JavaScript

◆ 함수

정수아

Contents

01 함수

함수

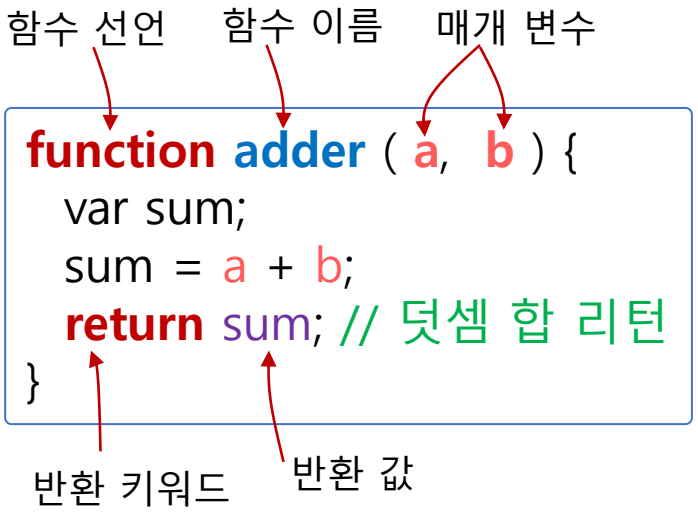
❖ 함수란?

- 목적을 가지고 작성된 코드 블록
- 데이터를 전달받아 처리한 후, 결과를 돌려주는 코드 블록

함수의 구성과 호출

❖ 함수의 구성

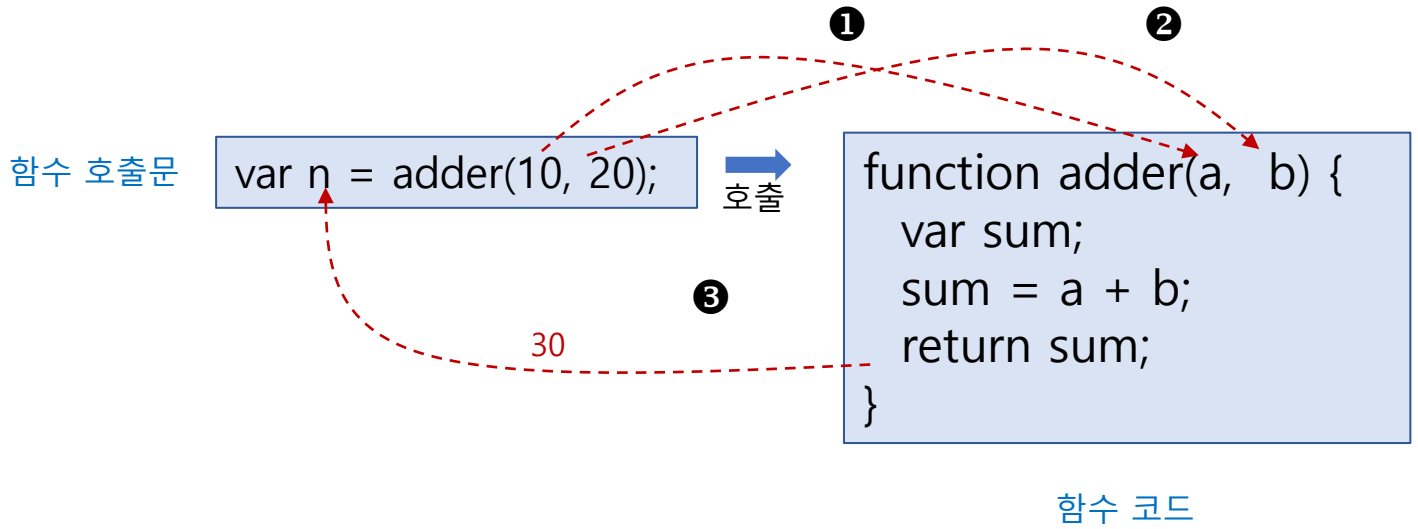
```
function 함수이름(arg1, arg2,..., argn) {
    ...프로그램 코드...
    결과를 리턴하는 return 문
}
```



