

4강. 함수(메서드)



목 차

1

함수의 정의와 호출

2

함수의 유형

3

변수의 유효 범위

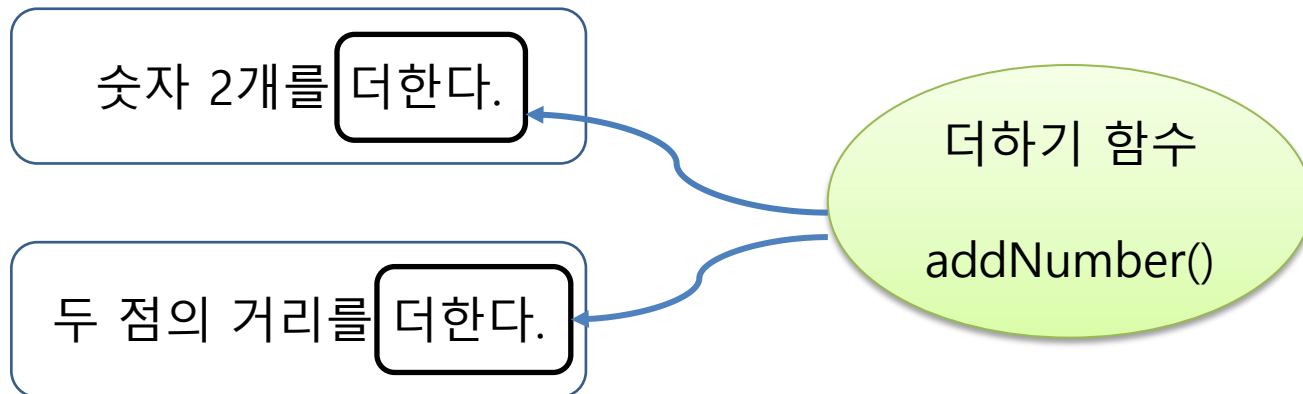
4

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>");
| }

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

- 매개변수가 있는 경우

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

sayHello("샷별이"); //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

```
<h2>return문 알아보기</h2>
<script>
  function addNumber(x, y){
    var add = x + y;
    return add;
  }

  // alert("두 수의 합 : " + addNumber(10, 20));
  var result = addNumber(10, 20);
  alert("두 수의 합 : " + result)
</script>
```

127.0.0.1:5500 내용:

두 수의 합 : 30

확인

함수 연습 문제

매개변수로 정수를 전달 받아 제곱수를 구하는 함수를 만들고 사용해 보세요

함수이름 : square

```
<h2>계산 결과</h2>
<script>
    var square = function(x){
        return x * x;
    }

    var result = square(4);

    document.write("4의 제곱 : " + result);
</script>
```


함수의 return 연습 문제

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

```
<script>
```

함수 코드 작성

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

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

```
</script>
```

전역 변수와 지역 변수

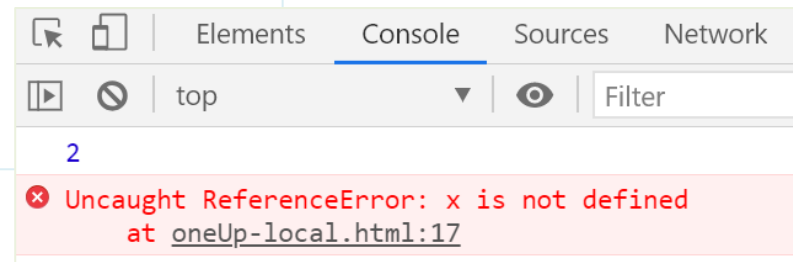
지역 변수(local variable)의 유효 범위(scope)

- 함수등 블록 안에서만 사용
- 소멸 시기 : **함수 호출후** 메모리(스택)에서 해제(삭제) 됨

```
<h3>지역변수의 유효 범위</h3>
<script>
  function oneUp(){
    var x = 1; //지역변수
    x++;
    return x;
  }

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

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



전역 변수와 지역 변수

메모리 영역 - 프로세스가 운영체제로 할당받은 메모리 영역 구분



전역 변수와 지역 변수

전역 변수(global variable)의 유효 범위

- 전체에 영향을 미치며, 값을 공유한다.
- 소멸 시기 : 프로그램이 종료되면 메모리(전역공간)에서 해제

