

4강. 함수(function)



목 차

1

함수의 정의와 호출

2

함수의 유형

3

변수의 유효 범위

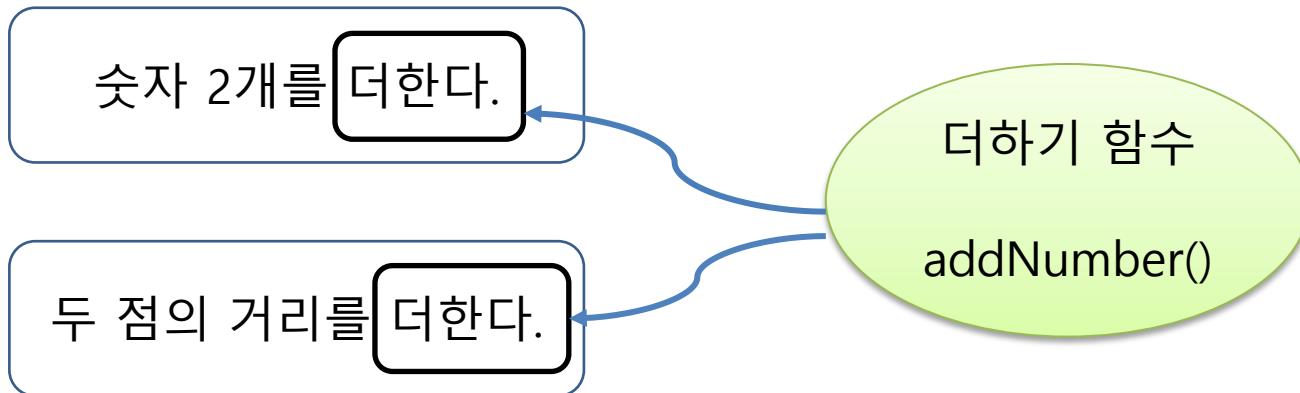
4

함수의 활용 - 이벤트 처리

함수의 정의

함수(Function)

- 특정한 하나의 기능을 수행하는 일련의 코드
- 중복되는 기능은 함수로 구현하여 그 함수를 호출하여 사용한다.



함수 만들기

함수 만들기(정의와 호출)

명시적 함수(이름있음)

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



함수 호출: 함수이름();

익명 함수(이름없음)

```
var 변수이름 = function(){  
    실행문;  
}
```



함수 호출: 변수이름();

함수의 유형

함수의 유형

- 매개변수 없는 경우

```
function sayHello(){ //함수 생성
    document.write("Hello~ <br>");
```

```
}
```

function1.html

```
sayHello(); //함수 호출
sayHello();
sayHello();
```

- 매개변수가 있는 경우

```
function sayHello(name){ //함수 생성
    document.write("Hello~ " + name + "<br>");
```

```
}
```

```
sayHello("샛별이"); //함수 호출
sayHello("Elsa");
sayHello("장그래");
```



함수의 유형

함수의 유형

- 익명함수로 만들기

```
<script>
    var hello = function(name){
        document.write("Hello~ " + name + '<br>');
    }
    hello('minsu');
    hello('younghee');
</script>
```

함수의 유형

함수의 유형

- 반환값이 있는 경우 – **return** 키워드 사용

```
function square(x){ //매개변수 1개
    return x * x;
}
```

func_return.html

```
function add(x, y){ //매개변수 2개
    var sum = x + y;
    return sum;
}
```

```
var num1 = square(4);
console.log("결과 : " + num1);
```

```
var num2 = add(10, 7);
console.log("결과 : " + num2);
```

함수의 정의와 호출

구구단 만들기

```
function gugudan(dan){  
    for(var i = 1; i <= 9; i++ ){  
        document.write(dan + " x " + i + " = " + (dan * i ) + "<br>");  
    }  
}
```

gugudan.html

```
/*var gugudan = function(dan){ //익명 함수  
    for(var i = 1; i <= 9; i++ ){  
        document.write(dan + " x " + i + " = " + (dan * i ) + "<br>");  
    }  
}*/
```

```
gugudan(6); //함수 호출
```

