

## ■ 문자열

- 문자를 표현할 때 사용하는 자료의 형태
- alert() 함수의 매개 변수로 쓰인 'Hello JavaScript..!' 와 같은 자료
- 문자열을 만드는 방법
  - "동해물과 백두산이" (큰 따옴표)
  - '동해물과 백두산이' (작은 따옴표)

## ■ 문자열

### ● 예외적인 문자열 사용

- 내부에 작은 따옴표를 쓰고 싶으면 외부에 큰 따옴표
- 내부에 큰 따옴표를 쓰고 싶으면 외부에 작은 따옴표

```
<script>  
    alert('This is "string"');  
    alert("This is 'string'");  
</script>
```

## ■ 문자열

### ● 예외적인 문자열 사용

- 한가지 방식으로만 사용하려는 경우 예외 문자 사용

```
<script>  
  alert('This is ₩string₩');  
  alert("This is ₩string₩");  
</script>
```

- 자주 사용되는 예외 문자

| 예외 문자 | 기능     |
|-------|--------|
| ₩t    | 탭      |
| ₩n    | 줄바꿈    |
| ₩'    | 작은 따옴표 |
| ₩"    | 큰 따옴표  |
| ₩₩    | 역 슬래시  |

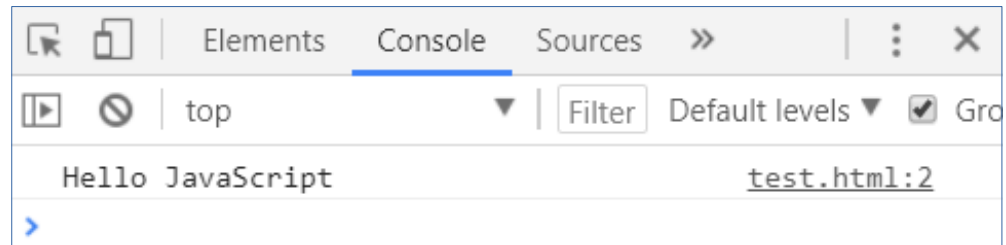
## ■ 문자열

### ● 문자열 연결

- 'Hello' + ' JavaScript'

-> Hello JavaScript

```
<script>  
  console.log("Hello JavaScript");  
</script>
```



## ■ 숫자

### ● 정수와 실수의 구분없이 사용

- 273

- 52.273

### ● 연산

- 우선순위에 따라 괄호 사용 가능

```
<script>  
    alert(5 + 3 * 2);  
    alert((5 + 3) * 2);  
</script>
```

이 페이지 내용:

11

확인

이 페이지 내용:

16

확인

## ■ 숫자

### ● 연산

#### – 나머지 구하기

```
<script>  
    alert(5 % 4);  
    alert(13 % 10);  
</script>
```

이 페이지 내용:

1

확인

이 페이지 내용:

3

확인

## ■ 논리

### ● 비교 연산자

- 두 대상을 비교할 수 있는 연산자

| 연산자 | 설명                           |
|-----|------------------------------|
| >=  | 좌변이 우편보다 크거나 같다              |
| <=  | 좌변이 우편보다 적거나 같다              |
| >   | 좌변이 크다                       |
| <   | 좌변이 적다                       |
| ==  | 좌변과 우변이 같다 (비교)              |
| !=  | 좌변과 우편이 같지 않다 (비교)           |
| === | 좌변과 우변의 자료형과 값이 같다 (일치)      |
| !== | 좌변과 우변의 자료형 또는 값이 같지 않다 (일치) |

