

JS : 함수와 배열

JS 함수와 배열

- ◆ 함수
- ◆ 배열
- ◆ 배열 관련 메소드
- ◆ 2차원 배열

예제 10-1 기본 함수 호출하기

```
<script>
  var text1="함수 선언 전 호출";
  var text2="함수 선언 후 호출";

  printMsg(text1);           // 함수 선언 전 호출

  function printMsg(msg) {
    document.write("함수 호출 메시지 : " + msg + "<br>");
  }

  printMsg(text2);           // 함수 선언 후 호출
</script>
```

Hoisting : ...

예제 10-2 onclick 속성값으로 함수 호출하기

```
<script>
    function printMsg(name, age) {
        document.write("학생 이름 : <b>" + name + "</b><br>");
        document.write("학생 나이 : <b>" + age + "</b><br>");
    }
</script>

<button type="button" onclick="printMsg('홍길동', 21)" >학생 정보</button>
```

무명 함수

```
var 변수명=function(매개 변수1, 매개 변수2, ...) { // 함수 선언
    실행 문장;
}
변수명(인자1, 인자2, ...); // 함수 호출
```

```
var x = function (a, b) {return a * b};
var z = x(4, 3);
```

예제 10-3 무명 함수 호출하기

```
<script>
  var text1="함수 선언 전 호출 예러";
  var text2="함수 선언 후 호출만 가능";
  // printMsg(text1);           // 함수 선언 전 호출 예러
  var printMsg = function(msg) {
    document.write("함수 호출 메시지 : " + msg + "<br>");
  }
  printMsg(text2);
</script>
```

예제 10-4 기본 함수와 무명 함수 호출 우선순위 살펴보기

```
<script>
    var printMsg=function(msg) {    // 무명 함수 선언
        document.write("무명 함수 : " + msg + "<br>");
    }
    function printMsg(msg){        // 기본 함수 선언
        document.write("기본 함수 : " + msg + "<br>");
    }
    printMsg("호출되었습니다.");
</script>
```

예제 10-5 변수를 이용하여 반환값 출력하기

```
<script>
  var result;
  function add(name, n) {
    document.write(name + " 학생이 1부터 " + n + "까지 덧셈 수행<br>");
    var sum=0;
    for(var i=1; i<=n; i++) {
      sum+=i;
    }
    return sum;
  }
  result=add('홍길동', 10);
  document.write("결과 : " + result + "<p/>");
  result=add('이영희', 100);
  document.write("결과 : " + result + "<p/>");
</script>
```


예제 10-7 서로 다른 변수로 같은 함수의 반환값 출력하기

```
<script>
    function add(x, y) {
        return x+y;
    }
    var calSum = add;
    var addUp = add;

    document.write("결과 값 : " + calSum(5, 10) + "<br>");
    document.write("결과 값 : " + addUp(3, 20) + "<br>");
</script>
```

예제 10-8 서로 다른 변수로 같은 함수의 반환값 출력하기

```
<script>
```

```
function add() {  
    var sum=1;  
    return sum;  
}  
function add(x) {  
    var sum=x+1;  
    return sum;  
}  
function add(x, y) {  
    var sum=x+y;  
    return sum;  
}  
function add(x, y, z) {  
    var sum=x+y+z;  
    return sum;  
}
```

```
var r0=add();  
var r1=add(1);  
var r2=add(2, 3);  
var r3=add(4, 5 ,6);  
var r4=add(7, 8, 9, 10);  
document.write("함수 호출 인자 없음 : " + r0 + "<p/>");  
document.write("함수 호출 인자 부족 : " + r1 + "<p/>");  
document.write("함수 호출 인자 부족 : " + r2 + "<p/>");  
document.write("정상적인 함수 호출 : " + r3 + "<p/>");  
document.write("7, 8, 9만 인자값으로 적용 : " + r4 + "<p/>");
```

```
</script>
```