함수의 구성과 호출

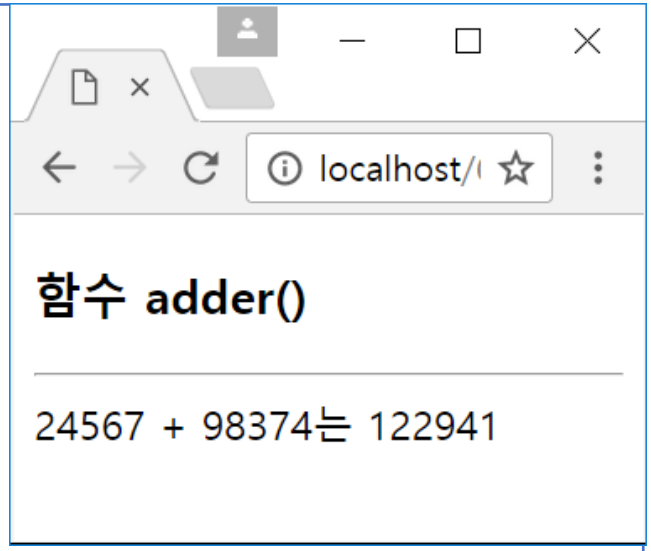
❖ 함수 호출

- 함수의 코드 실행 요청



[실습] 함수의 작성과 호출

```
<body>
   <h3>함수 adder()</h3>
   <hr>

   <script>
      // 함수 작성
      function adder(a, b) {
          var sum;
          sum = a + b;
          return sum;
      }


      // 함수 호출
      var n = adder(24567, 98374);
      document.write("24567 + 98374는 " + n + "<br>");
   </script>
</body>
```



자바스크립트에서 제공하는 전역 함수

❖ 대표적인 자바스크립트 함수

- eval() 함수
  - 식을 계산하고 결과 리턴
  - var res = eval("2*3+4*6"); // res는 30

- parseInt() 함수
  - 문자열을 숫자로 바꿔주는 함수
  - var I = parseInt("32"); // "32"를 10진수로 변환, 정수 32 리턴
  - var n = parseInt("0x32"); // "0x32"를 16진수로 해석, 정수 50 리턴

- isNaN() 함수
  - 숫자가 아님을 나타내는 함수(Not A Number)
  - isNaN(32) // false
  - isNaN(“32”) // false
  - isNaN(“hello”) // true
  - isNaN(NaN) // true

자바스크립트에서 제공하는 전역 함수

전역 함수명	설명
eval(exp)	exp의 자바스크립트 식을 계산하고 결과 리턴
parseInt(str)	str 문자열을 10진 정수로 변환하여 리턴
parseFloat(str)	str 문자열을 실수로 바꾸어서 리턴
inFinite(value)	value가 숫자이면 true 리턴
inNaN(value)	value가 숫자가 아니면 true 리턴

