것은 `0`, `NaN`, `""`(빈 문자열), `null`, `undefined`입니다.

01 다음 괄호 안을 채우시오.

- (1) 값을 만들어 내는 간단한 코드를 ()이라고 한다.
- (2) 표현식이 하나 이상 모여서 실행할 수 있는 코드가 되면, 이를 ()이라고 한다.
- (3) 특별한 의미를 갖는 단어로 자바스크립트가 처음 만들어질 때 정해진 단어를 ()라고 부른다.
- (4) 이름을 붙일 때 사용하는 단어를 ()라고 한다.
- (5) 참(true)과 거짓(false)을 갖는 값을 ()이라고 한다.
- (6) `true || false`의 결과는 ()이다.
- (7) `true && false`의 결과는 ()이다.
- (8) `30 > 20 > 10`의 결과는 ()이다.
- (9) 값을 저장할 때 사용하는 식별자로, 생성 이후에 값을 변경할 수 있는 것은 ()라고 한다.
- (10) 값을 저장할 때 사용하는 식별자로, 생성 이후에 값을 변경할 수 없는 것은 ()라고 한다.
- (11) 상수를 만들 때는 () 키워드를 사용한다.
- (12) 변수를 만들 때는 () 키워드를 사용한다.
- (13) 자료형을 검사할 때는 () 키워드를 사용한다.
- (14) 자바스크립트에는 '숫자 자료형이지만 숫자가 아닌 것'을 의미하는 ()이라는 개념이 있다.

02 다음 질문에 ○×를 표시하시오.

- (1) 자바스크립트는 문장 마지막에 문장을 종결한다는 의미로 반드시 세미콜론(;)을 찍어야 한다.
- (2) 자바스크립트가 처음 만들어질 때 정해진 단어였던 키워드는 자바스크립트가 발전하면서 조금씩 추가된다.
- (3) 식별자를 만들 때는 특수문자로 _와 \$만 허용한다.
- (4) 식별자는 공백을 포함할 수 있다.
- (5) 식별자는 숫자로 시작하면 안 된다.
- (6) 주석은 프로그램의 실행에 영향을 준다.
- (7) 한 줄 주석을 만들 때는 \\\ 기호를 사용한다.
- (8) 템플릿 문자열을 만들 때는 `` 기호와 \${} 기호를 사용한다.
- (9) 자바스크립트는 변수에 값을 넣어 자료형이 한번 결정되면, 이를 이후에 변경할 수 없다.
- (10) $10 + '10'$ 의 결과는 1010이다.
- (11) $10 * '10'$ 의 결과는 NaN이다.

03 다음 중 키워드가 아닌 것은?

- | | |
|---------|-----------|
| ① for | ② if |
| ③ const | ④ console |

04 다음 중 false로 변환되는 것은?

- | | |
|-------|-----------|
| ① '0' | ② 'false' |
| ③ '' | ④ 20 |

05

- ① `typeof(value)`

② alert(value)

③ console.log(value)

④ Number(value)

06

- ① \t

② \n

③ \'

④ //

07

```
console.log(52 + 273);
```

```
console.log("52" + 273);
```

```
console.log(52 + "273");
```

```
console.log("52" + "273");
```

08

```
console.log(10 + 20 + '!!!!');
```

09 다음 코드의 실행 결과를 적으시오.

```
const number = 10;
```

```
console.log(number++);
```

```
console.log(++number);
```

```
console.log(number++);
```

```
console.log(++number);
```

01 if 조건문, if else 조건문, if else if 조건문의 기본 형태

• if 조건문 기본 형태

```
if (불_표현식) {  
  
}
```

• if else 조건문 기본 형태

```
if (불_표현식) {  
    // 불_표현식이 참일 때 실행할 문장  
} else {  
    // 불_표현식이 거짓일 때 실행할 문장  
}
```

• if else if 조건문 기본 형태

```
if (불_표현식) {  
  
} else if (불_표현식) {  
  
} else if (불_표현식) {  
  
} else {  
  
}
```

02 switch 조건문의 기본 형태

```
switch (비교할_값) {  
    case 값:  
        문장  
        break;  
    case 값:  
        문장  
        break;  
    default:  
        문장  
        break;  
}
```

03 삼항 연산자의 기본 형태

불_표현식 ? 참 : 거짓

04 짧은 초기화 조건문의 기본 형태

변수 = 대상_변수 || 대상_변수가_undefined일_때_넣을_값

01 다음 괄호 안을 채우시오.

