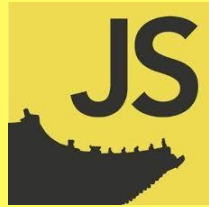


{ } + []



Javascript 07

-Function and Scope-

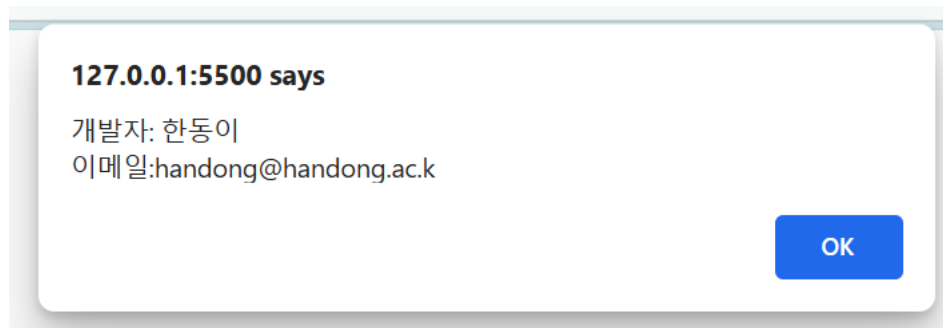
{ Na Daeyoung } + [soriru@handong.edu]

Contents

- Function

P01. 기본 함수-출력

- 함수를 호출하면 프로그램 개발자의 이름과 이메일을 출력하는 프로그램을 작성
 - 함수의 이름 proAuthor()
 - 알림창(alert())으로 출력
 - 파라미터 없음, return 없음
 - 개발자:자신의 이름, 이메일: 자신의 이메일



P01. 기본 함수-출력

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    function proAuthor(){
      alert("개발자: 한동이\n이메일: handong@handong.ac.k");
    }

    proAuthor();

  </script>
</body>
</html>
```

P02. 기본 함수-추가1

- 함수를 호출하면 프로그램 개발자의 이름과 이메일을 출력하는 프로그램을 작성
 - P01에 이름과 이메일을 prompt로 입력 받아서 출력하도록 수정
 - 단, printAuthor() 함수에 **파라미터로 이름과 이메일 전달**
 - prompt()는 printAuthor()함수 외부에서 사용

127.0.0.1:5500 says

개발자 이름:

OK Cancel

127.0.0.1:5500 says

이메일 주소:

OK Cancel

127.0.0.1:5500 says

개발자: 홍길동
이메일:hong@handong.ac.kr

OK

P02. 기본 함수-추가1

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    let name=prompt('개발자 이름:');
    let email=prompt('이메일 주소:');

    function proAuthor(developerName, developerEmail){
      alert(`개발자: ${developerName}\n이메일:${developerEmail}`);
    }

    proAuthor(name,email);

  </script>
</body>
</html>
```

P03. 기본 함수-추가2

- 함수를 호출하면 프로그램 개발자의 이름과 이메일을 출력하는 프로그램을 작성
 - prompt()로 개발자의 정보를 입력 받는 코드를 함수로 변경
 - 함수명: InputInfo()
 - 함수에서 prompt()로 입력 받은 값을 반드시 지역변수에 저장
 - 개발자 이름만 알림창으로 출력

127.0.0.1:5500 says

개발자 이름:

OK Cancel

127.0.0.1:5500 says

이메일 주소:

OK Cancel

127.0.0.1:5500 says

개발자: 한동이
이메일: undefined

OK

P03. 기본 함수-추가2

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    function infoInput(){
      let name=prompt('개발자 이름:');
      let email=prompt('이메일 주소:');
      return name;
    }

    function proAuthor(developerName, developerEmail){
      alert(`개발자: ${developerName}\n이메일:${developerEmail}`);
    }

    let name=infoInput();
    proAuthor(name);

  </script>
</body>
</html>
```

이름, 이메일 을 반환 해야 하지만
return은 하나의 데이터만 반환할 수 있음

이름 정보만 받아서 파라미터로 전달

코드의 문제점은?

함수에서 여러 개의 데이터 반환

- 앞의 코드의 문제점은 여러 개의 데이터를 반환해야 하는 상황
- Return은 하나의 데이터만 반환 가능
- 이 문제를 해결하기 위해서는??



- 두 개의 데이터를 하나의 데이터 형태처럼 반환
- 앞에서 배웠던 개념 중

하나의 변수 이름에 여러 개의 데이터를 저장 할 수 있었던 것은? **배열 or 객체**

여러 개의 데이터를 배열이나 객체에 저장하면 하나의 변수처럼 전달, 반환 가능

함수에서 여러 개의 데이터 반환

```
<script>
  function infoInput(){
    let name=prompt('개발자 이름:');
    let email=prompt('이메일 주소:');

    return [name, email]; //하나의 배열에 두 개의 데이터를 저장하여 반환
  }

  function proAuthor(developerName, developerEmail){
    alert(`개발자: ${developerName}\n이메일:${developerEmail}`);
  }

  let [tempName, tempEmail]=infoInput(); //반환된 배열을 배열로 받아서 tempName, tempEmail 변수에 저장

  //배열 구조 분해 할당(Destructuring Assignment) 이라고 함
  proAuthor(tempName,tempEmail);
</script>
```

P04. 평균 계산 함수

- 사용자에게 두 개의 수를 입력 받고 평균을 출력하는 프로그램을 작성
 - 두 개의 수를 입력 받고 평균을 계산하는 함수: avgScore() 작성
 - 평균을 <h1>태그에 출력하는 함수: printAvg() 작성

127.0.0.1:5500 says

Input 1st score:

OK Cancel

127.0.0.1:5500 says

Input 2nd score:

OK Cancel

