

JAVASCRIPT

자바스크립트 레퍼런스 북

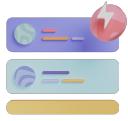




INDEX

데이터 저장하기

- | | |
|---------------------------|----------------------------|
| 01. 변수 : 데이터 저장 | 09. 객체 : 표현 방법1 |
| 02. 변수 : 데이터 저장 + 데이터 변경 | 10. 객체 : 표현 방법2 |
| 03. 변수 : 데이터 저장 + 변경 + 추가 | 11. 객체 : 표현 방법3 |
| 04. 상수 : 데이터 저장 (변하지 않음) | 12. 객체 : 표현 방법4 |
| 05. 배열 : 표현 방법1 | 13. 객체 : 표현 방법5 : 배열 속에 객체 |
| 06. 배열 : 표현 방법2 | 14. 객체 : 표현 방법6 : 객체 속에 배열 |
| 07. 배열 : 표현 방법3 | 15. 객체 : 표현 방법7 : 객체 속에 변수 |
| 08. 배열 : 표현 방법4 | 16. 객체 : 표현 방법8 : 객체 속에 함수 |



INDEX

데이터 불러오기

- 01. 변수 : 기본
- 02. 상수 : 기본
- 03. 배열 : 기본
- 04. 배열 : 2차 배열
- 05. 배열 : 갯수 구하기
- 06. 배열 : for문
- 07. 배열 : forEach()
- 08. 배열 : for of문
- 09. 배열 : for in문
- 10. 배열 : map()문
- 11. 배열 : 펼침 연산자
- 12. 객체 : 기본
- 13. 객체: Object
- 14. 객체 : 변수
- 15. 객체 : for in문
- 16. 객체 : map()
- 17. 객체 : hasOwnProperty()
- 18. 객체 : 펼침 연산자 - 복사
- 19. 객체 : 펼침 연산자 - 추가
- 20. 객체 : 펼침 연산자 - 결합
- 21. 객체 : 비구조화 할당
- 22. 객체 : 비구조화 할당 : 별도 이름 저장



INDEX

데이터 실행하기

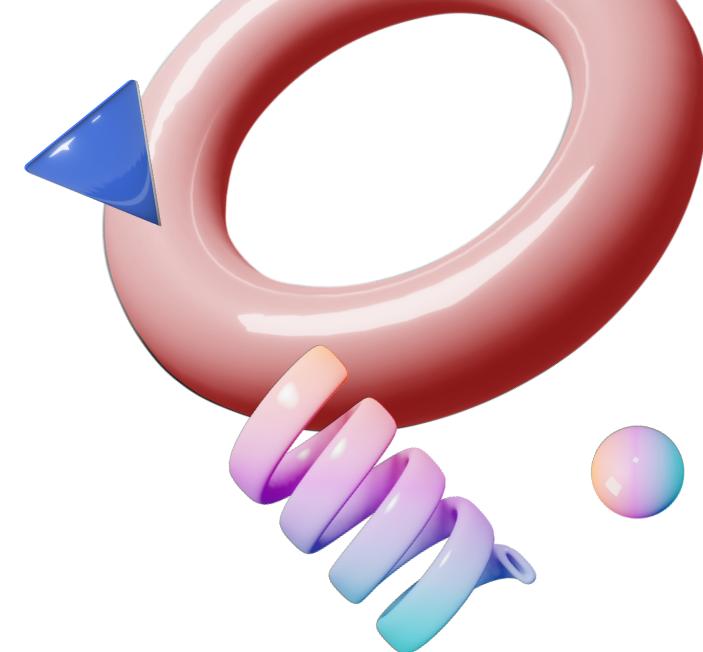
01. 선언적 함수
02. 익명 함수
03. 매개변수 함수
04. 리턴값 함수
05. 화살표 함수 : 선언전 함수
06. 화살표 함수 : 익명함수
07. 화살표 함수 : 리턴값 함수
08. 화살표 함수 : 익명함수 + 매개변수 + 리턴값
09. 화살표 함수 : 팔호생략
10. 화살표 함수 : 팔호생략 + 리턴 생략
11. 화살표 함수 : 팔호 생략 + 리턴 생략
12. 파라미터 함수

데이터 제어하기

01. if문
02. 다중 if문
03. 중첩 if문
04. switch문
05. 조건부 연산자
06. if문 생략
07. while문
08. do while문
09. for문
10. 중첩 for문
11. forEach()
12. for of문
13. for in문
14. break문
15. continue문

자바스크립트란?

JavaScript는 웹을 위한 인터프리터 언어이자 객체기반의 스크립트 프로그래밍 언어이며 HTML의 특정 요소(들)을 선택하여 다양한 이벤트 (마우스 클릭, 키보드 입력 등)에 따라 어떤 동작을 하도록 기능을 넣을 수 있으며 발생하는 이벤트에 따라 HTML, CSS를 조작할 수도 있고 그 외에도 여러가지를 할 수 있다.