- (1) 오전을 `hours < 12`로 구분한다면, 오후는 ()로 구분할 수 있다.
- (2) 변수 `number`에 숫자가 들어 있다고 할 때, 이 값이 홀수인지 확인하려면 ()을 사용한다(다양한 방법이 있겠지만, 나머지 연산자를 활용하는 코드를 넣어 주세요).
- (3) 변수 `number`에 숫자가 들어 있다고 할 때, 이 값이 짝수인지 확인하려면 ()을 사용한다(다양한 방법이 있겠지만, 나머지 연산자를 활용하는 코드를 넣어 주세요).
- (4) 조건 표현식은 '불_표현식 () 참일_때_실행하고_밖으로_낼_값 () 거짓일_때_실행하고_밖으로_낼_값'이라는 형태의 문법을 사용한다.

02 다음 질문에 ○×를 표시하시오.

- (1) if 조건문은 뒤에 배치하는 불 표현식의 값이 true일 때 문장을 실행한다.
- (2) 현재 시간을 구할 때는 Date라는 객체를 사용한다.
- (3) if else 조건문은 오전과 오후처럼 두 가지로 나뉘는 상황에 유용하다.
- (4) 조건문은 중첩해서 사용할 수 없다.
- (5) switch 조건문은 반드시 break 키워드를 함께 사용해야 한다.

03 다음 중 조건문과 관련된 키워드가 아닌 것은?

- | | |
|----------|---------|
| ① if | ② else |
| ③ switch | ④ const |

04 학점을 나타내는 변수를 score라고 할 때, 학점 4.2~4.5(4.2와 4.5 포함)를 나타낼 수 있는 조건 표현식을 고르시오.

- | | |
|---|---|
| ① <code>4.2 <= score <= 4.5</code> | ② <code>4.2 <= score score <= 4.5</code> |
| ③ <code>4.2 <= score && score <= 4.5</code> | ④ <code>4.2 < score score < 4.5</code> |

05 자바스크립트에서 Node.js 입력을 받을 때 사용하는 모듈을 고르시오.

- (1) input (2) REPL
 (3) message (4) RIPL

06 다음 중 switch 조건문의 기본 형태와 직접적으로 연관 없는 키워드를 고르시오.

- ① case ② break
③ default ④ continue

07 다음 코드의 주석 부분에서 변수 x와 y에 표와 같은 값을 할당할 때 나오는 출력 결과를 쓰시오.

```
let x, y;
// 값_할당_부분

if (x > 4) {
    if (y > 2) {
        console.log(x * y);
    }
} else {
    console.log();
}
```

입력 값	실행 결과
$x = 2, \quad y = 10$	
$x = 1, \quad y = 4$	
$x = 10, \quad y = 2$	

08 다음 중첩 조건문을 논리 연산자를 사용해 하나의 if 조건문으로 작성하시오.

```
if (x > 10) {  
    if (x < 20) {  
        console.log("조건에 맞습니다.");  
    }  
}
```

09 다음 if 조건문으로 작성된 프로그램을 switch 조건문으로 옮기시오.

```
// 학년 변수를 생성합니다.  
let level = 1;  
  
// 출력합니다.  
if (level == 1) {  
    console.log("수강해야 하는 전공 학점: 12학점");  
} else if (level == 2) {  
    console.log("수강해야 하는 전공 학점: 18학점");  
} else if (level == 3) {  
    console.log("수강해야 하는 전공 학점: 10학점");  
} else if (level == 4) {  
    console.log("수강해야 하는 전공 학점: 18학점");  
}
```

01 배열

- 배열을 선언하는 기본 방법

```
let 이름 = [자료, 자료, 자료, 자료 자료]
```

- 배열의 요소에 접근하는 방법

```
배열[인덱스]
```

02 while 반복문의 기본 형태

```
while (불_표현식) {
    // 불_표현식이 참인 동안 실행할 문장
}
```

03 for 반복문의 기본 형태

```
for (let i = 0; i < 반복 횟수; i++) {
}
```

04 for in 반복문과 for of 반복문의 기본 형태

```
for(let 인덱스 in 배열) {
}

for(let 요소 of 배열) {
}
```

05 break 키워드와 continue 키워드를 사용하면 반복문의 코드 실행을 변경할 수 있습니다.

01 다음 괄호 안을 채우시오.