```
<script>
  var x = 1;  //전역 변수
  function oneUp(){
    x += 1;
    return x;
  }

  var 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.
</script>
```

전역 변수(Global Variable)의 유효 범위

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

const(상수 키워드)

const

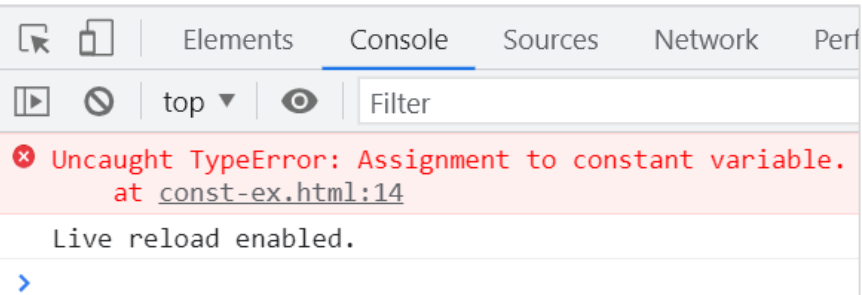
- const로 선언한 변수는 상수이다.
- 재선언하거나 재할당 할 수 없다.

```
<h2>상수 </h2>
<script>
  const PI = 3.14;

  //PI = 3;

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

상수



const(상수 키워드)

const

```
<h2>상수 </h2>
<script>
  const PI = 3.14;
  var radius = 5;
  var area;

  //PI = 3;
  //document.write(PI);

  //원의 넓이
  area = PI * radius * radius;
  document.write("원의 면적 : " + area);
</script>
```

let VS var

let

let 키워드는 변수를 재 선언하여 할당할 수 없다는 뜻이다.

var는 가능한데, 오류 발생의 여지가 있다. (w3schools – JS let 참조)

```
<script>
  var x = 1;

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

    console.log(x);
  }

  console.log(x);

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

```
<script>
  let x = 1;

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

    console.log(x);
  }

  console.log(x);

  let x = 3; //재선언 불가
  console.log(x);
</script>
```

let VS var

```
<h2>JavaScript let</h2>
```

```
<p id="demo"></p>
```

```
<script>
```

```
var i = 5;
```

```
for (var i = 0; i < 10; i++) {  
    // some statements  
}
```

```
document.getElementById("demo").innerHTML = i;
```

```
</script>
```

for문 밖의 변수 i가
재 선언됨

```
<h2>JavaScript let</h2>
```

```
<p id="demo"></p>
```

```
<script>
```

```
let i = 6;
```

```
for (let i = 0; i < 10; i++) {  
    //document.write("hello" + ' ');  
}
```

```
document.getElementById("demo").innerHTML = i;
```

```
</script>
```

for문 밖의 변수 i가
재 선언되지 않음

함수의 사용

❖ 문서 본문의 특정 위치에 출력하기

`document.getElementById("result").innerHTML`

속성	설명
getElementById()	id 선택자의 태그 요소에 접근하는 함수
innerHTML	태그 요소의 내용을 바꾸는 속성

```
<h1>My Web Page</h1>

<p id="myP">This is a p element.</p>

<div id="myDIV">This is a div element.</div>

<script>
document.getElementById("myP").innerHTML = "Hello Dolly.";
document.getElementById("myDIV").innerHTML = "How are you?";
</script>
```

My Web Page

Hello Dolly.

How are you?

이벤트 다루기

이벤트 효과

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

마우스 이벤트

속성	설명
<code>onclick</code>	마우스를 눌렀을때 이벤트가 발생
<code>onmouseover</code>	마우스 포인터가 요소 위로 옮겨질때 이벤트 발생
<code>onmousedown</code>	마우스를 눌렀을때 이벤트가 발생

함수의 사용

❖ 문서 본문의 특정 위치에 출력하기