구조

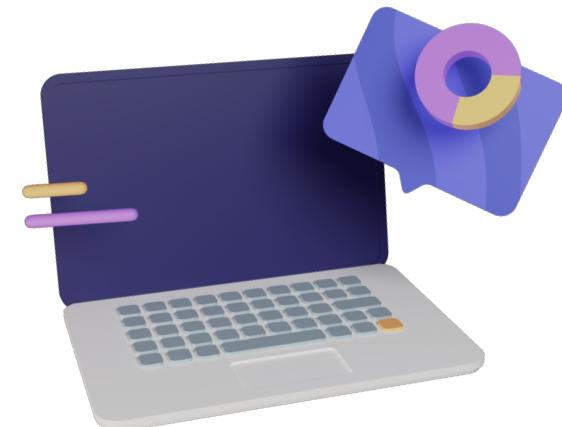


디자인



동작

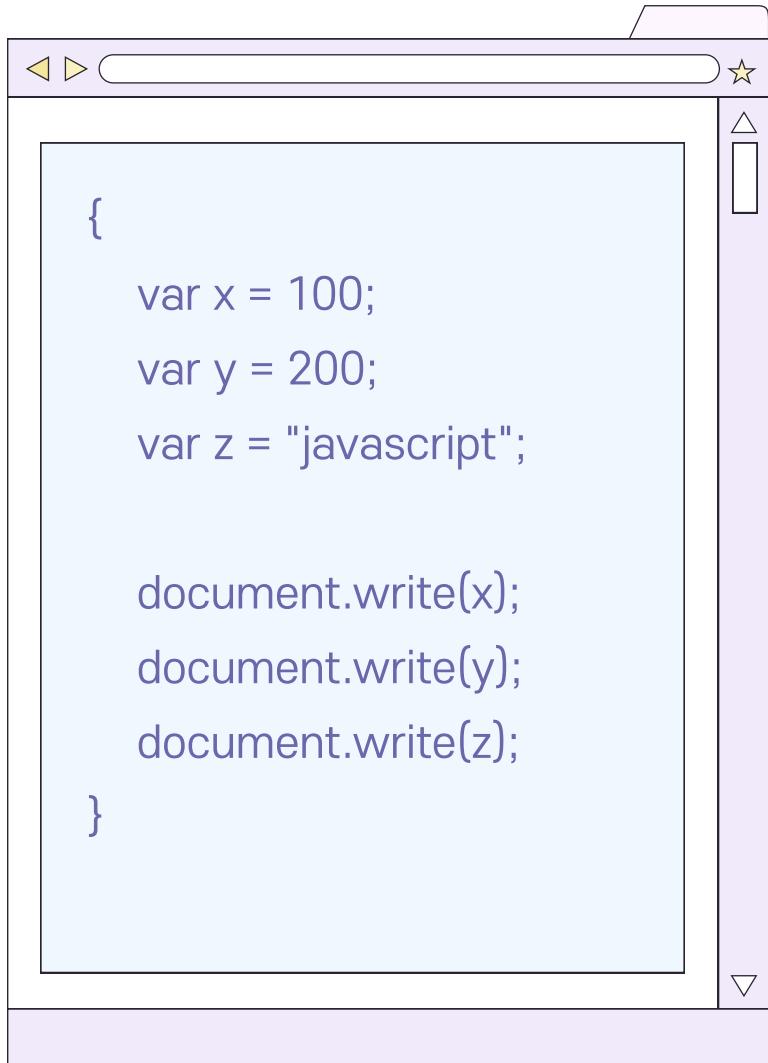
=



웹사이트

데이터 불러오기_01.변수

JAVASCRIPT 문법정리



A screenshot of a web browser window. The code in the script tag is:

```
{  
    var x = 100;  
    var y = 200;  
    var z = "javascript";  
  
    document.write(x);  
    document.write(y);  
    document.write(z);  
}
```