- (1) 자바스크립트에서 여러 개의 자료를 한꺼번에 다루고 싶을 때는 () 자료형을 사용한다.
- (2) 배열 안에 있는 각각의 자료를 ()라고 부른다.
- (3) 배열의 범위를 넘는 요소에 접근하면 ()자료가 나온다.
- (4) 무한하게 반복하는 반복문을 ()라고 부른다.
- (5) 반복문을 강제로 벗어날 때는 () 키워드를 사용한다.
- (6) 반복문에서 다음 단계로 넘어갈 때는 () 키워드를 사용한다.
- (7) ()는 '변수를 사용할 수 있는 범위'를 나타낸다.
- (8) 과거의 자바스크립트에서는 변수를 만들 때 일반적으로 () 키워드를 사용했다.
- (9) 스코프 내부에서 이름이 같아서, 상위 스코프에 있는 변수를 가리키는 것을 ()이라고 부른다.

02 다음 질문에 O×를 표시하시오.

- (1) 배열을 만들 때는 [] 기호를 사용한다.
- (2) 배열은 요소로 숫자, 문자열, 불 등 한 가지 자료형만 가질 수 있다.
- (3) 배열을 만들 때는 어떤 자료형의 배열인지를 따로 지정해야 한다.
- (4) 배열의 범위를 넘는 요소에 접근하면, index out of range 오류가 발생한다.
- (5) 배열의 인덱스는 1부터 시작한다.
- (6) while (true) { } 형태의 반복문은 무한하게 반복한다.
- (7) while 반복문을 벗어나려면, 반복문의 불 표현식을 false로 만들 수 있는 조건이 필요하다.
- (8) for in 반복문은 반복 변수에 요소가 들어간다.
- (9) for of 반복문은 반복 변수에 인덱스가 들어간다.
- (10) 반복문은 중첩해서 사용할 수 없다.

03 배열 안에 있는 요소 수를 확인할 때 사용하는 식별자를 고르시오.

- ① size ② length ③ capacity ④ count

04 다음 중 반복문과 관련되지 않은 키워드를 고르시오.

- ① for ② while ③ break ④ continue

05 다음 중 실행 때 오류가 발생할 수 있는 반복문의 기본 형태를 고르시오(실제로 코드를 입력하고 실행해 보는 것을 추천합니다).

- ① while (조건_표현식) {}
② for (const 요소 of 배열) {}
③ for (const 인덱스 in 배열) {}
④ for (const i = 0; i < 반복_횟수; i++) {}

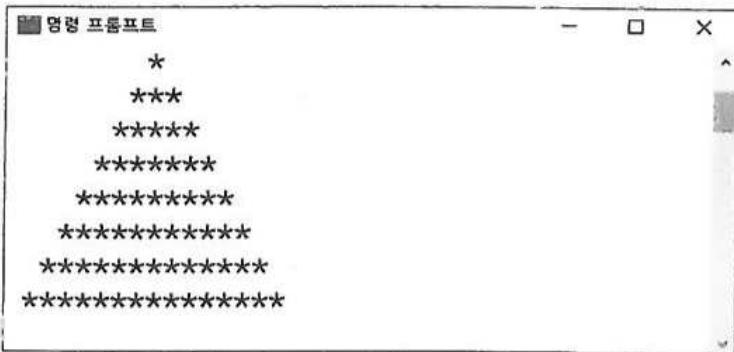
06 다음 코드의 for 반복문을 while 반복문으로 바꿔 보시오.

```
for (let i = 0; i < 10; i++) {  
    console.log("출력");  
}
```

07 다음 코드의 실행 결과를 예측해 보시오.

```
for (let i = 0; i < 20; i = i + 2) {  
  console.log("출력");  
}
```

08 다음 출력을 생성하는 코드를 반복문으로 만들어 보시오.



09 다음 for of 반복문을 결과가 같은 for in 반복문으로 변경하시오.

```
// 배열을 선언합니다.
let array = [52, 273, 32, 93, 103];

// for of 반복문을 적용합니다.
for (let element of array) {
    console.log(element);
}
```

10 다음 코드에서 빈 부분을 구성해 배열의 가장 작은 수와 큰 수를 출력하는 프로그램을 작성하시오.

```
// 배열을 선언합니다.  
let array = [273, 52, 103, 57, 271];  
  
// 최솟값과 최댓값을 계산합니다.  
let min = Number.MAX_VALUE;  
let max = Number.MIN_VALUE;  
  
// 구성합니다.  
  
// 출력합니다.  
console.log(`가장 큰 수: ${max}`);  
console.log(`가장 작은 수: ${min}`);
```

11 배열 [52, 71, 32, 103, 273, 93]을 반복문을 사용하여 역순으로 출력하는 코드를 작성하시오.

