

5장 함수

5.1 함수

5.1.1 함수란?

- 함수란 일련의 실행문을 저장할 수 있는 공간을 가리킨다. 함수에 저장된 일련의 실행문은 함수 호출문에 의해 언제든지 실행할 수 있다.

5.1.2 기본 함수 정의문

```
// 기본형
function 함수명(){ //함수 정의문
    일련의 실행문;
}
함수명(); //함수 호출문

function test(){
    document.write("hello");
}
test();
```

[실습파일: function_1_test.html, 완성파일: function_1.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>함수</title>
<script type="text/javascript">
    //
    var i = 0;
    function myFnc() { // myFnc함수 정의문
        i++;
        document.write("hello" + i, "&lt;br /&gt;");
    }

    myFnc(); //myFnc 함수 호출

    var theFnc = function() { //익명 함수를 변수 theFnc에 참조 시킴.
        i++;
        document.write("bye" + i, "&lt;br /&gt;");
    }

    theFnc(); //변수 theFnc에 참조된 함수를 호출
    //]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="125 845 527 859" data-label="Text"><p>[실습파일: function_2_test.html, 완성파일: function_2.html]</p></div><div data-bbox="125 870 825 908" data-label="Text"><pre>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"&gt;
&lt;html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko"&gt;</pre></div>
```

```
function colorBg() {
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>함수</title>
<script type="text/javascript">
    //
        function myFnc(name, area) {
            document.write("안녕하세요. " + name + "입니다.", "&lt;br /&gt;");
            document.write("사는곳은 " + area + "입니다.", "&lt;br /&gt;&lt;br /&gt;");
        }

        // "홍길동"은 name에 "서울"은 area에 전달합니다.
        myFnc("홍길동", "서울");
    ]&gt;
&lt;/script&gt;
&lt;/html&gt;</pre>
```

```

        // "김부자"는 name에 "부산"은 area에 전달합니다.
        myFnc("김부자", "부산");
    //]]>
</script>
</head>
<body>
</body>
</html>

```

[실습파일: function_4_test.html, 완성파일: function_4.html]

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>함수</title>
<script type="text/javascript">
    //
        function login(id, pw) {
            if (id == "hanguk") { //방문자가 입력한 id가 "hanguk"과 일치할 경우

                if (pw == "1234") { //방문자가 입력한 pw가 "1234"와 일치할 경우
                    document.write(id + "님 방문을 환영합니다.");
                } else { //비밀번호가 일치하지 않을 경우
                    alert("잘못된 비밀번호입니다.")
                }

            } else { //방문자가 입력한 id가 일치하지 않을 경우

                alert("존재하지 않는 아이디입니다.")

            }

        }

        /*질의응답 창을 이용해 방문자에게 아이디와
        비밀번호를 입력받습니다.*/
        var user_id = prompt("아이디를 입력하세요.", "");
        var user_pw = prompt("패스워드를 입력하세요.", "");

        //방문자가 입력한 비밀번호와 아이디를 함수의 매개변수에 전달합니다.
        login(user_id, user_pw);
    //]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>
</div>
<div data-bbox="114 728 273 747" data-label="Section-Header">
<h2>5.1.4 내장 함수</h2>
</div>
<div data-bbox="114 761 883 794" data-label="List-Group">
<ul>
<li>■ 내장 함수는 자바스크립트 엔진에 내장된 함수 정의문을 말한다. 함수 정의문의 선언 없이 단지 함수 호출만으로 자바스크립트에 이미 내장된 함수들을 사용할 수 있다.</li>
</ul>
</div>
<div data-bbox="116 806 862 900" data-label="Table">
<table>
<tr>
<th>종류</th><th>설명</th><th>사용 예</th></tr>
<tr>
<td>parseInt()</td><td>문자형 데이터를 정수형 데이터로 바꾼다.</td><td>parseInt("5.12") -&gt; 5</td></tr>
<tr>
<td>parseFloat()</td><td>문자형 데이터를 실수형 데이터로 바꾼다.</td><td>parseFloat("5.12") -&gt; 5.12</td></tr>
<tr>
<td>String()</td><td>문자형 데이터로 바꾼다.</td><td>String(5) -&gt; "5"</td></tr>
<tr>
<td>Number()</td><td>숫자형 데이터로 바꾼다.</td><td>Number("5") -&gt; 5</td></tr>
<tr>
<td>Boolean()</td><td>논리형 데이터로 바꾼다.</td><td>Boolean(5) -&gt; true</td></tr>
<tr>
<td>isNaN()</td><td>데이터에 숫자가 아닌 문자를 포함하면 true를 반환한다.</td><td>isNaN("5-3") -&gt; true</td></tr>
</table>
</div>
```