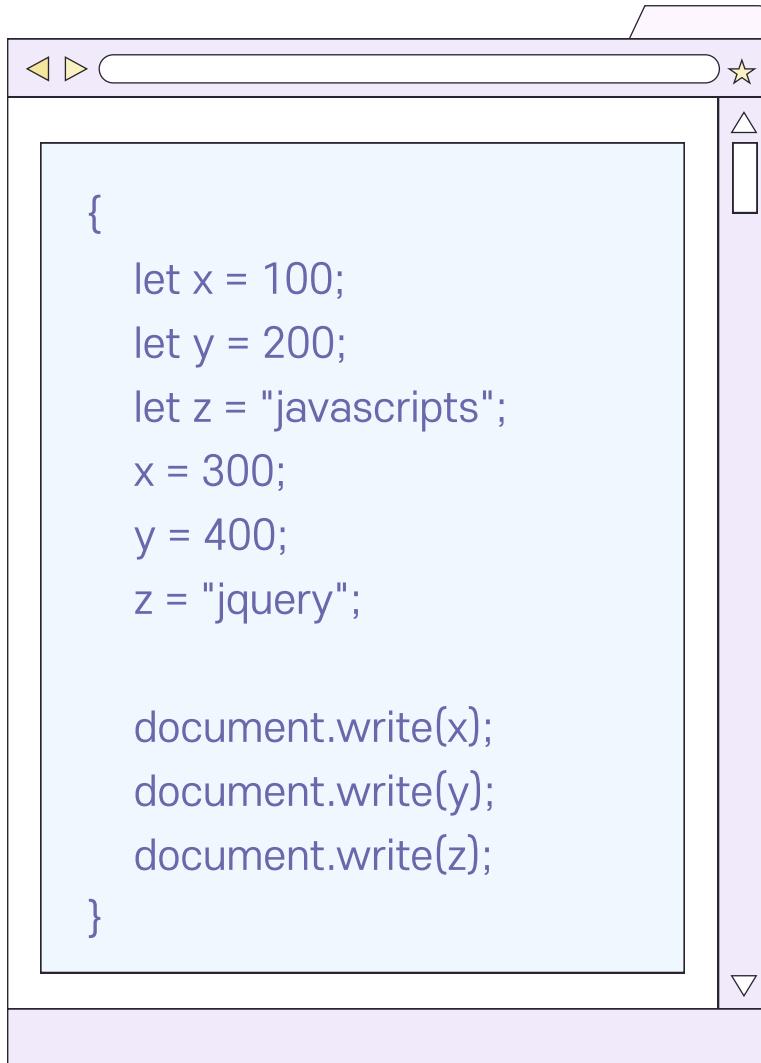
결과값
100
200
javascript

변수는 데이터를 저장할 수 있습니다.
변수는 var, let , const로 표현 할 수 있습니다.
var 은 중복 선언이 가능 합니다.
const, let 은 중복 선언이 불가능 합니다.

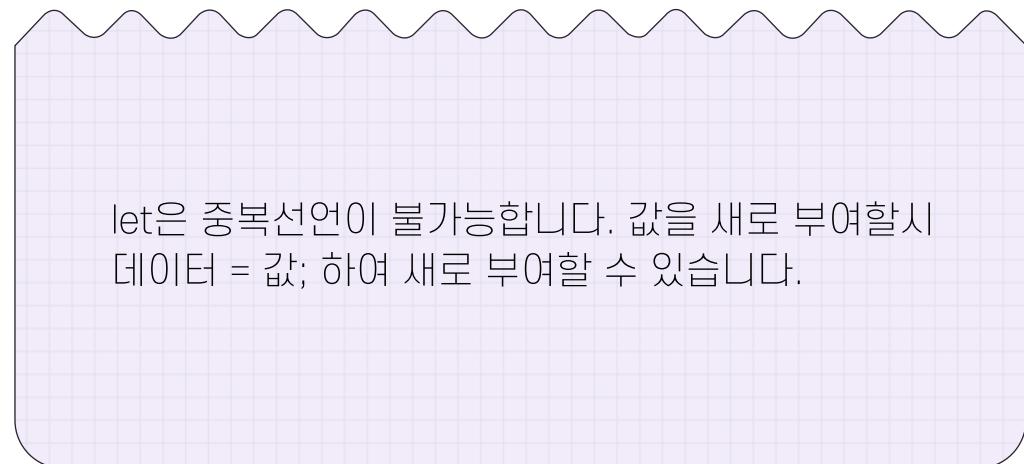
데이터 불러오기_02. 변수

변수 : 데이터 저장 + 변경

JAVASCRIPT 문법정리

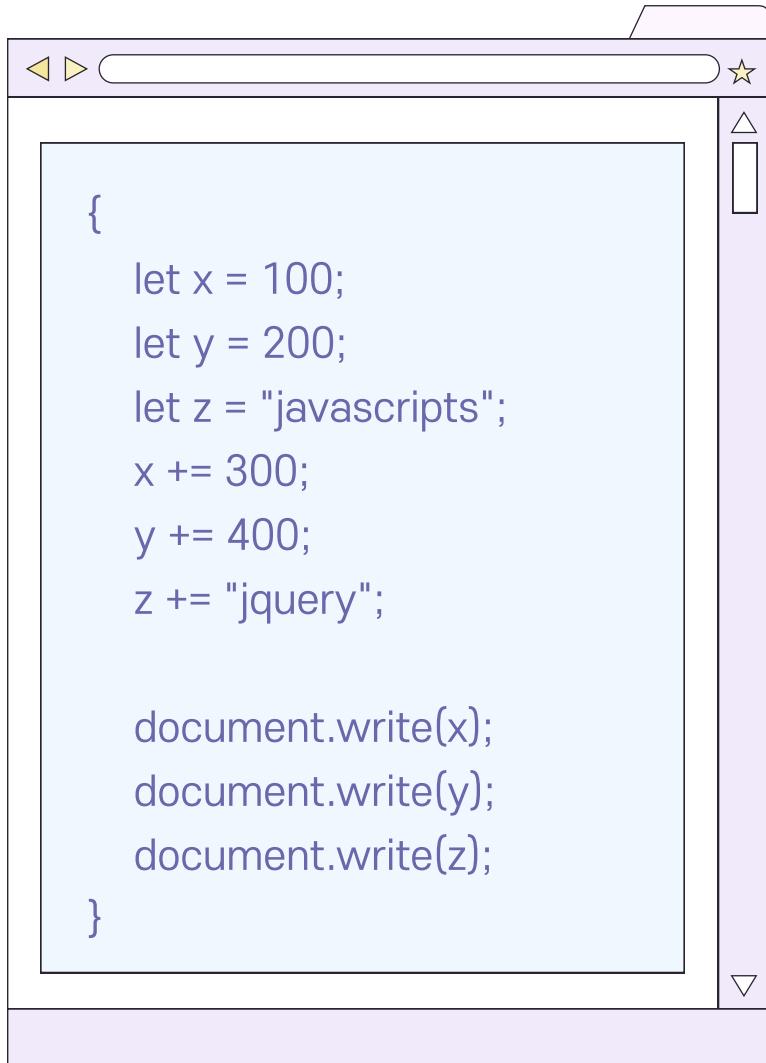


```
{  
    let x = 100;  
    let y = 200;  
    let z = "javascripts";  
    x = 300;  
    y = 400;  
    z = "jquery";  
  
    document.write(x);  
    document.write(y);  
    document.write(z);  
}
```