## ■ 논리

### ● 비교 연산자

- 문자열은 사전의 앞쪽에 있을수록 작은 값을 가짐

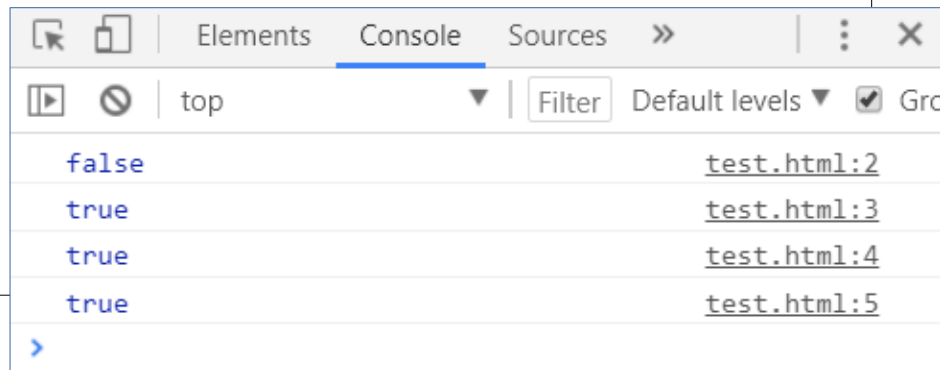
• '가' > '마' -> false

• '가' > '1' -> true

• 'A' > '1' -> true

• 숫자 < 영어 < 한글(특수문자)

```
<script>
  console.log('가' > '마');
  console.log('가' > '1');
  console.log('A' > '1');
  console.log('가' > 'A');
</script>
```



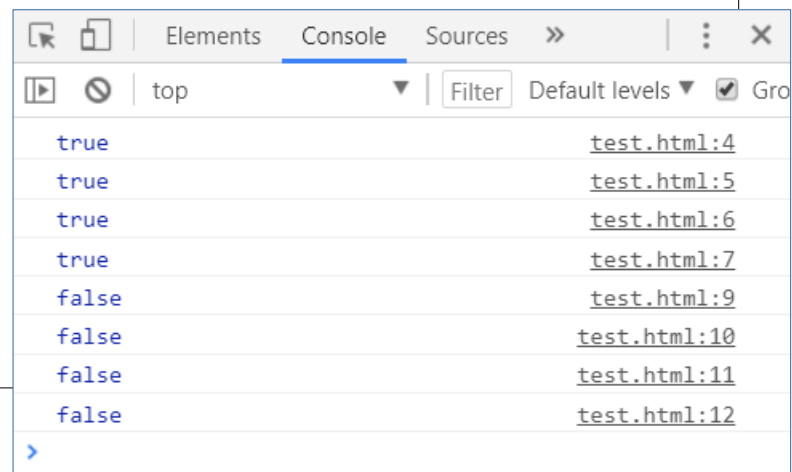


## ■ 논리

### ● 비교 / 일치 연산자

```
<script>
  console.log("" == false);
  console.log("" == 0);
  console.log(0 == false);
  console.log("99" == 99);

  console.log("" === false);
  console.log("" === 0);
  console.log(0 === false);
  console.log("99" === 99);
</script>
```

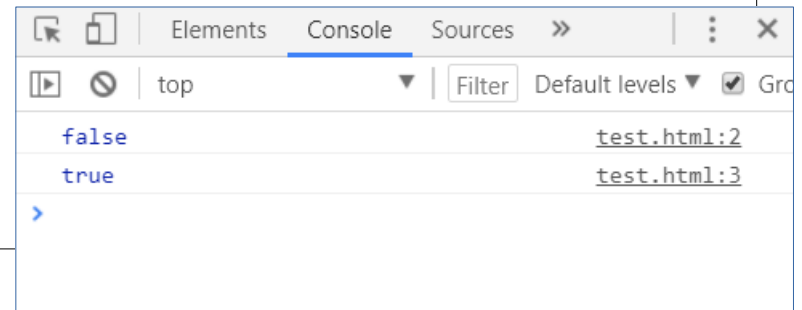


## ■ 논리

### ● 논리 연산자

| 연산자 | 설명    |
|-----|-------|
| !   | 논리 부정 |
| &&  | 논리 곱  |
|     | 논리 합  |

```
<script>  
  console.log(!true);  
  console.log(!(3 > 4));  
</script>
```