```
<p id="demo" onclick="myFunction()">여기를 클릭하면 내용이 바뀝니다.</p>

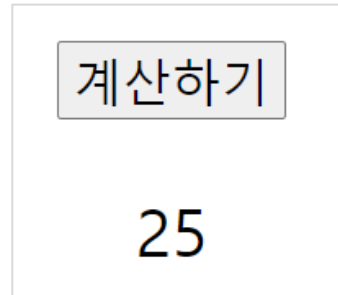
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed!";
}
</script>
```

```
<h2>계산 결과</h2>
<p id="demo"></p>
<script>
    function square(x){
        return x * x;
    }

    var result = square(5);
    document.getElementById('demo').innerHTML = result;
</script>
```

함수의 사용

버튼에서 함수 호출하기



```
<!-- 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}
```

const 상수

현재 년도는 2021로 선언하고 변경할 수 없게 만드세요.

127.0.0.1:5500 내용:
태어난 연도를 입력하세요.

나이를 계산하는 프로그램

계산하기

당신의 나이는 26세 입니다.

```
<div class="content">  
  <h2>나이를 계산하는 프로그램</h2>  
  <script>  
    function calc(){
```

코드 작성

```
      document.getElementById("show").innerHTML =  
        "당신의 나이는 " + age + "세 입니다.";  
    }  
  </script>  
  <button onclick="calc()">계산하기</button>  
  <hr>  
  <p id="show"></p>  
</div>
```

함수로 이벤트 처리하기

- 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>
```

함수로 이벤트 처리하기

- 이미지 숨기기/보이기



숨기기

보이기

```
.image{  
  margin: 10px;  
}  
button{  
  margin-left: 10px;  
  padding: 5px;  
}
```

```
<div class="image">  
    
</div>  
<button onclick="hide()">숨기기</button>  
<button onclick="show()">보이기</button>  
<script>  
  function hide(){  
    document.getElementById('img').style.display = "none";  
  }  
  function show(){  
    document.getElementById('img').style.display = "block";  
  }  
</script>
```

롤링 갤러리 만들기

롤링(rolling) 갤러리 만들기

이미지 넘기기



화살표 버튼을 클릭하면 이미지가 바뀜

```
<div id="wrapper">
  <h2>이미지 넘기기</h2>

  <a href="#" onclick="prevGallery()"></a>

  <a href="#" onclick="nextGallery()"></a>

</div>
```

```
#wrapper{
  text-align: center;
}
img{
  vertical-align: middle; /*top middle bottom*/
}
```


롤링 갤러리 만들기

롤링 갤러리 만들기

document.getElementById("gallery").src="이미지 파일 경로"

```
var num = 1;    //전역변수

function nextGallery(){
    num++;        //1 증가
    if(num > 7)    //최대 7보다 크면 다시 1로 초기화
        num = 1;
    document.getElementById("gallery").src="images/img" + num + ".jpg";
}

function prevGallery(){
    num--;        //1 감소
    if(num < 1)
        num = 7;
    document.getElementById("gallery").src="images/img" + num + ".jpg";
}
```