데이터 불러오기_03.변수

JAVASCRIPT 문법정리



```
{  
let x = 100;  
let y = 200;  
let z = "javascripts";  
x += 300;  
y += 400;  
z += "jquery";  
  
document.write(x);  
document.write(y);  
document.write(z);  
}
```

결과값
400
600
javascriptjquery

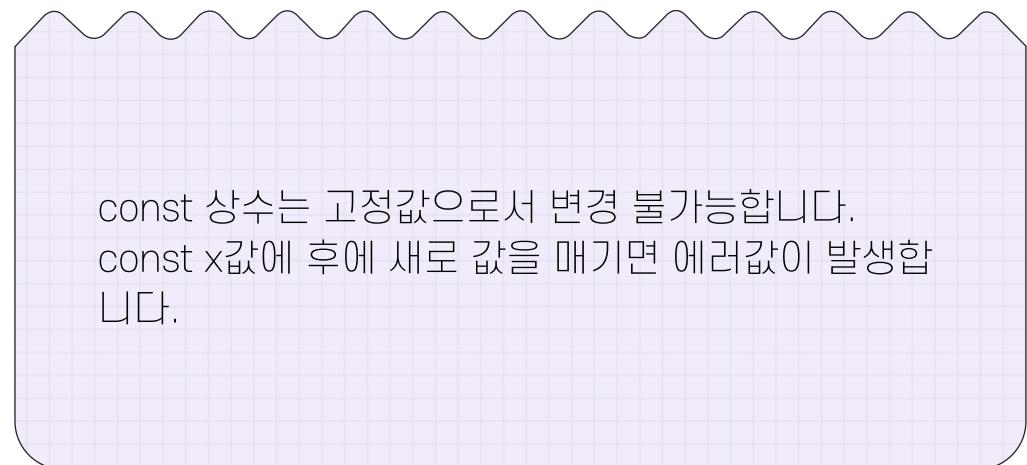
+는 왼쪽의 값에 오른쪽의 값을 추가로 더한것으로서
 $x = x + 300 / y = y + 400 / z = z + "jquery"$ 로 결과
값이 도출됩니다.

데이터 불러오기_04.상수

상수 : 데이터 저장소

JAVASCRIPT 문법정리

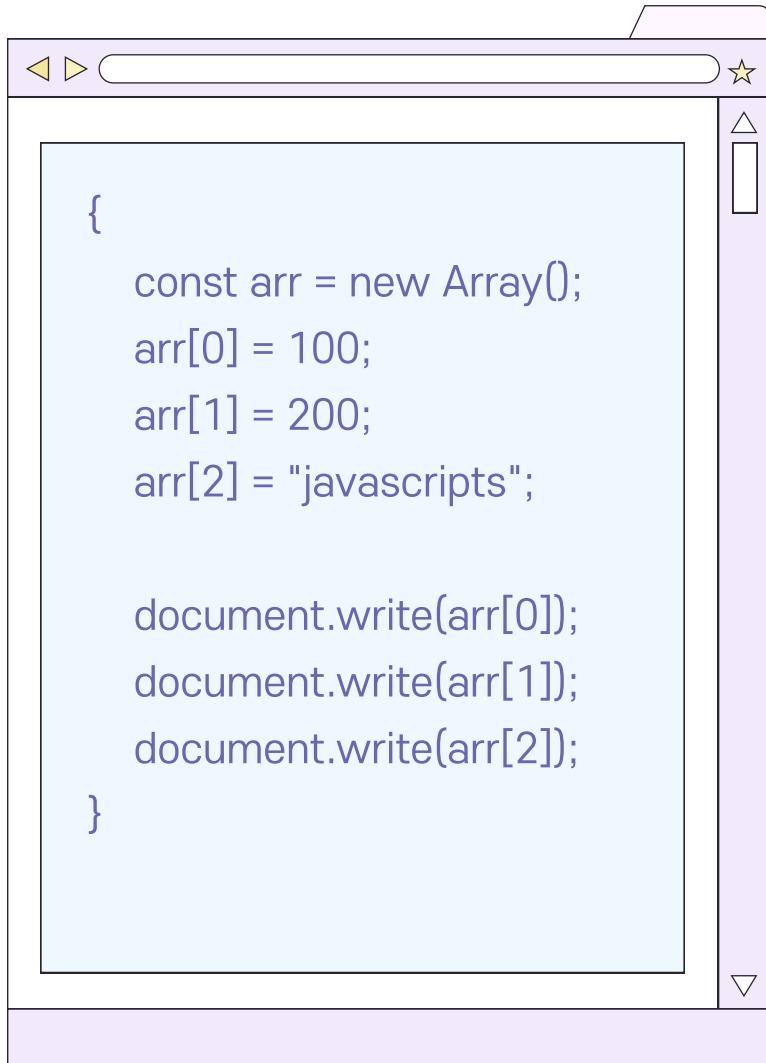
```
{  
    const x = 100;  
    const y = 200;  
    const z = "javascript";  
  
    document.write(x);  
    document.write(y);  
    document.write(z);  
}
```