01 익명 함수는 다음 방법으로 선언합니다.

```
function (매개_변수, 매개_변수, ...) {
    // 함수_코드
    return 리턴_값;
}
```

02 선언적 함수는 다음 방법으로 선언합니다.

```
function 함수_이름(매개_변수, 매개_변수, ...) {
    // 함수_코드
    return 리턴_값;
}
```

03 화살표 함수는 다음 방법으로 선언합니다.

```
(매개_변수, 매개_변수, ...) => {
    // 함수 코드
    return 리턴_값;
};
```

04 매개 변수로 전달된 함수를 '콜백 함수'라고 합니다.

05 자바스크립트는 다음 내장 함수를 표준으로 제공합니다.

함수	설명
parseInt()	문자열을 정수로 변환합니다.
parseFloat()	문자열을 실수로 변환합니다.
setTimeout(함수, 시간)	특정 시간 후에 함수를 실행합니다.
setInterval(함수, 시간)	특정 시간마다 함수를 실행합니다.
clearInterval(아이디)	특정 시간마다 실행하던 함수 호출을 정지합니다.

01 다음 괄호 안을 채우시오.

- (1) 이름이 없는 함수를 ()라고 부른다.
- (2) 문자 그대로의 자료를 나타내는 것을 ()이라고 부른다.
- (3) 함수의 매개 변수로 전달하는 함수를 ()라고 부른다.
- (4) 함수 외부에서 함수 내부에 자료를 전달할 때는 ()를 사용한다.
- (5) 함수를 실행했을 때, 함수의 결과로 만들어지는 값을 ()이라고 부른다.
- (6) 함수 내부에서 어떤 값을 밖으로 전달하고 싶을 때는 () 키워드를 사용한다.
- (7) `function () {}` 형태의 함수와 `() => {}` 형태의 함수는 내부에서 () 키워드의 의미가 다르다.

02 다음 질문에 O×를 표시하십시오.

- (1) 화살표 함수를 만들 때는 `() => {}` 형태를 사용한다.
- (2) 화살표 함수의 본문이 한 줄이라면, `() => {}` 형태에서 `{}` 괄호를 생략할 수 있다.
- (3) 함수는 리턴 값을 넣어서 매개 변수를 만들어 낸다.
- (4) 함수에 리턴 값을 만들었다면, 매개 변수도 반드시 만들어야 한다.
- (5) `let x = 010`이라는 코드의 010은 16진수로 취급되어 x에 16이 들어간다.
- (6) `setTimeout()` 함수와 `setInterval()` 함수의 두 번째 매개 변수에 지정하는 시간은 초 단위로 지정한다.
- (7) 함수는 변수에 저장할 수 없다.
- (8) 함수를 함수의 매개 변수로 전달할 수 있다.
- (9) 익명 함수와 화살표 함수는 작성 방법만 다르고, 기능은 완전히 동일하다.

03 다음 중 함수의 선언 방법으로 옳지 않은 것은?

- | | |
|---|---|
| ① function test() {
console.log("테스트 함수입니다.")
} | ② let test = function () {
console.log("테스트 함수입니다.")
}; |
| ③ function test() = > {
console.log("테스트 함수입니다.")
} | ④ let test = () = > {
console.log("테스트 함수입니다.")
}; |

04 다음 중 문자열 자료형을 숫자 자료형으로 바꾸는 함수가 아닌 것은?

- | | |
|----------------|-----------------|
| ① Number() | ② parseInt() |
| ③ parseFloat() | ④ parseDouble() |

05 다음 중 자바스크립트가 기본으로 제공하는 타이머 관련 함수가 아닌 것은?

- | | |
|-----------------|-------------------|
| ① Timer() | ② setTimeout() |
| ③ setInterval() | ④ clearInterval() |

06 다음 코드의 실행 결과를 예측하시오.

```
function test(a, b, c) {  
    console.log(a);  
    console.log(a * b);  
    console.log(a * b * c);  
}
```

```
test(10, 100);
```

07 다음 중에서 10, "문자열", function () {}처럼 있는 그대로의 자료를 나타내는 단어를 고르시오.

- ① 리터럴
- ② 프리미티브
- ③ 디폴트
- ④ 스탠다드

08 다음 중 16진수를 만들 때 앞에 붙이는 접두사를 고르시오.

- ① 0h
- ② 0x
- ③ 0
- ④ h

09 다음 중 특정 시간마다 계속해서 동작하는 타이머를 만들 때 사용하는 함수를 고르시오.

- ① setTimeout
- ② setInterval
- ③ setLoop
- ④ setTimeTimer

10 다음 중 특정 시간 이후 한 번만 동작하는 타이머를 만들 때 사용하는 함수를 고르시오.

- ① setTimeout
- ② setTimeout
- ③ timeAfter
- ④ callAfter

11 다음 중에서 10을 출력하는 코드를 모두 고르시오.