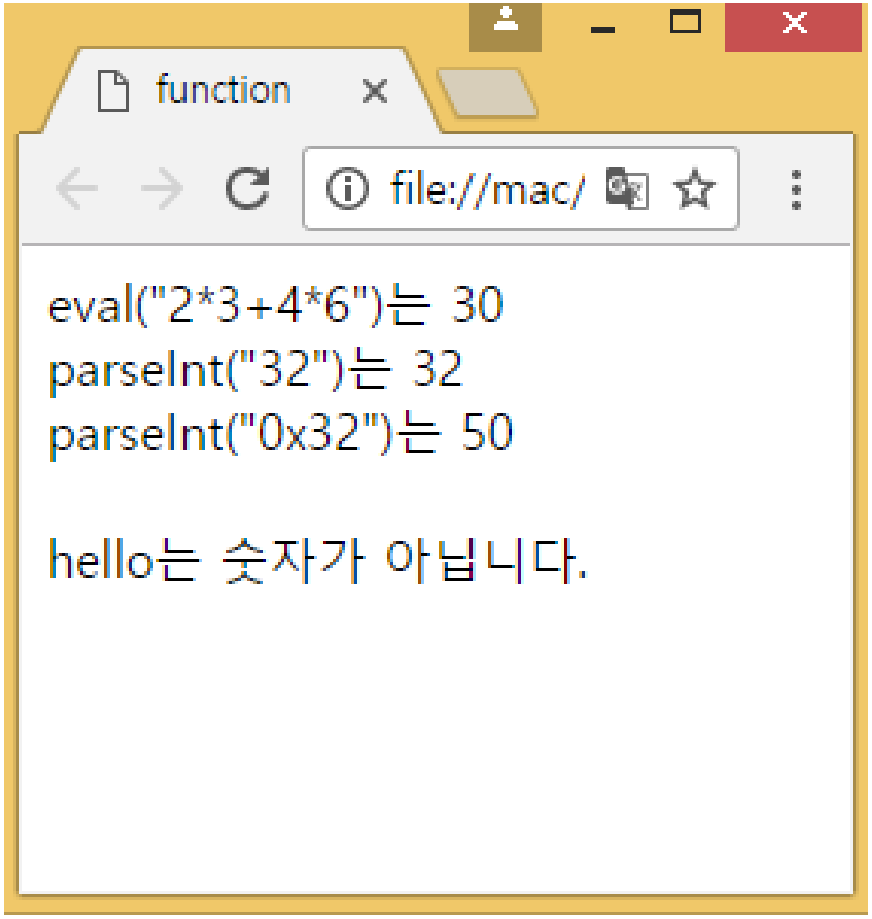
[실습] eval(), parseInt(), isNaN()

```
<script>
  function evalParseIntIsNaN() {
    var res = eval("2*3+4*6");
    document.write("eval(₩"2*3+4*6₩")는 " + res + "<br>");

    var m = parseInt("32");
    document.write("parseInt(₩"32₩")는 " + m + "<br>");

    var n = parseInt("0x32");
    document.write("parseInt(₩"0x32₩")는 " + n + "<br><br>");

  // "hello"는 정수로 변환할 수 없으므로 parseInt("hello")는 NaN 리턴
    n = parseInt("hello");

    if(isNaN(n)) // true
      document.write("hello는 숫자가 아닙니다.");
  }
</script>
```

[실습] eval(), parseInt(), isNaN()

```
<body>
  <h3>eval(), parseInt(), isNaN()</h3>
  <hr>
  <script>
    evalParseIntIsNaN();
  </script>
</body>
```

[실습] eval(), parseInt(), isNaN()