## ■ 논리

- 논리 곱 : 좌변과 우변이 모두 참일때만 참

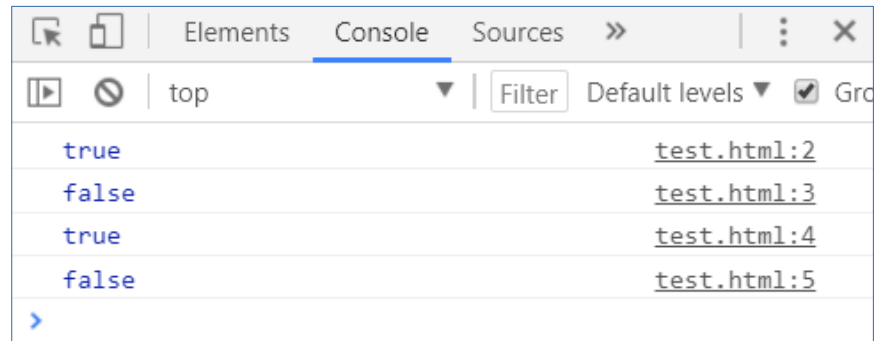
| 좌변          | 우변          | 결과          |
|-------------|-------------|-------------|
| <u>true</u> | <u>true</u> | <u>true</u> |
| true        | false       | false       |
| false       | true        | false       |
| false       | false       | false       |

- 논리 합 : 좌변과 우변이 모두 거짓일때만 거짓

| 좌변           | 우변           | 결과           |
|--------------|--------------|--------------|
| true         | true         | true         |
| true         | false        | true         |
| false        | true         | true         |
| <u>false</u> | <u>false</u> | <u>false</u> |

## ■ 논리

```
<script>  
  console.log(10 > 5 && 10 < 11);  
  console.log(10 > 5 && 10 < 10);  
  console.log(2 < 1 || 2 < 10);  
  console.log(2 < 1 || 2 < -10);  
</script>
```



## ■ 변수

### ● 값을 저장할 때 사용하는 식별자

#### – 변수 선언

var 식별자;

```
<script>  
  var data;  
</script>
```

#### – 변수 초기화 (값 할당)

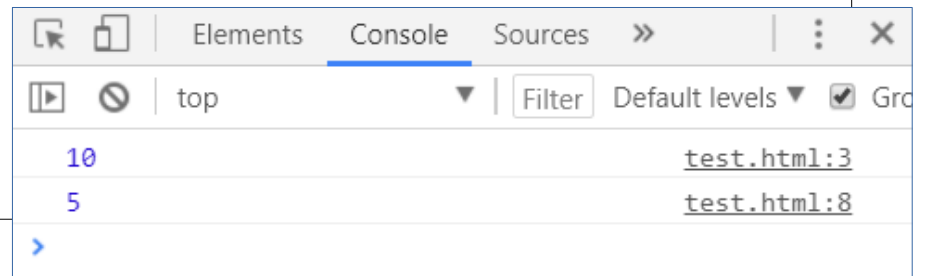
식별자 = 값;

```
<script>  
  var data;  
  data = 10; // var data = 10;  
</script>
```

## ■ 변수

### ● 변수 사용

```
<script>  
  var data = 10;  
  console.log(data);  
  
  var a = 2;  
  var b = 3;  
  var c = a + b;  
  console.log(c);  
</script>
```



## ■ 변수

### ● 변수로 사용 가능한 자료형

```
<script>
  var str = 'string'; // 문자열
  console.log(str);

  var num = 2324; // 숫자
  console.log(num);

  var isFile = true; // 논리
  console.log(isFile);

  var myFunc = function() {}; // 함수
  console.log(myFunc);

  var obj = {}; // 객체
  console.log(obj);

  var value = null; // null
  console.log(value);

  var arr = []; // 배열
  console.log(arr);

  var def; // undefined
  console.log(def);
</script>
```

## ■ 변수

### ● 복합 대입 연산자

| 연산자   | 설명                |
|-------|-------------------|
| $+=$  | 기존 변수에 값을 더함      |
| $-=$  | 기존 변수에 값을 뺌       |
| $*=$  | 기존 변수에 값을 곱함      |
| $/=$  | 기존 변수에 값을 나눔      |
| $\%=$ | 기존 변수의 값에 나머지를 구함 |