데이터 불러오기_05. 배열

배열 : 여러개의 데이터 저장

JAVASCRIPT 문법정리



```
{  
    const arr = new Array();  
    arr[0] = 100;  
    arr[1] = 200;  
    arr[2] = "javascript";  
  
    document.write(arr[0]);  
    document.write(arr[1]);  
    document.write(arr[2]);  
}
```



new Array();로 배열을 선언한 후,
arr[0]은 첫번째 배열 저장소에 숫자 100을 저장한 값
arr[1]은 두번째 배열 저장소에 숫자 200을 저장한 값
arr[3]은 세번째 배열 저장소에 문자를 저장한 값으로서
[0]부터 첫번째로 시작됩니다.

데이터 불러오기_06. 배열

배열 : 여러개의 데이터 저장 : 표현 방법2

[JAVASCRIPT 문법정리](#)

The screenshot shows a web browser window. On the left, a code editor pane displays the following JavaScript code:

```
{  
    const arr = new Array(100, 200, "javascript");  
  
    document.write(arr[0]);  
    document.write(arr[1]);  
    document.write(arr[2]);  
}
```

On the right, a results panel displays the output of the code:

결과값
100
200
javascript

배열 선언시 할당 값을 매겨주는 것으로, 따로 0에 100, 1에 200, 3에 javascript라고 선언해주지 않아도 값이 할당이 됩니다.

데이터 불러오기_07. 배열

배열 : 여러개의 데이터 저장

JAVASCRIPT 문법정리

```
{  
    const arr = [ ];  
    arr[0] = 100;  
    arr[1] = 200;  
    arr[2] = "javascript";  
  
    document.write(arr[0]);  
    document.write(arr[1]);  
    document.write(arr[2]);  
}
```



new Array(); 대신 []; 로 배열을 선언한 후,
arr[0]은 첫번째 배열 저장소에 숫자 100을 저장한 값
arr[1]은 두번째 배열 저장소에 숫자 200을 저장한 값
arr[3]은 세번째 배열 저장소에 문자를 저장한 값으로서
[0]부터 첫번째로 시작됩니다.

데이터 불러오기_08.배열

배열 : 여러개의 데이터 저장 : 표현 방법4

JAVASCRIPT 문법정리

A screenshot of a web browser window. On the left, a code editor pane displays the following JavaScript code:

```
{  
    const arr =[100, 200,"javascript"];  
  
    document.write(arr[0]);  
    document.write(arr[1]);  
    document.write(arr[2]);  
}
```

The browser's title bar shows a yellow double arrow icon. The results panel on the right shows the output of the code:

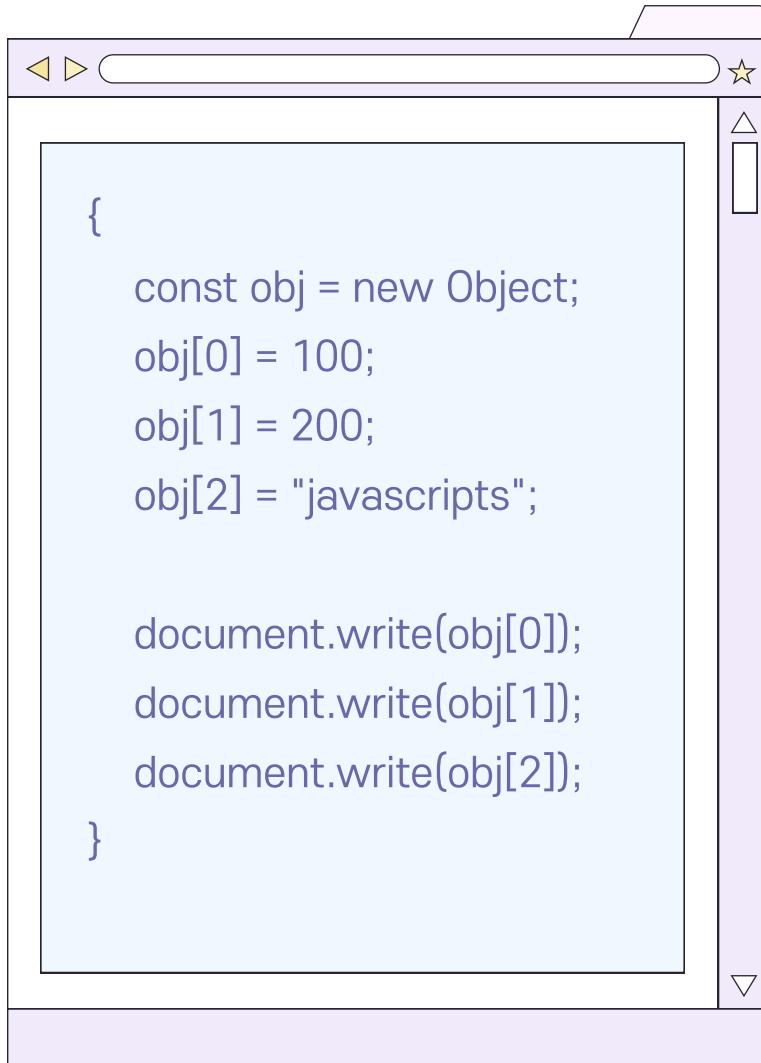
결과값
100
200
javascript

배열 선언시 [] ;안에 할당 값을 매겨주는 것으로 따로 0에 100, 1에 200, 3에 javascript 라고 선언해주지 않아도 값이 할당이 됩니다.

데이터 불러오기_09. 객체

객체 : 데이터 저장소(키와 값)

JAVASCRIPT 문법정리



```
{  
    const obj = new Object;  
    obj[0] = 100;  
    obj[1] = 200;  
    obj[2] = "javascript";  
  
    document.write(obj[0]);  
    document.write(obj[1]);  
    document.write(obj[2]);  
}
```



new Object();로 객체 선언한 후,
obj[0]은 첫번째 배열 저장소에 숫자 100을 저장한 값
obj[1]은 두번째 배열 저장소에 숫자 200을 저장한 값
obj[3]은 세번째 배열 저장소에 문자를 저장한 값으로서
[0]부터 첫번째로 시작됩니다.

데이터 불러오기_10. 객체

객체 : 데이터 저장소(키와 값) : 표현방법2

JAVASCRIPT 문법정리

A screenshot of a web browser window. The address bar shows a placeholder URL. The main content area contains a script editor with the following code:

```
{  
    const obj = new Object();  
    obj.a = 100;  
    obj.b = 200;  
    obj.c = "javascript";  
  
    document.write(obj.a);  
    document.write(obj.b);  
    document.write(obj.c);  
}
```

The browser's status bar at the bottom indicates "Page 1 of 1".



new Object()로 객체를 선언해준 후,
obj.a는 첫번째 배열 저장소에 숫자 100을 저장한 값
obj.b는 두번째 배열 저장소에 숫자 200을 저장한 값
obj.c는 세번째 배열 저장소에 문자를 저장한 값으로서
알파벳 a부터 첫번째로 시작됩니다.

데이터 불러오기_11. 객체

객체 : 데이터 저장소 : 표현방법3 : 키워드생략

JAVASCRIPT 문법정리

A screenshot of a web browser window. The address bar shows a placeholder URL. The main content area contains a script editor with the following code:

```
{  
    const obj = { };  
    obj.a = 100;  
    obj.b = 200;  
    obj.c = "javascript";  
  
    document.write(obj.a);  
    document.write(obj.b);  
    document.write(obj.c);  
}
```

The browser's status bar at the bottom indicates "Page 1 of 1".



new Object(); 대신 {}; 로 배열을 선언한 후,
obj.a은 첫번째 배열 저장소에 숫자 100을 저장한 값
obj.b은 두번째 배열 저장소에 숫자 200을 저장한 값
obj.c은 세번째 배열 저장소에 문자를 저장한 값으로서
알파벳 a부터 첫번째로 시작됩니다.