함수 호출 인자 없음 : NaN

함수 호출 인자 부족 : NaN

함수 호출 인자 부족 : NaN

정상적인 함수 호출 : 15

7, 8, 9만 인자값으로 적용 : 24

예제 10-9 인자의 개수가 적을 때 처리 방법 살펴보기

```
<script>
function add(x, y, z) {
    var sum;
    if((y===undefined) && (z===undefined)) {
        sum=x;
    }
    else if(z===undefined) {
        sum=x+y;
    }
    else {
        sum=x+y+z;
    }
    return sum;
}
var r1=add(2);
var r2=add(2, 3);
var r3=add(4, 5, 6);

document.write("함수 호출 인자 부족 : " + r1 + "<p/>");
document.write("함수 호출 인자 부족 : " + r2 + "<p/>");
document.write("정상적인 함수 호출 : " + r3 + "<p/>");
</script>
```

예제 10-10 인자를 <<arguments 객체>>>로 처리하기

```
<script>
  function add() {
    var i, sum=0;
    for(i=0; i<arguments.length; i++) {
      sum=sum+arguments[i];
    }
    document.write("수행 결과 : " + sum + "<p/>");
  }

  add(2, 3);
  add(2, 3, 4);
  add(4, 5, 6, 7, 8);
  add(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
</script>
```

JS 함수와 배열

- ◆ 함수
- ◆ **배열**
- ◆ 배열 관련 메소드
- ◆ 2차원 배열

배열 생성 I

◆ 배열 리터럴로 생성

```
var 배열명=[원소1, 원소2, 원소3, ... ];
```



그림 10-3 배열의 구조

예제 10-11 배열 변수에 초깃값을 할당하여 배열 만들기

```
<script>
    var city=["Seoul","Busan","Incheon"];    // 배열 리터럴
    function printArr() {
        var i;
        for(i=0; i<city.length; i++) {
            document.write("배열 데이터["+ i + "] = " + city[i] + "<br>");
        }
    }
    printArr();
</script>
```

예제 10-13 배열에 공백 데이터 포함하기

```
<script>
    var city=["Seoul", , "Busan", , "Incheon"];    // 공백 리터럴 포함
    function printArr() {
        var i;
        for(i=0; i<city.length; i++) {
            document.write("배열 데이터 [" + i + "] = " + city[i] + "<br>");
        }
    }
    printArr();
</script>
```

```
배열 데이터 [0] = Seoul
배열 데이터 [1] = undefined
배열 데이터 [2] = Busan
배열 데이터 [3] = undefined
배열 데이터 [4] = Incheon
```


예제 10-15 공백 데이터 제외하고 연산하기

```
<script>
    var com=[95, 88, ,72 ,68, ,99 ,82 ,78, 87];    // 10명 중 8명의 점수만 입력
    var getAvg ;
    function printArr() {
        var i;
        var sum=0;
        var count=0;
        var n=com.length;
        document.write( n + "명의 점수 입력<p/>");
        for(i=0; i<n; i++) {
            if(com[i]==undefined) {
                continue;
            }
            else {
                sum+=com[ i];
                count++
            }
        }
        document.write("점수를 입력한 학생 : " + count + "명<p/>");
        document.write("총합 : " + sum + "<p/>");
        return (sum/count);
    }
    getAvg=printArr();
    document.write("평균 : " + getAvg + "<p/>");
</script>
```

예제 10-16 배열에 <<<다양한 데이터 타입>>>을 가진 원소 저장하기

```
<script>
  var x=5;
  var arr=[100, "Seoul", true, x];
  function printArr() {
    var i;
    for(i=0; i<arr.length; i++) {
      document.write("배열 데이터 [" + i + "] = " + arr[i] + "<br>");
    }
  }
  printArr();
</script>
```

예제 10-18 다른 데이터 타입을 가진 배열 연산 -- 주의

```
<script>
    var arr=[10, 20, 30, 40, '50'];    // 다른 데이터 타입 요소
    function printArr() {
        var i, sum=0;
        for(i=0; i<arr.length; i++) {
            sum+=arr[i];
        }
        return sum;
    }
    var result=printArr();
    document.write("배열 원소 합 : " + result + "<br>");
</script>
```

배열 생성 II

◆ 배열 생성자

```
var 배열명=new Array(원소1, 원소2, 원소3, ... );
```

예제 10-19 배열 객체 생성하기

```
<script>
  var city=new Array("Seoul","Busan","Incheon");
  function printArr() {
    var i;
    for(i=0; i<city.length; i++) {
      document.write("배열 데이터 ["+ i + "] = " + city[i] + "<br>");
    }
  }
  printArr();
</script>
```