	// is Not a Number ?	isNaN("53") -> false
eval()	문자형 데이터를 큰따옴표가 없는 스크립트 코드로 처리한다.	eval("15+5") -> 20

[과제] 로또

- Math.random() 을 이용해서 1~45 사이의 난수를 6개 추출 하는 프로그램을 작성하세요.
(조건 1.중복 되지 않는 값을 구해야 함, 2.구해진 값을 오름차순으로 정렬해서 출력)

[실습파일: 5장_미션/lotto_test.html]

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> lotto </title>
<script>

function find(){

}

</script>
</head>

<body>
<form>
    <input type=button value= "추첨하기" onClick="find()">
</form>
</body>
</html>
```

5.2 함수에서 return 문의 역할

- 함수 정의문에 return문이 사용되면 반환과 정지에 의미를 담고 있다.

5.2.1 데이터를 반환하는 return문

```
// 기본형
function 함수명(){
    실행문;
    return 데이터값;
}
var 변수 = 함수명();

function calc(){
    var result = 100+200;
    return result;
}
var num = calc();
document.write(num);
```

[실행파일: function_7_test.html, 완성파일: function_7.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

```

<title>함수</title>
<script type="text/javascript">
    <![CDATA[
        function myFnc(a, b) {
            var num = a * b;
            return num;
        }

        var result = myFnc(10, 3);

        document.write(result);
    </script>
</head>
<body>
</body>
</html>

```

5.2.2 강제 종료 역할을 하는 return문

```

// 기본형
function 함수명(){
    실행문1;
    return;
    실행문2; //실행문2는 실행되지 않고 함수를 종료함.
}
함수명();

```

[실행파일: function_6_test.html, 완성파일: function_6.html]

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>함수</title>
<script type="text/javascript">
    <![CDATA[
        function myFnc() {
            document.write("hello");
            return; //함수를 강제 종료합니다.
            document.write("javascript")
        }

        myFnc();
    </script>
</head>
<body>
</body>
</html>

```

5.2.3 재귀 함수 호출

- 최초 함수를 호출하면 함수 정의문 내에서 실행문으로 함수를 다시 호출하는 것을 재귀 함수 호출이라 한다. 함수를 반복문처럼 여러 번 호출하기 위해 사용한다.

```

// 기본형

```

```
function myFnc(){
    실행문;
    myFnc();
}
myFnc();
```

[실습파일: function_8_test.html, 완성파일: function_8.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>함수</title>
<script type="text/javascript">
    <![CDATA[
        var num = 0;
        function testFnc() {
            num++; //num의 데이터가 1씩 증가됩니다.
            document.write(num, "<br />");
            if (num == 10)
                return;