① function 함수(매개변수) {
 if (매개변수) {
 매개변수 = 10
 }
 console.log(매개변수)
}
함수()

② function 함수(매개변수 = 10) {
 console.log(매개변수)
};
함수()

③ function 함수(매개변수) {
 if (typeof(매개변수) === 'undefined'){
 매개변수 = 10
 }
 console.log(매개변수)
}
함수()

④ function 함수(매개변수) {
 매개변수 = 매개변수 || 10
 console.log(매개변수)
}
함수()

12 다음 함수를 만드시오.

- 함수 이름 : power
 - 매개 변수를 하나 넣으면 제곱해 준다(예 : power(2) = 4).
 - 매개 변수를 2개 넣으면 첫_번째_매개_변수를 두_번째_매개_변수만큼 제곱해 준다(예 : power(2, 3) = 8).
- 함수 이름 : multiply
 - 매개 변수로 넣은 값을 모두 곱해 준다(예 : multiply(1, 2, 3, 4, 5) = 120).

13 다음은 함수 내부와 외부에서 `this` 키워드의 의미가 같은지 다른지 확인하는 실험입니다. 직접 코드를 입력하고 실행해서, `true`를 출력하는 위치와 `false`를 출력하는 위치를 적어 보세요.

참고로 `() => {}()`와 같은 코드는

❶ `()=>{}`로 함수를 만들고,

❷ `(만들어진_함수)()`로 이를 곧바로 호출하는 형태입니다.

이상하게 보일 수 있지만, 많이 사용되는 형태이므로 익숙해지면 좋습니다.

```
let 외부 = this
```

```
// ❶
```

```
console.log(`❶번 위치: ${외부 == this}`);
```

```
// ❷
```

```
() => {
```

```
  console.log(`❷번 위치: ${외부 == this}`);
```

```
}());
```

```
// ❸
```

```
(function () {
```

```
  console.log(`❸번 위치: ${외부 == this}`);
```

```
}());
```

```
// ❹
```

```
function 이름() {
```

```
  console.log(`❹번 위치: ${외부 == this}`);
```

```
}
```

```
이름();
```

01 객체는 다음 방법으로 생성합니다. 객체 내부에 있는 값을 '속성'이라고 하며, 속성 중에 함수 자료형을 '메소드'라고 합니다.

```
let 객체 = {
  속성A: 값,
  속성B: 값,
  메소드A: () => { }
};
```

02 객체는 다음 방법으로 내부 값에 접근합니다.

```
객체.키
객체['키']
```

03 클래스는 기본적으로 다음과 같은 형태를 갖습니다.

```
// 클래스를 선언합니다.
class 클래스 {
  constructor(매개_변수) {
    this.속성 = 매개_변수
  }

  메소드() {
    // 메소드의 내용
  }
}

// 클래스로 인스턴스(객체)를 생성합니다.
let 인스턴스 = new 클래스(매개_변수)
인스턴스.속성
인스턴스.메소드()
```


01 다음 괄호 안을 채우시오.

- (1) 배열은 요소에 인덱스로 접근하고, 객체는 속성에 ()로 접근한다.
- (2) 객체의 속성 중에 함수 자료형인 속성을 ()라고 부른다.
- (3) ()는 '동일한 형태의 객체를 쉽게 만들 수 있게 해 주는 문법'이다.
- (4) 클래스를 기반으로 객체를 만들 때는 () 키워드를 사용한다.

02 다음 질문에 ○×를 표시하십시오.

- (1) 객체는 대괄호([])로 만든다.
- (2) 객체는 키와 값을 쌍으로 저장한다.
- (3) 객체는 배열과 마찬가지로 여러 자료형을 한꺼번에 저장하는 자료형이다.
- (4) 객체의 요소에 접근할 때는 객체[키] 또는 객체.키와 같은 코드를 사용한다.
- (5) 식별자로 사용할 수 있는 문자열만 키로 사용할 수 있다.
- (6) 객체의 메소드에서 자기 자신의 속성에 접근할 때는 this 키워드를 생략할 수 있다.
- (7) 객체는 속성으로 단순한 자료형만 가질 수 있으며, 객체 또는 배열과 같은 복잡한 자료형은 속성으로 가질 수 없다.
- (8) 객체에는 for in 반복문과 for of 반복문을 모두 적용할 수 있다.

03 다음 중 생성자를 만들 때 사용하는 메서드의 이름은?

- | | |
|-------------|-----------------|
| ① construct | ② constructor |
| ③ init | ④ 클래스 이름과 같은 이름 |

04 클래스를 기반으로 만들어진 객체를 부르는 용어를 고르시오.