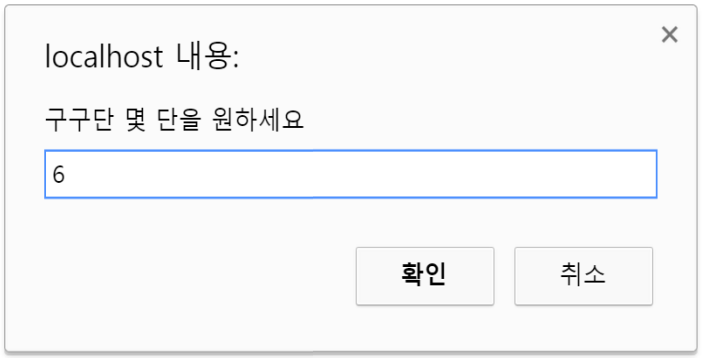


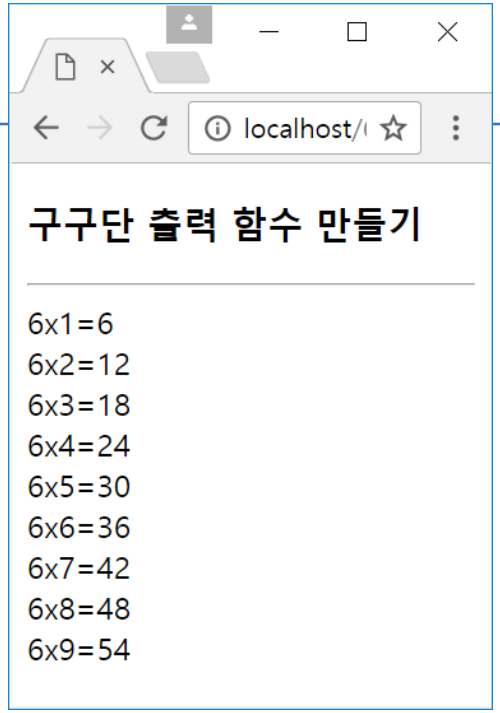
[실습] 구구단 출력 함수 만들기

```
<script>
  function gugudan(n) {      // 함수 작성
    var m = parseInt(n);      // 문자열 n을 숫자로 바꿈

    if(isNaN(m) || m < 1 || m > 9) {
      alert("잘못입력하셨습니다.");
      return;
    }

    // i는 1~9까지 반복
    for(var i=1; i<=9; i++) {
      document.write(m + "x" + i + "=" + m*i + "<br>");
    }
  }
</script>
```

[실습] 구구단 출력 함수 만들기

```
<body>
  <h3>구구단 출력 함수 만들기</h3>
  <hr>
  <script>
    var n = prompt("구구단 몇 단을 원하세요", ""); // n은 문자열
    gugudan(n); // 함수 호출
  </script>
</body>
```





02 다이얼로그

다이얼로그

❖ **자바스크립트 다이얼로그**

- 사용자 입력 및 메시지 출력

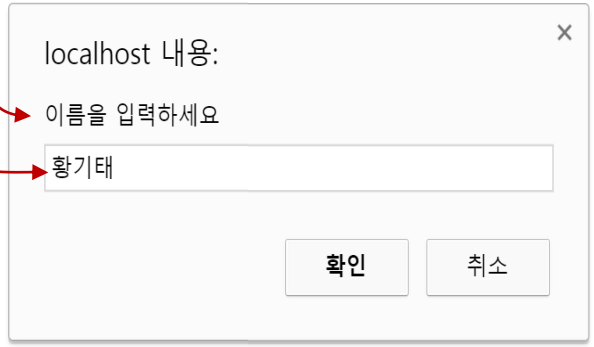
- 종류
  - 프롬프트(prompt) 다이얼로그
  - 확인(confirm) 다이얼로그
  - 경고(alert) 다이얼로그

prompt() 함수

❖ **prompt("메시지", "디폴트 입력값") 함수**

- 사용자로부터 문자열을 입력 받아 리턴
- 디폴트 값은 생략 가능

```
var ret = prompt("이름을 입력하세요", "황기태");
if(ret == null) {
   // 취소 버튼이나 다이얼로그를 닫은 경우
}
else if(ret == "") {
   // 문자열 입력 없이 확인 버튼 누른 경우
}
else {
   // 문자열 입력 후 확인 버튼 누른 경우
}
```

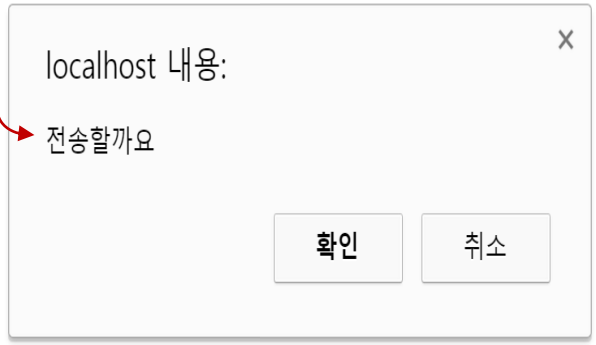


confirm() 함수

❖ **confirm(“메시지”) 함수**

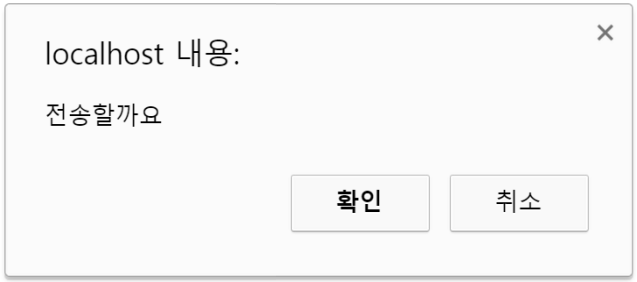
- “메시지”를 출력하고 확인/취소(OK/CANCEL) 버튼을 가진 다이얼로그 출력
- ‘확인’ 버튼을 누르면 true 리턴
- ‘취소’ 버튼을 누르거나 강제로 닫으면 false 리턴

```
var ret = confirm("전송할까요");
if(ret == true) {
   // 사용자가 "확인" 버튼을 누른 경우
}
else {
   // 취소 버튼이나 다이얼로그를 닫은 경우
}
```