```
<body>  
  <h1 id="printavg">평균:</h1>  
  <script> ...  
</script>  
</body>
```

평균:79.062

- 추가문제
 - 아래 이미지와 같이 입력 받은 점수도 출력하는 코드 추가
 - 점수를 출력할 <h1> 태그 추가
 - 단, printAvg()에서 모든 출력 담당

P04. 평균 계산 함수

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1 id="printavg">평균:</h1>
  <script>
    function avgScore(){
      let score1=Number(prompt('Input 1st score:'));
      let score2=Number(prompt('Input 2nd score:'));
      let avg=(score1+score2)/2;

      return avg;
    }

    function printAvg(scoreAvg){
      document.getElementById('printavg').innerHTML+=scoreAvg;
    }

    let tempAvg=avgScore();
    printAvg(tempAvg);
  </script>
</body>
</html>
```

P04. 평균 계산 함수-추가문제

- 사용자에게 두 개의 수를 입력 받고 평균을 출력하는 프로그램을 작성
 - 추가문제
 - 이미지와 같이 입력 받은 점수도 출력하는 코드 추가
 - 점수를 출력할 <h1> 태그 추가
 - 단, printAvg()에서 모든 출력 담당
 - 평균을 출력 시 소수점 이하 3자리 까지만 출력(추가 학습 내용)

127.0.0.1:5500 says

Input 1st score:

OK Cancel

127.0.0.1:5500 says

Input 2nd score:

OK Cancel

입력 받은 점수:98, 65.12345

평균:81.562

P04. 평균 계산 함수-추가문제

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1 id="input">입력 받은 점수:</h1>
  <h1 id="printavg">평균:</h1>
  <script>
    const h1Input=document.getElementById('input');
    const h1Print=document.getElementById('printavg');

    function avgScore(){
      let score1=Number(prompt('Input 1st score:'));
      let score2=Number(prompt('Input 2nd score:'));
      let avg=(score1+score2)/2;

      avg=Number(avg.toFixed(3)); //소수점 이하 3자리까지 표시 4자리에 반올림 후 문자로 변환, 변환된 문자를 숫자로 다시 형변환
      return [score1, score2, avg];
    }

    function printAvg(score1,score2,scoreAvg){
      h1Input.innerHTML=`입력 받은 점수:${score1}, ${score2}`;
      h1Print.innerHTML=`평균:${scoreAvg}`;
    }

    let [tempScore1, tempScore2, tempAvg]=avgScore();
    printAvg(tempScore1, tempScore2, tempAvg);
  </script>
</body>
</html>
```

P05. 함수와 반복문

- 사용자에게 출력하기 위한 구구단을 입력 받고 출력하는 프로그램 작성
 - inputTimes() 함수
 - 사용자가 end를 입력하면 프로그램 종료
 - Good bye 알림을 화면에 출력
 - end가 입력되기 전까지 무한 반복하며 입력 받음
 - 사용자의 입력이 2~9사이가 아니면 알림창 메시지 출력
 - 입력 오류입니다. 다시 입력하세요.
 - printTimes()함수에게 받은 결과를 하나의 <p>태그에 출력
- printTimes() 함수
 - 구구단의 출력을 생성하여 inputTimes()에게 반환
 - Ex) 2를 받으면 2단 내용을 생성하여 반환

127.0.0.1:5500 says
구구단 단수를 입력하세요:
3
OK Cancel

127.0.0.1:5500 says
구구단 단수를 입력하세요:
5
OK Cancel

127.0.0.1:5500 says
구구단 단수를 입력하세요:
end
OK Cancel

127.0.0.1:5500 says
good bye
OK

구구단 출력

3단

3*1=3

3*2=6

3*3=9

3*4=12

3*5=15

3*6=18

3*7=21

3*8=24

3*9=27

5단

5*1=5

5*2=10

5*3=15

5*4=20

5*5=25

5*6=30

5*7=35

5*8=40

5*9=45

P05. 함수와 반복문-hint

- 사용자에게 출력하기 위한 구구단을 입력 받고 출력하는 프로그램 작성
 - printTimes() 함수
 - 구구단의 출력을 생성하여 inputTimes()에게 반환
 - Ex) 2를 받으면 2단 내용을 생성하여 반환

```
function printTimes(number){  
  let result=number+"단<br>";  
  
  for(let i=1;i<10;i++){  
    result+=`${number}*${i}=${(number*i)}<br>`;  
  }  
  
  return result;  
}
```

P05. 함수와 반복문-hint

```
<body>
  <p>구구단 출력<br></p>
  <script>
    const printTag=document.querySelector('p');

    function inputTimes(){
      while(true){
        let number=prompt('구구단 단수를 입력하세요:');

        if(number=='end'){
          alert("good bye");
          return;
        }else{
          if(number>10 || number<2){
            alert("입력 오류 입니다. 다시 입력하세요.");
          }else{
            printTag.innerHTML+=printTimes(number)+"<br>";
          }
        }
      }
    }

    function printTimes(number){
      let result=number+"단<br>";
      for(let i=1;i<10;i++){
        result+=` ${number}*${i}=${(number*i)}<br>`;
      }
      return result;
    }

    inputTimes();
  </script>
</body>
```

{ 감사합니다 } + [Thankyou]

{
본 교재 또는 강좌는 2025년도 과학기술정보통신부 및 정보통신기획평가원에서 주관하여 진행하는 'SW중심대학사업'의
결과물입니다.
}