- | | |
|----------|----------|
| ① 인터리스트 | ② new 객체 |
| ③ 메소드 객체 | ④ 인스턴스 |

05 자바스크립트에서 존재하지 않는다는 것을 나타내는 값을 모두 고르시오.

- ① undefined ② nothing
③ null ④ nil

06 다음 표를 자바스크립트 객체로 나타내시오(키 이름과 자료형은 적절하다고 생각하는 것으로 사용합니다).

속성	값
이름	Nature of Code
가격	30000원
저자	다니엘 슈프만
ISBN	9788968481901
페이지 수	620페이지

07 다음 표의 객체를 생성할 수 있는 클래스를 만드시오(클래스 이름은 Product, 키 이름과 자료형은 적절하다고 생각하는 것으로 사용합니다).

속성	값
이름	돼지삼겹살
무게	100g
가격	1690원

메소드	설명
calculate(⟨무게⟩)	무게를 기반으로 가격을 계산합니다.

예 : `product.calculate(200)`을 입력하면, $100g * 2 = 200g$ 이므로 '3380원'을 출력합니다.

08 다음 코드의 실행 결과를 예측하시오.

```
let object = {  
  "+": 1,  
  "-": 2,  
  "*": 3,  
  "/": 4  
};
```

```
console.log(object.+);  
console.log(object.-);  
console.log(object.*);  
console.log(object./);
```

01 자바스크립트의 기본 자료형은 숫자, 문자열, 불입니다.

02 Number 객체에는 다음 메소드가 있습니다.

메소드	설명
toExponential()	숫자를 지수 표시로 나타낸 문자열을 리턴합니다.
toFixed()	숫자를 고정소수점 표시로 나타낸 문자열을 리턴합니다.
toPrecision()	숫자를 길이에 따라 지수 표시 또는 고정소수점 표시로 나타낸 문자열을 리턴합니다.

03 Number 클래스에는 다음 속성이 있습니다.

속성	설명
MAX_VALUE	자바스크립트의 숫자가 나타낼 수 있는 최대 숫자
MIN_VALUE	자바스크립트의 숫자가 나타낼 수 있는 최소 숫자
NaN	자바스크립트의 숫자로 나타낼 수 없는 숫자
POSITIVE_INFINITY	양의 무한대 숫자
NEGATIVE_INFINITY	음의 무한대 숫자

04 String 객체에는 다음 속성과 메소드가 있습니다.

• 속성

속성	설명
length	문자열의 길이를 나타냅니다.

• 메소드

메소드	설명
charAt(position)	position에 위치하는 문자를 리턴합니다.
charCodeAt(position)	position에 위치하는 문자의 유니코드 번호를 리턴합니다.
concat(args)	매개 변수로 입력한 문자열을 이어 리턴합니다.
indexOf(searchString, position)	앞에서부터 일치하는 문자열의 위치를 리턴합니다.
lastIndexOf(searchString, position)	뒤에서부터 일치하는 문자열의 위치를 리턴합니다.
match(regExp)	문자열 안에 regExp가 있는지 확인합니다.
replace(regExp, replacement)	regExp를 replacement로 바꾼 후 리턴합니다.
search(regExp)	regExp와 일치하는 문자열의 위치를 리턴합니다.
slice(start, end)	특정 위치의 문자열을 추출해 리턴합니다.
split(separator, limit)	separator로 문자열을 잘라 배열을 리턴합니다.
substr(start, count)	start부터 count만큼 문자열을 잘라서 리턴합니다.
substring(start, end)	start부터 end까지 문자열을 잘라서 리턴합니다.
toLowerCase()	문자열을 소문자로 바꾸어 리턴합니다.
toUpperCase()	문자열을 대문자로 바꾸어 리턴합니다.

05 Date 객체는 다음 방법으로 생성합니다.

클래스	설명
new Date()	현재 시간으로 Date 객체를 생성합니다.
New Date(유닉스_타임)	유닉스 타임(1970년 1월 1일 00시 00분 00초부터 경과한 밀리초)으로 Date 객체를 생성합니다.
New Date(시간_문자열)	문자열로 Date 객체를 생성합니다.
New Date(연, 월 - 1, 일, 시간, 분, 초, 밀리초)	시간 요소(연, 월 - 1, 일, 시간, 분, 초, 밀리초)를 기반으로 Date 객체를 생성합니다.