함수의 return 연습 문제

두 수를 매개변수 전달하여 서로 같으면 곱하고, 서로 다르면 더하는 함수를 정의하고 호출하는 프로그램을 작성하세요

```
<script>
    var data = function(x, y){ //매개변수 2개
        if(x==y){
            return x * y;
        }
        else{
            return x + y;
        }
    }

    var num1 = data(10, 10);
    var num2 = data(5, 10);

    document.write("num1 = " + num1 + '<br>');
    document.write("num1 = " + num2 + '<br>');
</script>
```

func_return2.html

전역 변수와 지역 변수

● 지역 변수(local variable)의 유효 범위(scope)

- 생성 : 함수 또는 제어문의 코드블럭(중괄호{ }) 내부에서 생성됨 – 지역, 매개변수
- 소멸 : **함수나 제어문의 영역{ }를 벗어났을 때** 메모리(스택)에서 해제(삭제)

```
<h3>지역변수의 유효 범위</h3>
<script>
    function oneUp(){
        let x = 1; //x는 지역 변수
        x = x + 1;
        return x;
    }

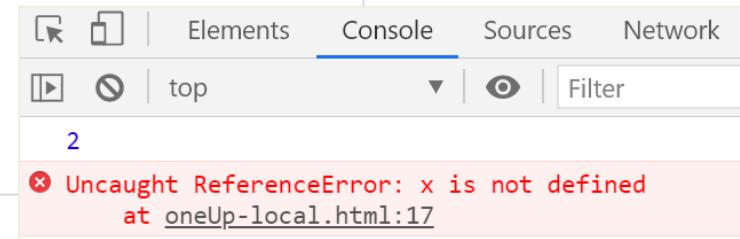
```

local_scope.html

```
let num = oneUp(); //호출되면 x는 2를 반환하고 소멸함
```

```
document.write("num = " + num); //2
document.write("x = " + x); //오류
```

```
</script>
```



전역 변수와 지역 변수

● 전역 변수(global variable)의 유효 범위

- 생성 : 메인 영역에서 생성하며 전체에 영향을 미치며, 값을 공유한다.
- 소멸 : 프로그램이 종료될 때 메모리(전역공간)에서 해제

```
let x = 1; //전역 변수  
  
function oneUp(){  
    x += 1;  
    return x;  
}  
  
let num = oneUp(); //1번 호출  
document.write("num = " + num + "<br>");  
  
num = oneUp(); //2번 호출  
document.write("num = " + num + "<br>");  
document.write("x = " + x); //x는 프로그램이 종료되면 소멸.
```

global_scope.html

전역 변수(Global Variable)의 유효 범위

```
num = 2  
num = 3  
x = 3
```



전역 변수와 지역 변수

메모리 영역 - 프로세스가 운영체제로 할당받은 메모리 영역 구분



let VS var

➤ 변수의 재선언과 재할당

let 키워드는 변수를 재선언하여 할당할 수 없다는 뜻이다.

var는 재선언이 가능하나, 오류 발생의 여지가 있다.

```
var x = 1;

if (x === 1) {
    x = x + 1;

    console.log(x);
}

var x = 3; //재선언 가능
console.log(x); //3
```

let x = 1;

if (x === 1) {
 x = x + 1;

 console.log(x);
}

let x = 3; //재선언 불가
console.log(x);

let_var.html

Identifier 'x' has already been declared

let VS var

➤ var의 호이스팅(hoisting)

변수를 선언하기전에 실행할 수도 있음.

이유 : 자바스크립트 해석기가 기억하면서 선언한 것과 같은 효과를 가짐

```
//코드는 위에서 아래로 순차적으로 해석됨.  
console.log(a);  
var a = 1;      //호이스팅(인터프리터가 나중에 선언해도 기억함)  
console.log(a)  
  
console.log(b);  
let b = 2;      //let은 호이스팅이 안됨.
```

let_var2.html

이벤트 다루기

이벤트 효과

웹페이지에서는 사용자가 메인 메뉴를 누르면 서브 메뉴가 펼쳐지기도 하고, 페이지 로딩이 끝나면 배경화면이 움직이기도 한다

마우스 이벤트

속성	설명
onclick	마우스를 눌렀을때 이벤트가 발생
onmouseover	마우스 포인터가 요소 위로 옮겨질때 이벤트 발생
onmousedown	마우스를 눌렀을때 이벤트가 발생