데이터 불러오기_12. 객체

객체 : 데이터 저장소 : 표현방법4

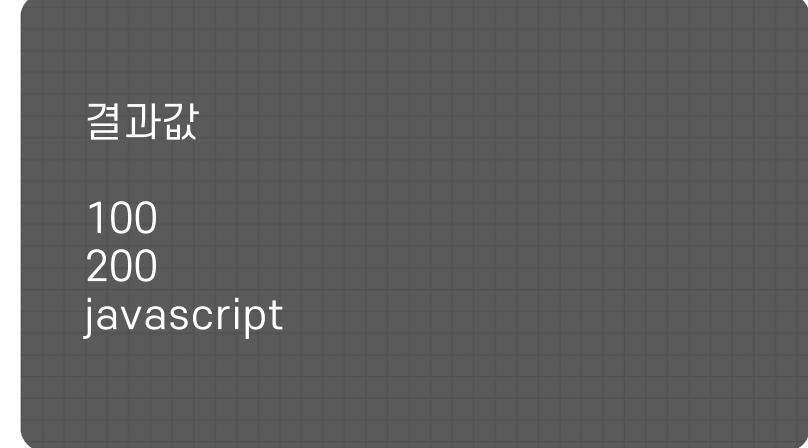
JAVASCRIPT 문법정리

A screenshot of a web browser window. On the left, a code editor pane displays the following JavaScript code:

```
{  
    const arr =[a:100, b:200, c:"javascript"];  
  
    document.write(obj.a);  
    document.write(obj.b);  
    document.write(obj.c);  
}
```

The browser's title bar shows a yellow star icon. The results panel on the right shows the output of the code:

결과값
100
200
javascript



객체 선언시 [];안에 할당 값을 매겨주는 것으로 따로 0에 100, 1에 200, 3에 javascript 라고 선언해주지 않아도 값이 할당이 됩니다.

데이터 불러오기_13. 객체

객체 : 표현 방법5 : 배열 속에 객체

JAVASCRIPT 문법정리

A screenshot of a web browser window. The address bar shows a placeholder URL. The main content area contains a code editor with the following JavaScript code:

```
{  
  const obj = [  
    { a :100, b : 200 },  
    { c : "javascripts"}  
];  
  
document.write(obj[0].a);  
document.write(obj[0].b);  
document.write(obj[1].c);  
}
```

The browser's status bar at the bottom indicates the page is loaded.

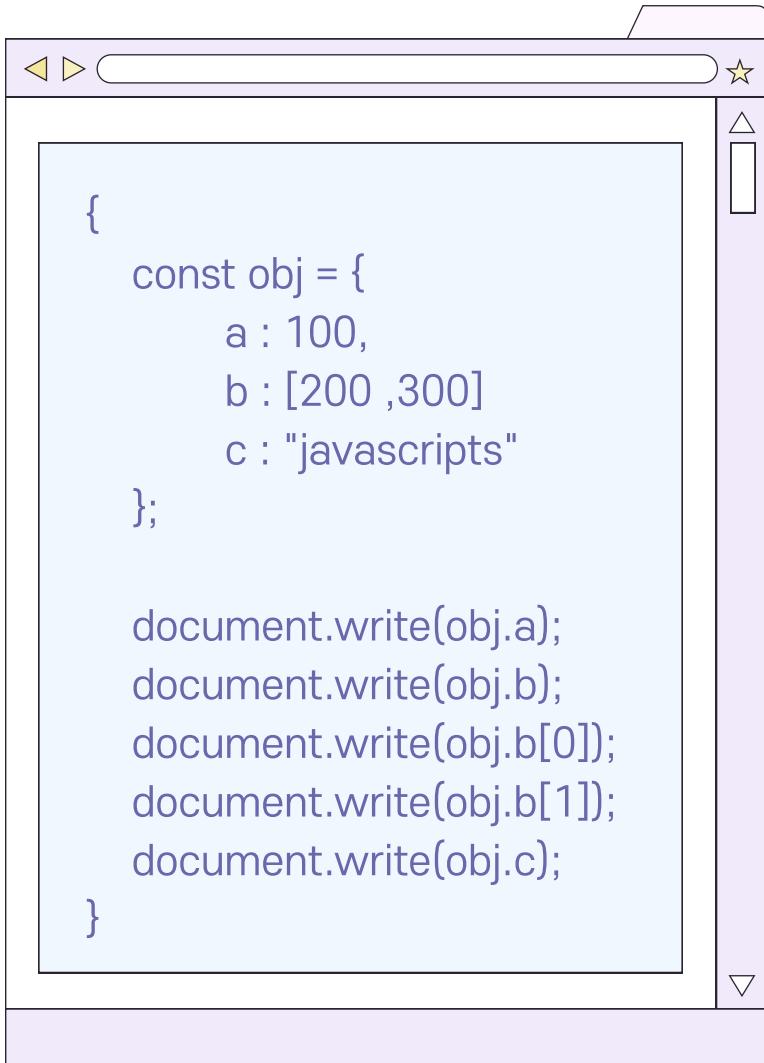


배열[]속에 객체{}를 선언한 것으로서, a값 b값을 [0]배열 안에 넣어놓는 값으로 설정하였습니다.
결과값을 도출할 때에, obj[0]배열.a{객체값}을
불러내어 결과를 도출합니다.