            //재귀함수 호출
            testFnc(); //testFnc 함수를 호출합니다.
        }

        testFnc(); //testFnc 함수를 호출합니다.
    ]]>
</script>
</head>
<body>
</body>
</html>
```

5.3 지역 변수와 전역 변수

- 지역변수는 함수 내에 var를 선언한 변수이며 함수 내에서만 사용할 수 있다. 그와 반대로 전역변수는 현재 페이지 어디서든 사용할 수 있다.

[실습파일: function_5_test.html, 완성파일: function_5.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>함수</title>
<script type="text/javascript">
    <![CDATA[
        var num = 200; //전역 변수
        function myFnc() {
            var num = 500; //지역 변수
        }

        myFnc(); //함수를 호출합니다.

        document.write(num); //전역 변수 num의 200이 출력됩니다.
    ]]>
</script>
</head>
<body>
</body>
```

```
</html>
```

- 함수에 실행문으로 var를 붙여 작성한 변수(var num)는 지역 변수이므로 함수의 밖에서는 불러와 사용할 수 없다.

[실습] 함수를 사용하여 사진 갤러리 만들기

1. 실습 파일[function_gallery_test.html]을 열어 아래와 같이 작성해 보자. 주석에 유의하여 코딩해 보자.

[실습파일: function_gallery_test.html, 완성파일: function_gallery.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> 사진 갤러리 </title>
<style type="text/css">
*{margin:0;padding:0;}
li{list-style:none;}
a{text-decoration:none;color:#000;}
#gallery{width:350px;margin:30px auto;
border:1px solid #999;
box-shadow:0 5px 5px rgba(0, 0, 0, 0.3);}
#gallery h1{text-align:center;}
#gallery img{border:1px solid #666;}
#photo{text-align:center;}
#gallery ul{width:800px;margin:0 auto;
overflow:hidden;}
#photo_list ul li{width:100px;float:left;text-align:center;}
#photo_list{width:300px;margin:10px auto;
/*border:1px dashed #000;*/overflow:hidden;}
.btn_wrap{text-align:center;}
.btn_wrap a{font-weight:bold;}
</style>
<script type="text/javascript">
//
window.onload=function(){
    /*① 포토 목록에 작은 이미지를 클릭했을 때
        큰 이미지를 바꿉니다. */

    /* 클릭 이벤트를 등록 하기위해서
        포토 목록에 8개의 &lt;a&gt;를 모두 선택합니다.*/
    var list_zone=document.getElementById("inner_list");
    var list_a=list_zone.getElementsByTagName("a");

    /*
        포토 리스트의 모든 &lt;a&gt;에 클릭 이벤트를 등록하기 위해서
        반복문을 이용하여 8개의 &lt;a&gt;에 모두 이벤트를 등록했습니다.
    */
    for(var i=0; i&lt;list_a.length; i++){
        list_a[i].onclick=function(){
            var ph=document.getElementById("photo").children[0];
            ph.src=this.href;
            return false; // &lt;a&gt;를 클릭했을 때 링크가 되지 않도록 합니다.
        }
    }

    /*② 이전, 다음 버튼을 클릭할 때 마다 &lt;ul&gt;이 100px만큼
        증가 또는 감소되어 좌·우측으로 이동됩니다. */

    /*다음 버튼을 클릭했을 때 마다 &lt;ul&gt;이 100px만큼</pre></div>
```

```

        좌측으로 이동해야 합니다.*/
        var b_btn=document.getElementById("next_btn");
        var m_num=0;
        b_btn.onclick=function(){
            if(m_num>=list_a.length-3) return false;
            m_num++;
            list_zone.style.marginLeft=-100*m_num+"px";

            return false; // <a>를 클릭했을 때 링크가 되지 않도록 합니다.
        }

        /*이전 버튼을 클릭했을 때 마다 <ul>이 100px만큼
        우측으로 이동해야 합니다.*/
        var n_btn=document.getElementById("before_btn");

        n_btn.onclick=function(){
            if(m_num<=0) return false;
            m_num--;
            list_zone.style.marginLeft=-100*m_num+"px";

            return false; // <a>를 클릭했을 때 링크가 되지 않도록 합니다.
        }
    }
//]]>
</script>
</head>
<body>
    <div id="gallery">
        <h1>동해안 갤러리</h1>
        <p id="photo">
            