◆ 배열 객체 확인

방법	사용 예	결과
타입 확인 연산자인 typeof 사용	typeof city	object
배열 객체의 메소드인 isArray() 사용	Array.isArray(city)	true
Array 생성자의 연산자인 instanceof 사용	city instanceof Array	true

예제 10-20 배열 객체 생성 확인하기

```
<script>
var city=new Array("Seoul","Busan","Incheon");
function printArr() {
    if(city instanceof Array) {
        document.write("배열 객체가 생성되었습니다.<p/>");
        var i;
        for(i=0; i<city.length; i++) {
            document.write("배열 데이터 [" + i + "] = " + city[i] + "<br>");
        }
    }
    else {
        document.write("배열 객체가 아닙니다.<br>");
        document.write("데이터 : " + city + "<br>");
    }
}
printArr();
document.write("<p/> city 변수 타입 : " + typeof city + "<br>");
document.write("배열 객체 확인 결과 : " + Array.isArray(city) + "<br>");
</script>
```

예제 10-21 배열에 1부터 100까지 저장한 후 모두 더하기

← → ↺ 🏠 ⓘ localhost:8080/WebClass/ch10/21_arr.html

배열 생성/조회/연산

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99 100

배열 데이터 덧셈 연산 결과 : 5050

[돌아가기](#)

← → ↺ 🏠 ⓘ localhost:8080/WebClass/ch10/21_arr.html

배열 생성/조회/연산

예제 10-21 배열에 1부터 100까지 저장한 후 모두 더하기

```
<script>
    var arrdata=[];
    function insertArr() {
        var i=0;
        for(i=0; i<=99; i++) {
            arrdata[i]=i+1;
        }
        selectArr();
    }
    function selectArr() {
        var i;
        for(i=0; i<arrdata.length; i++) {
            document.write(arrdata[i] + " ");
        }
        addArr();
    }
    function addArr() {
        var i;
        var sum=0;
        for(i=0; i<arrdata.length; i++) {
            sum+=arrdata[i];
        }
        document.write("<p/> 배열 데이터 덧셈 연산 결과 : " + sum + "<p/>");
        document.write("<a href='21_arr.html'>돌아가기</a>");
    }
}
</script>
<button type="button" onclick="insertArr()">배열 생성/조회/연산</button>
```