데이터 불러오기_14. 객체

객체 : 표현 방법6 : 객체 속에 배열

JAVASCRIPT 문법정리



A screenshot of a web browser window. The code in the script pane is:

```
{  
    const obj = {  
        a : 100,  
        b : [200 ,300]  
        c : "javascript"  
    };  
  
    document.write(obj.a);  
    document.write(obj.b);  
    document.write(obj.b[0]);  
    document.write(obj.b[1]);  
    document.write(obj.c);  
}
```

The output in the console pane shows the results of the code execution:

```
100  
200,300  
200  
300  
javascript
```

결과값

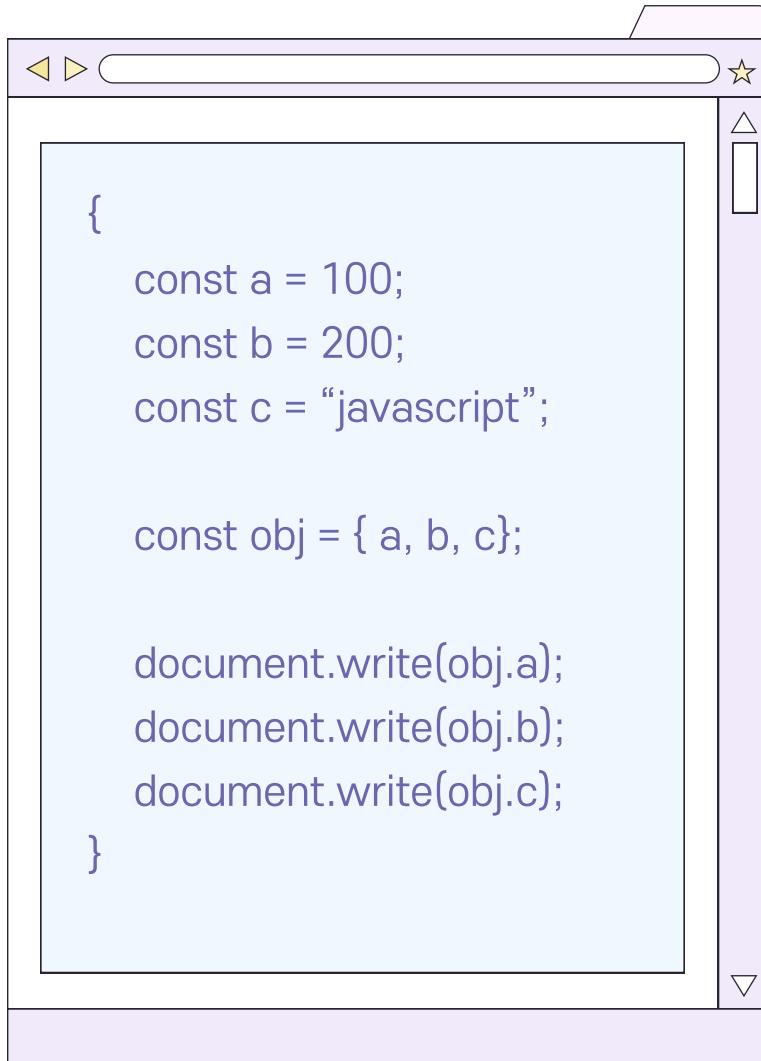
100
200,300
200
300
javascript

객체{}속에 배열[]을 선언한 것으로서, b값 안에 배열로 [200, 300]을 넣어놓는 값으로 설정하였습니다. 결과 값을 도출할 때에, obj.b객체[0]배열을 불러내어 결과를 도출합니다.

데이터 불러오기_15. 객체

객체 : 표현 방법7 : 객체 속에 변수

[JAVASCRIPT 문법정리](#)



```
{  
    const a = 100;  
    const b = 200;  
    const c = "javascript";  
  
    const obj = { a, b, c};  
  
    document.write(obj.a);  
    document.write(obj.b);  
    document.write(obj.c);  
}
```

결과값
100
200
javascript

변수 a는 100, b는 200, c는 "javascript"라고 선언한 후, obj오브젝트는 a,b,c를 포함하고 있다고 객체를 선언하는 방법입니다.

데이터 불러오기_16. 객체

객체 : 표현방법8 : 객체 속에 함수

JAVASCRIPT 문법정리

A screenshot of a web browser window. The code in the script pane is:

```
{  
    const obj = {  
        a : 100;  
        b : [200, 300],  
        c : "javascript";  
        d : function() {  
            document.write("자바스크립트가 실행되었습니다.")  
        },  
        e : function() {  
            document.write( obj.c "가 실행되었습니다.")  
        },  
        f : function() {  
            document.write( this.c "가 실행되었습니다.")  
        },  
    }  
    document.write(obj.a);  
    document.write(obj.b);  
    document.write(obj.b[0]);  
    document.write(obj.b[1]);  
    document.write(obj.c);  
    obj.d();  
    obj.e();  
    obj.f();  
}
```

The output in the console pane shows the results of the code execution:

결과값
100
200,300
200
300
javascript
자바스크립트가 실행되었습니다.
javascript가 실행되었습니다.
javascript가 실행되었습니다.

const obj를 선언 후 객체 안에 함수를 넣어놓은 것으로, 함수안에 document.write를 선언해놓았기 때문에, 결과값을 도출 할때 따로 작성하지 않아도 값이 나오게 됩니다.