함수의 유형

onclick 속성 사용하기

```
<button onclick="fnCheck()">로그인</button>
```

```
<script>
    /*function fnCheck(){
        alert("아이디나 비밀번호가 틀립니다.");
    }*/

```

fn_check.html

```
//익명함수로 만들기
var fnCheck = function(){
    alert("아이디나 비밀번호가 틀립니다!!!");
}
</script>
```



getElementById() 사용

- ❖ **id** 선택자의 태그 요소에 접근하는 함수- **getElementById()**

속성	설명
innerHTML	태그 요소의 내용(텍스트)을 바꾸는 속성
style	태그 요소의 스타일을 바꾸는 속성
src	파일의 경로

getElementById() ^t-8

My Web Page

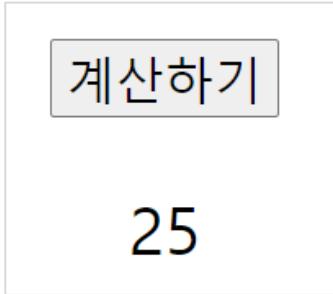
Hello Dolly.

How are you?

```
<h1>My Web Page</h1>  
<p id="myP">This is a p element.</p>                                         getId.html  
<div id="myDIV">This is a div element.</div>  
  
<script>  
document.getElementById("myP").innerHTML = "Hello Dolly.";  
document.getElementById("myDIV").innerHTML = "How are you?";  
</script>
```

getElementById() 사용

버튼에서 함수 호출하기



get_id2.html

```
<!-- 버튼에서 함수를 호출한 경우 : return 값이 없음 -->
<button onclick="square(7)">계산하기</button>
<p id="demo"></p>
<script>
    var square = function(x){
        var result = x * x;
        document.getElementById("demo").innerHTML = result;
    }
</script>
```

```
body{
    text-align: center;
}
button{
    margin-top: 40px;
    font-size: 1.3em;
}
p{font-size: 1.5em}
```

getElementById() 사용

나이 계산 프로그램.

127.0.0.1:5500 내용:
태어난 연도를 입력하세요.

확인 취소

나이를 계산하는 프로그램
계산하기
당신의 나이는 26세 입니다.

```
<div class="content">
    <h2>나이를 계산하는 프로그램</h2>
    <button onclick="calcAge()">계산하기</button>
    <hr>
    <p id="show"></p>
</div>

<script>
    function calcAge(){
        const CURRENT_YEAR = 2021;
        var birthYear = prompt("출생 연도를 입력하세요", "yyyy");

        var age = CURRENT_YEAR - birthYear + 1;

        document.getElementById("show").innerHTML =
            "당신의 나이는 " + age + "세 입니다.";
    }
</script>
```

calc_year.html

getElementById() 사용

- 버튼 클릭하면 1증가.

클릭하면 1증가

확인

4

```
<h3>클릭하면 1증가</h3>
<button onclick="oneUp()">확인</button>
<p id="demo"></p>
<script>
    let x = 0;

    function oneUp(){
        x += 1;
        document.getElementById('demo').innerHTML = x;
    }
</script>
```

getElementById() 사용

- 절대값 계산.

계산하기

5

```
<button onclick="myAbs(-5)">계산하기</button>
<p id="demo"></p>
<script>
    var myAbs = function(x){
        if(x < 0)
            document.getElementById("demo").innerHTML = -x;
        else
            document.getElementById("demo").innerHTML = x;;
    }
</script>
```