← → ↺ 🏠 ⓘ localhost:8080/WebClass/ch10/21_arr.html

배열 생성/조회/연산

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99 100

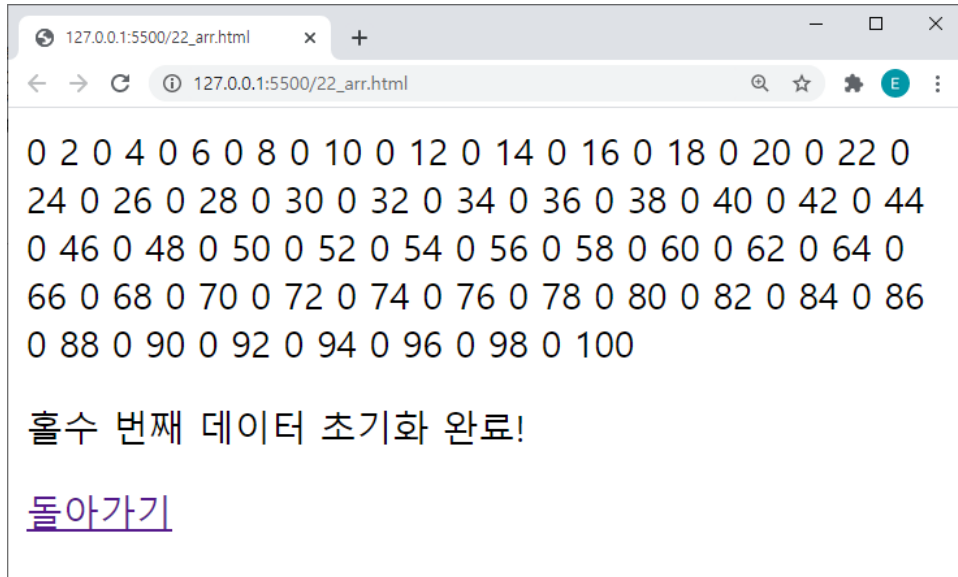
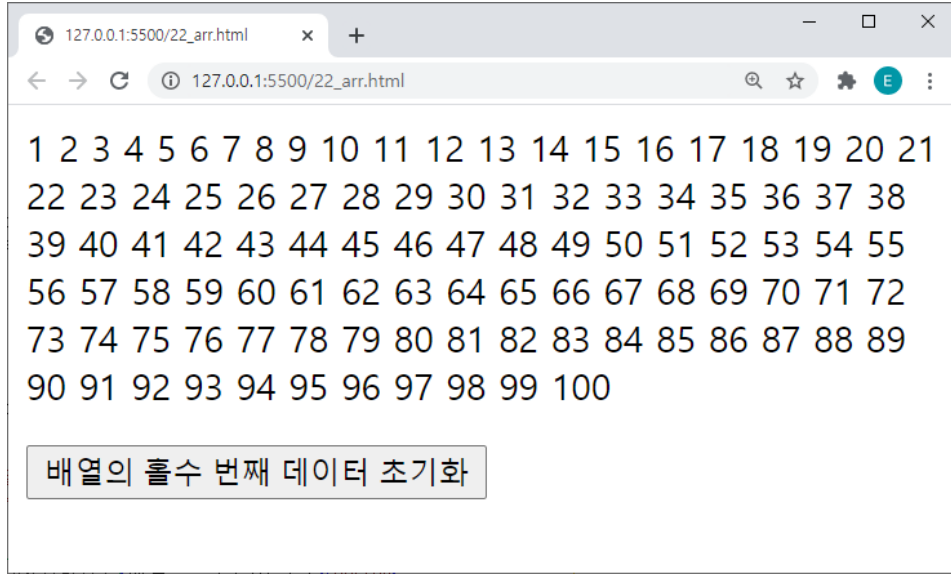
배열 데이터 덧셈 연산 결과 : 5050

[돌아가기](#)