gallery.js

숨기기와 보이기 효과

- 버튼을 클릭하여 이미지와 텍스트를 숨기고 보이는 효과 만들기

설명글 보이기



설명글 숨기기



민들레

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

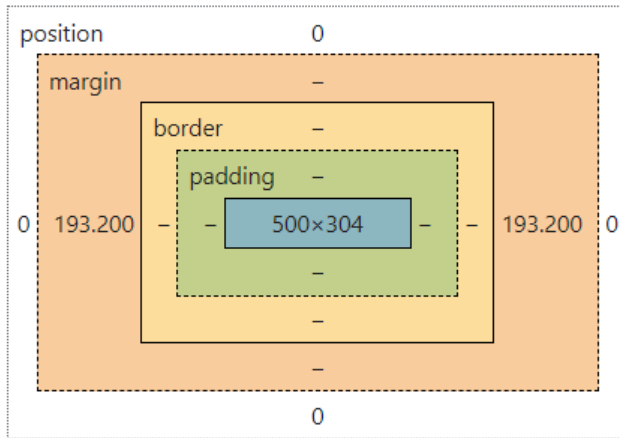
상세 설명 닫기

숨기기와 보이기 효과

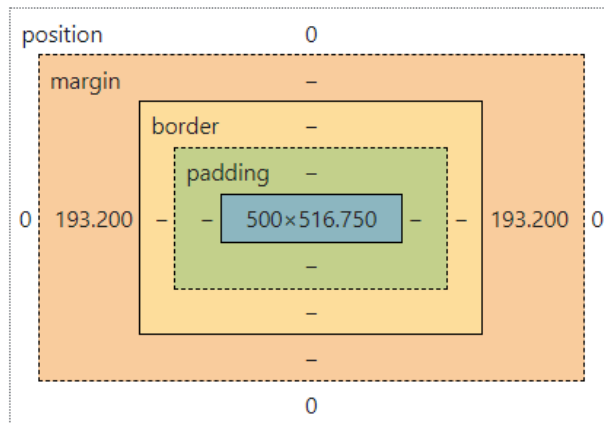
flower.html

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

숨기기와 보이기 효과



display: none



```
#content{  
  width: 500px;  
  margin: 0 auto;  
  position: relative;  
}  
.show{  
  position: absolute;  
  top: 260px;  
  left: 20px;  
}  
.text{  
  width: 400px;  
  display: none;  
  line-height: 1.5;  
}
```

숨기기와 보이기 효과

event.js

```
function showText(){
    //설명글 보이기
    document.getElementById('desc').style.display = "block";

    //보기 버튼 숨기기
    document.getElementById('open').style.display = "none";
}

function hideText(){
    //설명글 숨기기
    document.getElementById('desc').style.display = "none";

    //보기 버튼 보이기
    document.getElementById('open').style.display = "block";
}
```

할인가격 계산 프로그램

할인가격 계산 프로그램

text 박스에 입력 - `document.getElementById("result").value`



원래 가격 원

할인율 %

상품의 원래 가격은 10000원이고, 할인율은 20%입니다.

할인가격 계산 프로그램

```
<div id="contents">
  
  <ul>
    <li>
      <label for="oPrice">원래 가격</label>
      <input type="text" id="oPrice">원
    </li>
    <li>
      <label for="rate">할인율</label>
      <input type="text" id="rate">%
    </li>
    <li class="btn">
      <button onclick="calcPrice()">할인 가격 계산하기</button>
    </li>
  </ul>
  <div id="showResult"></div>
</div>
```

```
<script>
  function calcPrice(){
    var originPrice = 10000;
    var rate = 20;
    document.getElementById("showResult").innerHTML = "상품의 원래 가격은 " +
      originPrice + "원이고, 할인율은 " + rate + "%입니다.";
  }
</script>
```

할인가격 계산 프로그램

sale.html

```
#contents{
  width: 450px;
  margin: 0 auto;
}
img{
  float: left;
  margin-right: 25px;
}
ul{
  list-style: none;
  padding-top: 40px;
}
ul li{
  margin-bottom: 10px;
}
li > label{
  width: 80px;
  float: left;
  text-align: right;
}
```

```
input[type="text"]{
  width: 110px;
  margin: 0 5px 0 10px;
  padding: 5px 10px;
}
li:nth-child(3){
  margin-top: 20px;
  text-align: center;
}
button{
  padding: 5px 10px;
  font-size: 1em;
}
#showResult{
  margin-top: 20px;
  font-size: 1.3em;
}
```


할인가격 계산 프로그램

할인가격 계산 프로그램(완성)



원래 가격 원

할인율 %

할인 가격 계산하기

상품의 원래 가격은 20000원이고, 할인율은 30%입니다. 6000원을 절약한 14000원에 살 수 있습니다.

입력 코드 :
`document.getElementById("id").value`

할인금액 = 원래가격 x (할인율 ÷ 100)

최종금액 = 원래가격 - 할인금액

```
function calcPrice(){
    var originPrice = document.getElementById("oPrice").value; //원래 가격
    var rate = document.getElementById("rate").value;           //할인율
    var savedPrice = originPrice * (rate/100);                  //할인 금액
    var resultPrice = originPrice - savedPrice;                  //최종 가격

    document.getElementById("showResult").innerHTML = "상품의 원래 가격은 " +
        originPrice + "원이고, 할인율은 " + rate + "%입니다. " + savedPrice + "원을 절약한 " +
        resultPrice + "원에 살 수 있습니다.";
}
```