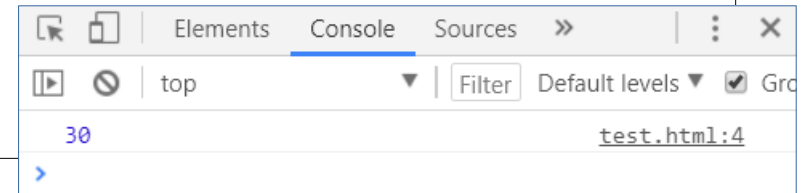


## ■ 변수

### ● 복합 대입 연산자

#### – 숫자

```
<script>
  var num = 20;
  num += 10;
  console.log(num);
</script>
```



#### – 문자열

```
<script>
  var str = '';
  str += '<ul>';
  str += '<li>A</li>';
  str += '<li>B</li>';
  str += '<li>C</li>';
  str += '</ul>';

  document.write(str);
</script>
```

- A
- B
- C

## ■ 변수

### ● 복합 대입 연산자

```
<html>
<head>
<script>
  var add = function() {
    var tag = "";
    tag += "<ul>";
    tag += "<li>Test</li>";
    tag += "</ul>";
    document.body.innerHTML += tag;
  };
</script>
</head>

<body>
  <input type="button" value="Tag Change" onclick="add()">
</body>
</html>
```

Tag Change

- Test

## ■ 변수

### ● 증감 연산자

- 복합 대입 연산자를 간략하게 사용한 형태

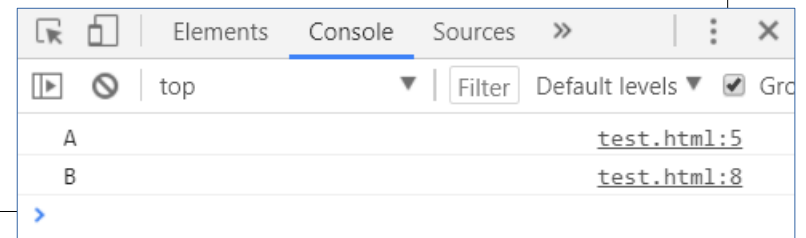
| 연산자  | 설명               |
|------|------------------|
| 변수++ | 기존 변수에 1을 더함(후위) |
| ++변수 | 기존 변수에 1을 더함(전위) |
| 변수-- | 기존 변수에 1을 뺌(후위)  |
| --변수 | 기존 변수에 1을 뺌(전위)  |

- 한줄에 독립적으로 사용할 때는 전위, 후위가 차이가 없음

## ■ 변수

### ● 변수 재선언

```
<script>  
  var name = 'A';  
  console.log(name);  
  
  var name = 'B';  
  console.log(name);  
</script>
```



## ■ 변수

### ● 변수 재선언

#### – 기존에 사용하던 식별자 재선언

```
<html>
<head>
<script>
  alert('출력');

  var alert = '재선언';

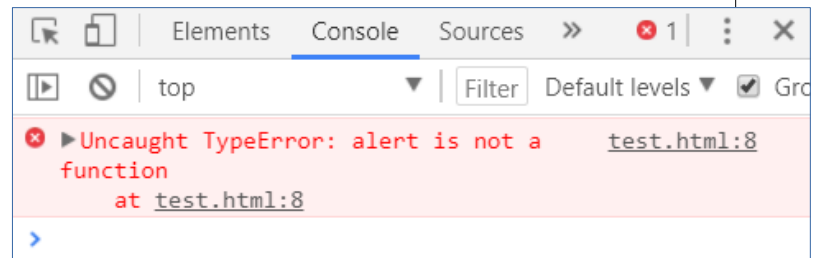
  alert('??');
</script>
</head>

<body>
</body>
</html>
```

이 페이지 내용:

출력

확인



## ■ 자료형

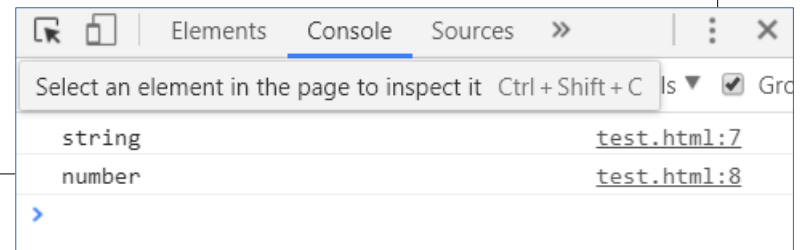
### ● typeof 연산자

– 자료형을 확인할 때 사용

```
<html>
<head>
<script>
  var str = typeof('string');
  var num = typeof(1234);

  console.log(str);
  console.log(num);
</script>
</head>

<body>
</body>
</html>
```



## ■ 자료형

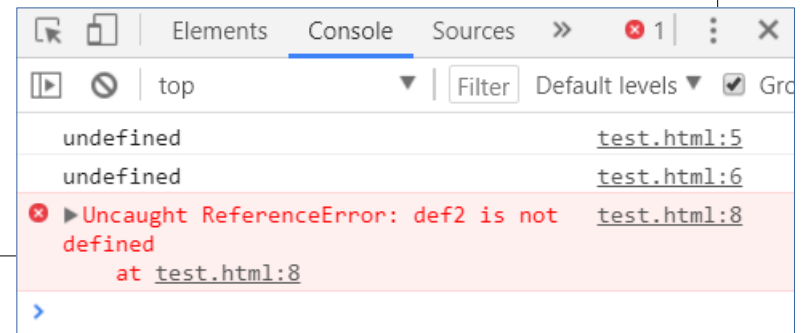
### ● typeof 연산자

– undefined

```
<html>
<head>
<script>
  var def;
  console.log(typeof(def));
  console.log(typeof(def2));

  console.log(def2);
</script>
</head>

<body>
</body>
</html>
```

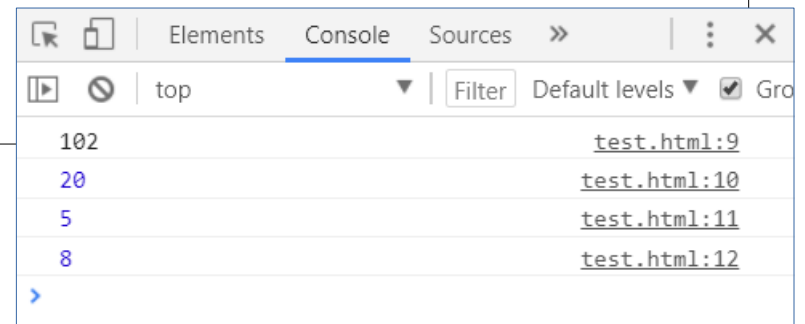


## ■ 숫자 / 문자열 자료형 변환

### ● 더하기 연산을 제외한 사칙 연산은 항상 숫자 우선

```
<script>
  var result1 = 10 + '2';
  var result2 = '10' * '2';
  var result3 = '10' / '2';
  var result4 = '10' - '2';

  console.log(result1);
  console.log(result2);
  console.log(result3);
  console.log(result4);
</script>
```





## ■ 숫자 / 문자열 자료형 변환

### ● prompt()로 입력받은 값을 숫자 변환 후 연산 - 1

```
<script>
  var num1 = prompt('첫번째 수를 입력해주세요.', '');
  var num2 = prompt('두번째 수를 입력해주세요.', '');

  var sum = (num1 * 1) + (num2 * 1);

  document.write('<h2>두 수의 합은 => ' + sum + '</h2>');
</script>
```

이 페이지 내용:

첫번째 수를 입력해주세요.

이 페이지 내용:

두번째 수를 입력해주세요.