06 Array 객체에는 다음 메소드가 있습니다.

메소드	설명
concat()	매개 변수로 입력한 배열의 요소를 모두 합쳐 배열을 만들어 리턴합니다.
join()	배열 안의 모든 요소를 문자열로 만들어 리턴합니다.
pop()*	배열의 마지막 요소를 제거하고 리턴합니다.
push()*	배열의 마지막 부분에 새로운 요소를 추가합니다.
reverse()*	배열의 요소 순서를 뒤집습니다.
slice()	배열 요소의 지정한 부분을 리턴합니다.
sort()*	배열의 요소를 정렬합니다.
splice()*	배열 요소의 지정한 부분을 삭제하고 삭제한 요소를 리턴합니다.
forEach()	배열의 요소를 하나씩 뽑아 반복을 돌립니다.
map()	콜백 함수에서 리턴하는 것을 기반으로 새로운 배열을 만듭니다.
filter()	콜백 함수에서 true를 리턴하는 것으로만 새로운 배열을 만들어 리턴합니다.

* 표시된 메소드는 자기 자신을 변화시킵니다.

07 자기 자신을 변화시키는 메소드를 '파괴적 메소드'라고 하며, 자기 자신을 변화시키지 않는 메소드를 '비파괴적 메소드'라고 합니다.

08 메소드를 연결해서 사용하는 것을 '메소드 체이닝'이라고 합니다.

01 다음 괄호 안을 채우시오.

- (1) 기본 자료형에 메소드를 추가하고 싶다면 ()을 사용한다.
- (2) 메소드를 연속해서 사용하는 것을 ()이라고 부른다.
- (3) 자기 자신을 변화시키는 메소드를 ()라고 부른다.
- (4) 자기 자신을 변화시키지 않고 리턴하는 메소드를 ()라고 부른다.
- (5) 1970년 1월 1일 00시 00분 00초부터 경과한 밀리초로 시간을 나타내는 단위를 ()이라고 부른다.
- (6) 배열에 요소 하나를 추가할 때는 () 메소드를 사용한다.
- (7) 배열의 마지막 요소를 제거하고 리턴할 때는 () 메소드를 사용한다.
- (8) JSON은 ()의 약자이다.

02 다음 질문에 ○×를 표시하시오.

- (1) typeof 키워드로 new Number()로 만든 숫자의 자료형을 확인하면, number가 나온다.
- (2) typeof 키워드로 new String()로 만든 숫자의 자료형을 확인하면, string이 나온다.
- (3) typeof 키워드로 new Boolean()로 만든 숫자의 자료형을 확인하면, boolean이 나온다.
- (4) let number = 10처럼 숫자를 만들면, 기본 자료형이므로 메소드를 사용할 수 없다.
- (5) JSON은 자바스크립트 객체를 활용한 데이터 표현 방법이므로, 자바스크립트에서 객체를 만드는 방법과 완전히 동일하게 데이터를 만들어 사용할 수 있다.

03 다음 코드를 실행했을 때 출력되는 날짜의 연도, 월, 일을 고르시오.

```
const date = new Date(2000, 12, 10)
console.log(date.toString())
```

- ① 2000년 12월 10일
- ② 2000년 11월 10일
- ③ 2001년 1월 10일
- ④ 오류 발생

04 다음 코드는 오류가 발생합니다. 오류의 이유와 해결 방법을 제시하시오.

```
// 프로토타입에 메소드를 추가합니다.
String.prototype.contain = (input) => {
  return this.indexOf(input) >= -1;
};

// 메소드를 활용합니다.
console.log('안녕하세요'.contain('안녕'));
console.log('안녕하세요'.contain('데굴데굴'));
```

05 다음 표의 데이터를 JSON 형식으로 나타내시오.

이름(name)	미네랄(mineral)	가스(gas)	보급품(supply)
일벌레	50	0	1
여왕	150	0	2
바퀴	75	25	2
히드라리스크	100	50	2
타락귀	150	100	2

06 다음 코드의 실행 결과를 적으시오.

```
let string = "hello world..!";
```

```
string.toUpperCase();
```

```
console.log(string);
```

```
string.toLowerCase();
```

```
console.log(string);
```

07 6번에서 'Hello World..!'와 'hello world..!'를 차례대로 출력하려면 무엇을 변경해야 하는가?

08 lodash 라이브러리 문서를 살펴보고, 배열(컬렉션)에 적용할 수 있는 메소드를 5개 이상 적고, 무엇을 하는 메소드인지 한 줄로 정리하시오.

예: `_.sort()`: 배열을 정렬합니다.

01 문서 객체 모델

웹 브라우저가 HTML 파일을 분석하고 출력하는 방식을 나타냅니다.

02 문서 객체 선택

문서 객체를 선택할 때는 다음 메소드를 사용합니다.