        </p>
        <div id="photo_list">
            <ul id="inner_list">
                <li><a href="images/pic_1.jpg">
                    </a></li>
                <li><a href="images/pic_2.jpg">
                    </a></li>
                <li><a href="images/pic_3.jpg">
                    </a></li>
                <li><a href="images/pic_4.jpg">
                    </a></li>
                <li><a href="images/pic_5.jpg">
                    </a></li>
                <li><a href="images/pic_6.jpg">
                    </a></li>
                <li><a href="images/pic_7.jpg">
                    </a></li>
                <li><a href="images/pic_8.jpg">
                    </a></li>
            </ul>
        </div>
        <p class="btn_wrap">
            <a href="#" id="before_btn"><lt;</a>
            <a href="#" id="next_btn">>></a>
        </p>
    </div>
</body>
</html>

```

2. 작성한 코드를 실행한다.



[과제] 도전 Mission!

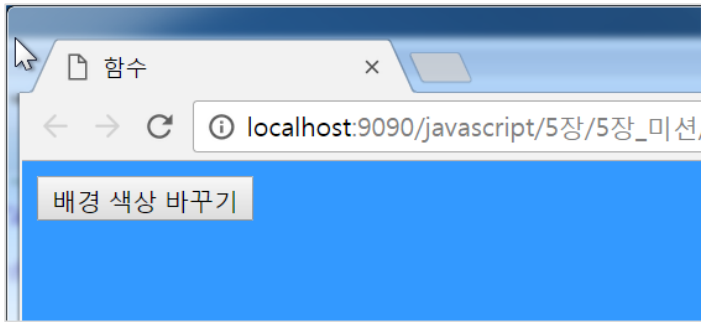
1. 다음 문제를 참고하여 코드를 완성하라.

버튼을 클릭할 때마다 함수가 실행되어 5개의 색상값 중 1개가 랜덤으로 선택되어 <body>에 적용되도록 만든다.

- 클릭할 때마다 chColor 함수가 호출되도록 한다.
- Math.random()과 배열의 데이터 개수(length) 속성을 이용하여 배열의 인덱스가 랜덤으로 나오도록 한다.
- 색상값을 요소명 선택자로 선택한 <body>에 스타일(CSS)을 적용한다.

[실습파일: 5장_미션/function_9_test.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> 함수 </title>
<script type="text/javascript">
//
function {
    var arrColor=["#ff0", "#6c0", "#fcf", "#cf0", "#39f"];
    var arrNum=Math.floor( );
    var bodyTag=document.getElementsByTagName("body")[0];
}
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;button onclick=""&gt;배경 색상 바꾸기&lt;/button&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
</div>
```



2. 다음 문제를 참고하여 코드를 완성해 보세요.

키와 몸무게의 값을 함수 매개 변수에 전달하여 적정 몸무게를 구한 후 적정 몸무게를 출력합니다. 그리고 몸무게가 적정 몸무게 이하인지 이상인지 출력되도록 합니다.

- ① 함수를 선언하고 키(h)와 몸무게(w)를 저장할 수 있도록 매개 변수를 작성하세요.
- ② 적정 몸무게 ($avg = (키 - 100) * 0.9$)
- ③ 만일 몸무게가 적정 몸무게 이하이면 "적정 몸무게 이하입니다"가 출력되고, 이상이면 "적정 몸무게 이상입니다"가 출력되도록 합니다.

[실습파일: 5장_미션/function_10_test.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> 함수 </title>
<script type="text/javascript">
//
{
    avg=
    document.write("평균 몸무게는"+avg+"kg 입니다. &lt;br /&gt;");

    if(){
        document.write("평균 몸무게 이하입니다");
    }else{
        document.write("평균 몸무게 이상입니다");
    }
}
avgWeight(180,74);
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;</pre></div>
```