getElementById() 사용

- ✓ 마우스를 클릭하여 글자색을 바꾸기

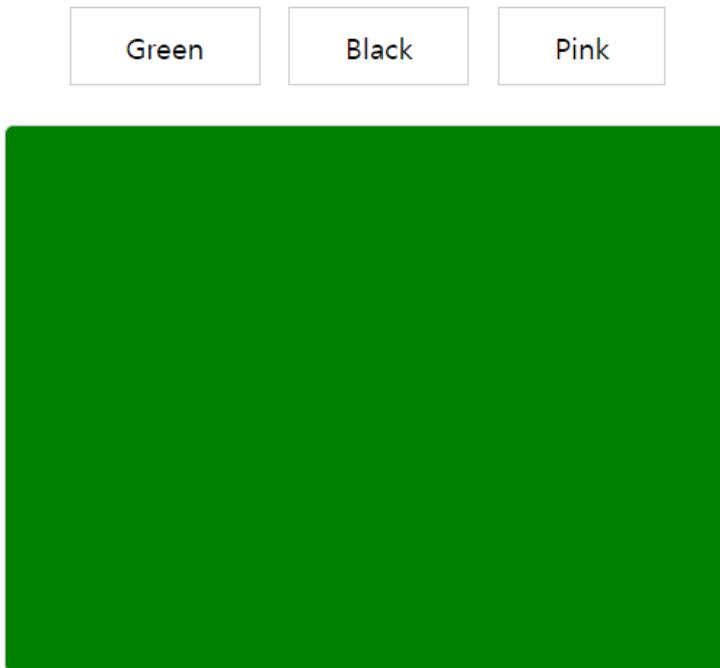
Click on this text! -----> **Click on this Text!!**
onclick

```
<h1 id="switch" onclick="changeText()">Click on this Text!!</h1>
<script>
    function changeText(){
        //document.getElementById("switch").innerHTML = "Good!"
        document.getElementById("switch").style.color = "blue";
    }
</script>
```

change_text.html

getElementById()의 style 속성

- 버튼을 클릭하여 배경색 넣기



Green

Black

Pink

버튼은 <a>태그로 만들기

Green을 클릭 -> 배경색 녹색

Black을 클릭 -> 배경색 검정색

Pink를 클릭 -> 배경색 분홍색

getElementById()의 style 속성

- 버튼을 클릭하여 배경색 넣기

```
<div class="menu">
  <ul>
    <li>
      |   <a href="#" onclick="changeBg('green')>Green</a>
    </li>
    <li>
      |   <a href="#" onclick="changeBg('black')>Black</a>
    </li>
    <li>
      |   <a href="#" onclick="changeBg('pink')>Pink</a>
    </li>
  </ul>
</div>
<div class="box" id="rect"></div>

<script>
  function changeBg(color){
    let rect = document.getElementById('rect')
    rect.style.backgroundColor = color;
  }
</script>
```

changebg.html



getElementById() 의 style 속성

```
.menu {width: 400px; margin: 30px 10px 10px 50px;}  
.menu ul{ list-style: none; padding: 0; text-align: center;}  
.menu ul li{  
    display: inline-block;  
    margin: 5px;  
    border: 1px solid #ccc;  
    padding: 10px 30px;  
}  
.menu ul li a{text-decoration: none; color: #000;}  
.box{  
    width: 400px;  
    height: 300px;  
    border: 1px solid #ccc;  
    border-radius: 5px;  
    margin-left: 50px;  
}
```