메소드	설명
<code>document.querySelector(선택자)</code>	선택자를 사용해 문서 객체를 선택합니다.
<code>document.querySelectorAll(선택자)</code>	선택자로 여러 개의 문서 객체를 선택합니다.

03 문서 객체 조작

- 글자 조작 : 문서 객체 내부의 글자를 조작할 때는 `innerHTML` 속성을 사용합니다.
- 스타일 조작 : 문서 객체의 스타일을 조작할 때는 `style` 속성을 사용합니다.
- 속성 조작 : 속성을 조작할 때는 다음 메소드를 사용합니다.

메소드	설명
<code>setAttribute(속성_이름, 속성_값)</code>	속성을 지정합니다.
<code>getAttribute(속성_이름)</code>	속성을 추출합니다.

표준에 있는 문서 객체의 속성을 조작할 때는 곧바로 속성 이름을 사용할 수 있습니다.

04 인라인 이벤트 모델과 고전 이벤트 모델

HTML 태그나 자바스크립트 문서 객체가 가지고 있는 다음 속성을 사용해 이벤트를 연결하는 방법입니다.

<code>onblur</code>	<code>onfocus</code>	<code>onfocusin</code>	<code>onfocusout</code>	<code>onload</code>
<code>onresize</code>	<code>onscroll</code>	<code>onunload</code>	<code>onclick</code>	<code>ondblclick</code>
<code>onmousedown</code>	<code>onmouseup</code>	<code>onmousemove</code>	<code>onmouseover</code>	<code>onmouseout</code>
<code>onmouseenter</code>	<code>onmouseleave</code>	<code>onchange</code>	<code>onselect</code>	<code>onsubmit</code>
<code>onkeydown</code>	<code>onkeypress</code>	<code>onkeyup</code>	<code>onerror</code>	

01 다음 괄호 안을 채우시오.

- (1) 간단하게 문서 객체 모델이란 () 객체와 관련된 객체의 집합을 의미한다.
- (2) () 메소드는 선택자를 사용해 문서 객체를 하나 선택한다.
- (3) () 메소드는 선택자를 사용해 문서 객체를 여러 개 선택한다.
- (4) 문서 객체 내부에 있는 글자를 HTML 형태로 조작할 때는 () 속성을 사용한다.
- (5) 문서 객체 내부의 HTML 글자를 조작할 때, < 문자를 넣고 싶을 때는 ()를 사용하고, > 문자를 넣고 싶을 때는 ()를 사용한다.
- (6) 문서 객체의 스타일을 조작할 때는 () 속성을 사용한다.
- (7) 웹 표준에서 지원하지 않는 속성은 일반적으로 이름 앞부분에 ()라는 접두사를 붙인다. 이를 사용자 지정 속성이라고 부른다.
- (8) 이벤트 속성에 넣는 함수를 () 또는 ()라고 부른다.

02 다음 질문에 OX를 표시하시오.

- (1) 노드는 요소 노드와 텍스트 노드로 구분된다.
- (2) 모든 노드는 텍스트 노드를 갖는다.
- (3) 웹 브라우저는 HTML 파일을 위에서 아래로 읽으며, 노드를 구성한다.
- (4) script 태그는 반드시 body 태그의 아래 부분에 놓아야 한다.
- (5) onready 이벤트를 사용하면, '웹 페이지 준비가 완료된 후'라는 이벤트를 잡을 수 있다.
- (6) 존재하지 않는 속성도 attr 속성으로 조작할 수 있다.

03 다음 중에서 문서 객체의 스타일을 변경할 때 사용하는 속성은?

- | | |
|--------------|------------|
| ① styles | ② style |
| ③ appearance | ④ exterior |

04 선택자를 활용해서 문서 객체 하나를 선택하는 메소드를 고르시오.

- ① document.query()
- ② document.querySelector()
- ③ document.querySelector()
- ④ document.queryBySelector()

05 다음 표의 오른쪽 부분을 채우시오.

스타일 시트의 스타일 속성	자바스크립트 문서 객체의 스타일 속성
font-family	
border-radius	
background-size	
border	

06 선택자를 활용해서 문서 객체 여러 개를 선택하는 메서드를 고르시오.

- ① document.querys()
- ② document.querySelectors()
- ③ document.querySelectorAll()
- ④ document.querysSelector()

07 다음 중 문서 객체 내부의 글자를 수정할 때 사용할 수 있는 속성을 모두 고르시오.

- ① text
- ② textContent
- ③ innerHTML
- ④ inner

08 다음 중 문서 객체의 속성을 변경할 때 사용하는 메서드를 고르시오.

- ① `setAttribute()`
- ② `setAttr()`
- ③ `setProperty()`
- ④ `setProp()`