**두 수의 합은 => 2324**

## ■ 숫자 / 문자열 자료형 변환

### ● prompt()로 입력받은 값을 숫자 변환 후 연산 - 2

```
<script>
  var num1 = prompt('첫번째 수를 입력해주세요.', '');
  var num2 = prompt('두번째 수를 입력해주세요.', '');

  var sum = parseInt(num1) + parseInt(num2);
  document.write('<h2>두 수의 합은 => ' + sum + '</h2>');
</script>
```

이 페이지 내용:

첫번째 수를 입력해주세요.

이 페이지 내용:

두번째 수를 입력해주세요.

**두 수의 합은 => 2324**

## ■ 숫자 / 문자열 자료형 변환

### ● prompt()로 입력받은 값을 숫자 변환 후 연산 - 3

```
<script>
  var num1 = prompt('첫번째 수를 입력해주세요.', '');
  var num2 = prompt('두번째 수를 입력해주세요.', '');

  var sum = Number(num1) + Number(num2);
  document.write('<h2>두 수의 합은 => ' + sum + '</h2>');
</script>
```

이 페이지 내용:

첫번째 수를 입력해주세요.

이 페이지 내용:

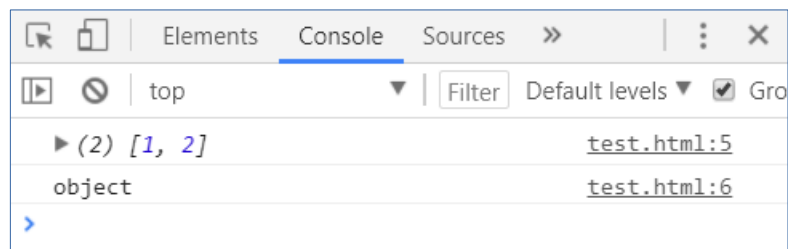
두번째 수를 입력해주세요.

**두 수의 합은 => 2324**

## ■ 배열

- 여러 개의 변수를 한꺼번에 다룰수 있는 자료형
- 객체 중 하나
- 대괄호를 사용해서 생성하고 쉼표로 구분해서 자료를 입력

```
<script>  
  var array = [1, 2];  
  console.log(array);  
  console.log(typeof(array));  
</script>
```

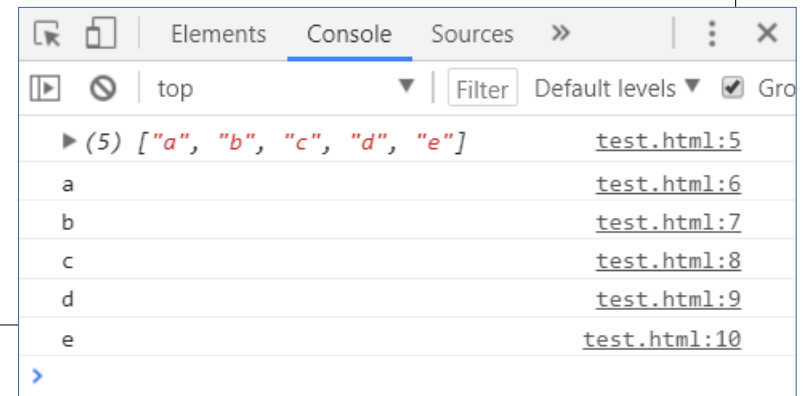


## ■ 배열

### ● 저장되어 있는 요소 출력

- 인덱스를 이용하여 배열에 저장된 요소 접근
- 배열의 첫번째 요소는 인덱스 0

```
<script>  
  var array = ['a', 'b', 'c', 'd', 'e'];  
  console.log(array);  
  console.log(array[0]);  
  console.log(array[1]);  
  console.log(array[2]);  
  console.log(array[3]);  
  console.log(array[4]);  
</script>
```