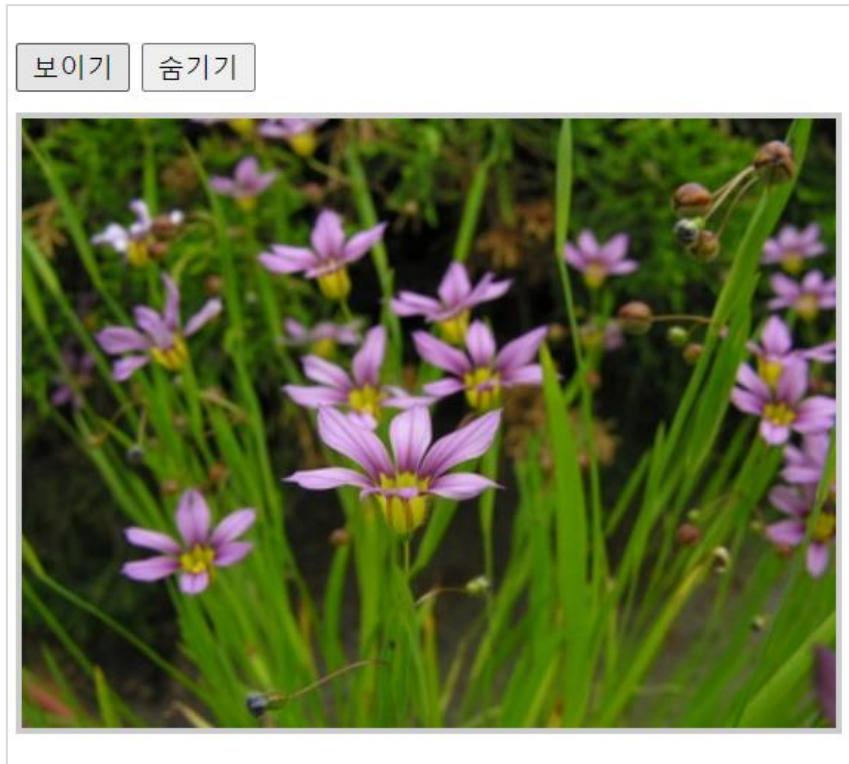
bg_color.css



getElementById()의 display 속성

- 이미지 숨기기/보이기

보이기 – **style.display = 'block'**, 숨기기 – **style.display = 'none'**



getElementById()의 display 속성

```
<button onclick="show()">보이기</button>
<button onclick="hide()">숨기기</button>          show_hide.html
<div id="item">
  
</div>

<script>
  function show(){ //이미지 보이기
    document.getElementById('pic').style.display = "block";
  }

  function hide(){ //이미지 숨기기
    document.getElementById('pic').style.display = "none";
  }
</script>
```

숨기기와 보이기 이벤트 효과

- 버튼을 클릭하여 이미지와 텍스트를 숨기고 보이는 효과 만들기

설명글 숨기기



설명글 보이기



민들레

어디서나 매우 흔하게 보이는 잡초로써 바닥에 딱 붙어서 꽃봉오리 하나가 쑥 올라온다 톱니 모양의 잎새와 눈에 확 띠는 노란 꽃이 인상적이다. 특히 꽃이 지고 나면 솜털모양의 깃을 가진 씨앗들이 나오는데 바람을 타고 날아가서 널리 퍼진다.

상세 설명 닫기

숨기기와 보이기 이벤트 효과

```
<div id="item">
    
    <button onclick="showText()" class="btn_show" id="show">상세 설명 보기</button>
    <div class="text" id="detail">
        <h4>민들레</h4>
        <p>어디서나 매우 흔하게 보이는 잡초로써 바닥에 딱 붙어서 꽃봉오리 하나가
           쑥 올라온다. 톱니 모양의 잎새와 눈에 확 띠는 노란 꽃이 인상적이다.
           특히 꽃이 지고 나면 솜털 모양의 깃을 가진 씨앗들이 나오는데 바람을
           타고 날아가서 널리 퍼진다.</p>
        <button onclick="hideText()" id="hide">상세 설명 닫기</button>
    </div>
</div>

<script src="js/event.js"></script>
```

show_hide2.html

숨기기와 보이기 이벤트 효과

```
/* flower.css */  
#item{width: 500px; margin: 0 auto; position: relative;}  
.open_btn{position: absolute; top : 260px; left : 20px;}  
.text{width: 400px; display: none; line-height: 1.5em;}
```

event.css

```
//상세 설명 보기  
function showText(){  
    //let text = document.getElementById("detail");  
    //text.style.display = "block"; //css의 style 속성 접근  
    document.getElementById("detail").style.display = "block"; //설명글 보기  
  
    //let btn = document.getElementById("show"); //보기 버튼 숨기기  
    //btn.style.display = "none";  
    document.getElementById('show').style.display = "none"  
}  
  
//상세 설명 닫기  
function hideText(){  
    document.getElementById("detail").style.display = "none"; //설명글 숨기기  
  
    document.getElementById('show').style.display = "block"; //보기 버튼 보기  
}
```

event.js