[실습] confirm() 함수

```
<body>
  <h3>confirm() 함수 만들기</h3>
  <hr>
  <script>
      var ret = confirm("전송할까요");

      if(ret == true) {
        // 사용자가 "확인" 버튼을 누른 경우
        document.write("확인 버튼을 눌렀습니다.");
      }
      else {
        // 취소 버튼이나 다이얼로그를 닫은 경우
        document.write("취소 버튼을 눌렀습니다.");
      }
  </script>
</body>
```

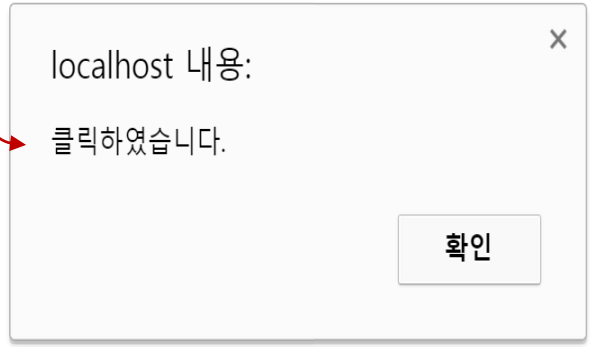


alert() 함수

❖ **alert(“메시지”) 함수**

- 메시지와 ‘확인’ 버튼을 가진 다이얼로그 출력
- 메시지 전달

```
alert("클릭하였습니다.");
```



03

화살표 함수

화살표 함수(arrow function)

❖ 화살표 함수

- 함수 표현식보다 단순하고 간결한 문법으로 함수를 만들 수 있는 방법

```
let 함수이름 = (arg1, arg2,..., argn) => {
    // 프로그램 코드
    // 결과를 리턴하는 return 문
}
```

[실습] 화살표 함수

```
let sum = (a, b) => {
    return a + b;
}

console.log(sum(1, 2));
```

❖ **실행 결과**

```
3
```

화살표 함수(arrow function)

❖ 매개변수가 없는 경우

```
let sum = () => {
    console.log("안녕하세요!");
}

sum();
```

- 중괄호 생략 가능
  - 코드가 한 줄인 경우

```
let sum = () => console.log(age);
```

화살표 함수(arrow function)

❖ 매개변수가 1개인 경우

```
let sum = (age) => {
    console.log(age);
}

sum(10);
```

- 소괄호, 중괄호 생략 가능
  - 소괄호 : 매개변수가 한 개인 경우
  - 중괄호 : 코드가 한 줄인 경우

```
let sum = age => console.log(age);
```

화살표 함수(arrow function)

❖ **매개변수가 2개 이상인 경우**

```
let sum = (age, name) => {
    console.log(age, name);
}

sum(10, "정수아");
```

- 실행 결과

```
10, '정수아'
```

화살표 함수(arrow function)

❖ **코드가 여러 줄인 함수**

- 중괄호로 코드를 묶어줘야 함
- return 키워드를 이용하여 결과값을 반환해야 함

```
let sum = (num1, num2) => {
    let result = num1 + num2;
    return result;
}

console.log(sum(10, 20));
```

화살표 함수(arrow function)

❖ **코드가 한 줄인 함수**

```
let sum = (num1, num2) => {
    return num1 + num2;
}

console.log(sum(10, 20));
```

- 중괄호, return 키워드 생략 가능

```
let sum = (num1, num2) => num1 + num2;

console.log(sum(10, 20));
```


THANK YOU