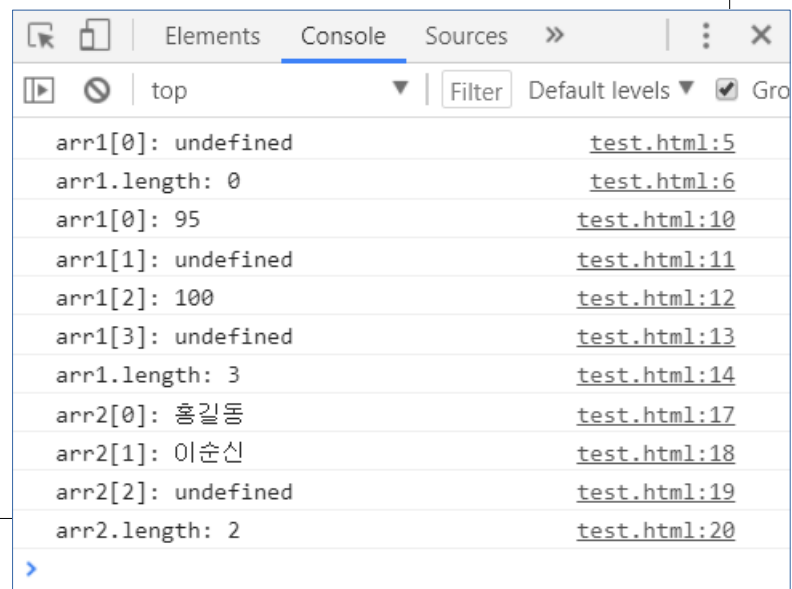


## ■ 배열 사용 - 1 (기본)

```
<script>
  var arr1 = [];
  console.log('arr1[0]: ' + arr1[0]);
  console.log('arr1.length: ' + arr1.length);

  arr1[0] = 95;
  arr1[2] = 100;
  console.log('arr1[0]: ' + arr1[0]);
  console.log('arr1[1]: ' + arr1[1]);
  console.log('arr1[2]: ' + arr1[2]);
  console.log('arr1[3]: ' + arr1[3]);
  console.log('arr1.length: ' + arr1.length);

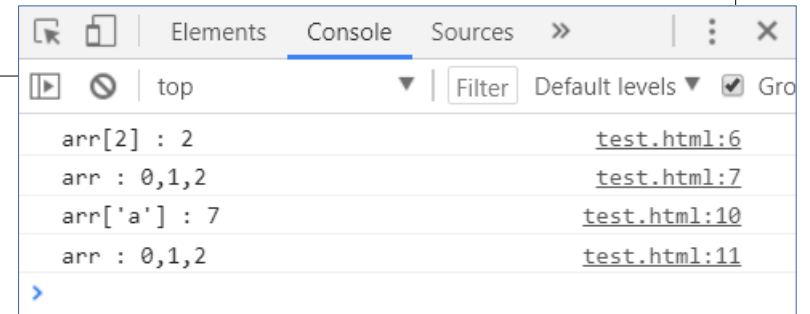
  var arr2 = [ '홍길동', '이순신' ];
  console.log('arr2[0]: ' + arr2[0]);
  console.log('arr2[1]: ' + arr2[1]);
  console.log('arr2[2]: ' + arr2[2]);
  console.log('arr2.length: ' + arr2.length);
</script>
```



## ■ 배열 사용 - 2 (push / 문자 인덱스)

```
<script>
  var arr = [0, 1];
  arr.push(2); // 현재까지 입력된 [숫자]인덱스 다음 요소로 추가
  console.log("arr[2] : " + arr[2]); // 2번 인덱스 접근
  console.log("arr : " + arr); // 모든 배열요소 출력

  arr['a'] = 7; // [숫자]인덱스가 아닌 [문자]인덱스를 이용하여 배열 요소 입력
  console.log("arr['a'] : " + arr['a']); // a번 인덱스 접근
  console.log("arr : " + arr); // 모든 배열요소 출력 (문자인덱스 제외)
</script>
```



## ■ 배열 사용 - 3 (pop)

```
<script>
  var arr = [0, 1, 2];

  var last = arr.pop(); // 마지막 요소 추출
  console.log("last : " + last);

  console.log("arr[1] : " + arr[1]); // 1번 인덱스 확인
  console.log("arr : " + arr); // 모든 배열요소 출력
</script>
```