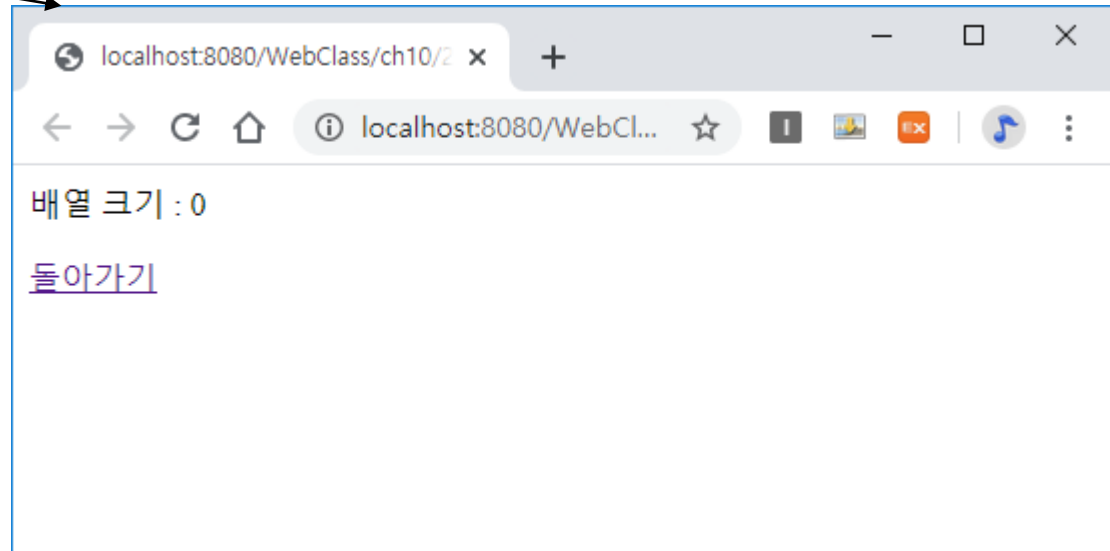
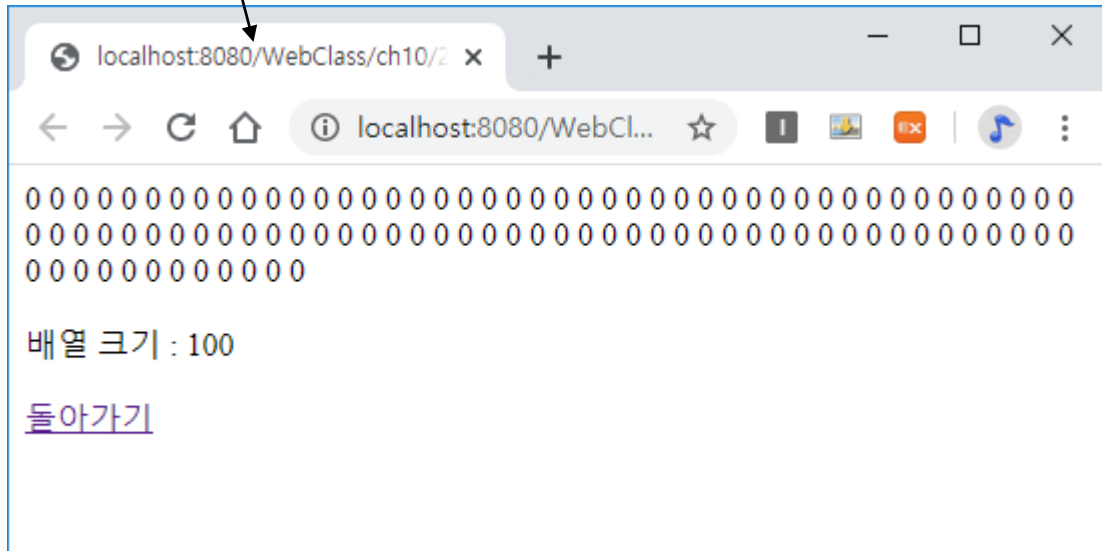
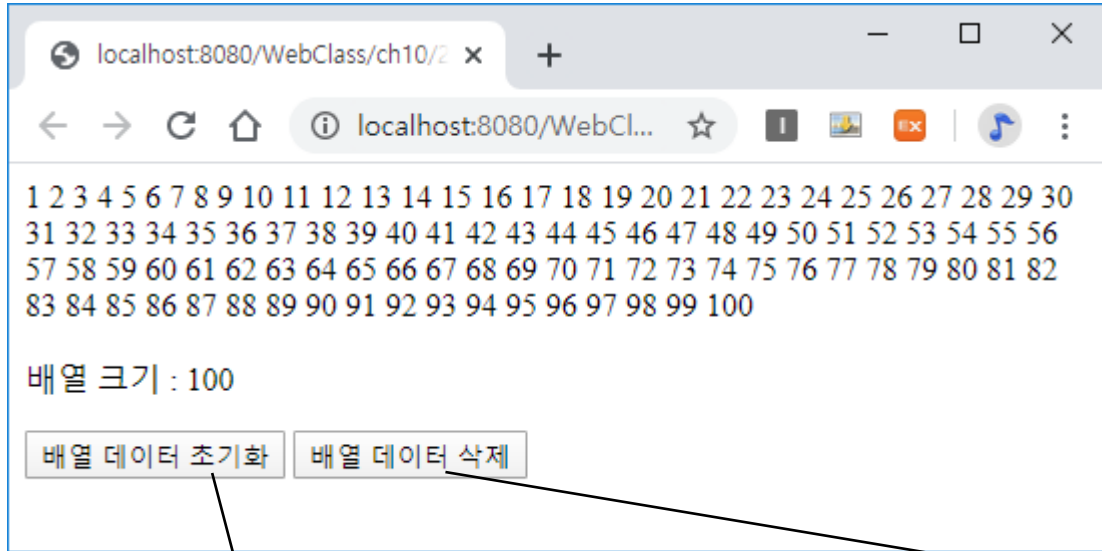
← → ↺ 🏠 ⓘ localhost:8080/WebClass/ch10/21_arr.html

배열 생성/조회/연산

예제 10-22 홀수 번째 저장된 데이터만 0으로 초기화하기



예제 10-23 배열에 저장된 데이터 삭제하기



JS 함수와 배열

- ◆ 함수
- ◆ 배열
- ◆ 배열 관련 메소드
- ◆ 2차원 배열