ㅂ

지난 시간에 이은 퀴즈

1. For문을 사용해서 25부터 100까지 출력하라
2. While문을 사용해서 100부터 0 까지 출력하는데, 값에 3씩 빼서 출력해라.(--금지)
3. 1번과 2번을 사용해서 출력하는 숫자가 짝수 인 경우 ${값} 은 짝수 입니다. , 홀수 인 경우 ${값}은 홀수입니다. 를 출력해주세요
4. For문과 while문에서 두드러지게 나타나는 차이는 무엇인가 ?
5. 배열이란 무엇인가 ? (<https://offbyone.tistory.com/133>) 참조
6. 인덱스란 무엇인가 .
7. 상수 arr\_1에 빈 배열을 할당하라
8. 상수 arr\_2에 0부터 10까지 각 방에 아이템을 넣어라
9. For문을 사용하여 arr\_1 배열에 0부터 100까지 아이템을 넣어라
10. While문을 사용해서 arr\_1배열을 모두 출력하는데, 3번문제와 같이 홀수와 짝수를 나누어라
11. 0번째 인덱스의 방에서 76번째 방까지 아이템이 들어있는 배열의 길이는 몇인가 ?
12. Arr\_1의 .length를 이용하여 for문을 사용하여 3번째 방마다 출력하라. If문이 아닌 인덱스 값에 3씩 더하여 출력할 것
13. 배열에는 슬라이스라는 기능이 있다. .slice(자르기 시작할 위치 , 가위질이 끝나는 위치)를 이용하여 arr\_2의 배열을 절반 만큼 자르고 자른 배열을 arr\_3에 할당하라. <https://hokeydokey.tistory.com>에서 배열에 관한 내용 참조 가능

함수란 무엇일까요 ? 수학적으로 함수란 어떤 값을 입력 받아 그 수를 가공하여 특정 값으로 배출하는 절차를 말하는 것 입니다. 프로그래밍에서 함수 또한 그 내용은 거의 다른 것이 없으나, 아주 약간씩의 차이가 있습니다.

자바스크립트에서 함수를 그림으로 표현하면 다음과 같습니다.

인풋이 있고, 아웃풋이 있는 형태, 인풋이 있지만, 아웃풋이 없는 형태, 인풋이 없지만 아웃풋은 있는 형태, 인풋도 아웃풋도 없는 형태로 나눌 수 있습니다. 자바스크립트에서는 이와 같은 인풋과 아웃풋의 형태 때문에 수학적인 의미의 함수의 의미보다 기능이라는 측면이 좀 더 어울립니다.

그리고 우리는 console.log와 alert, while, if등과 같은 것들을 사용해서 함수라는 기능을 사용하고 있었습니다. 함수라는 표현 때문에 어렵고 무서운 것 같지만 실제로는 별 다른 것이 없고, 우리 프로그램을 더욱 깔끔하고 멋지게 만들어 줄 수 있는 방법입니다.

간단하게 특정한 두 값을 전달 받아 두 값을 서로 출력해주고, 값을 더해주는 함수를 만들어 보도록 하겠습니다. 이 예제에서는 인풋은 있으나 아웃풋이 없는 형태 입니다.

자바스크립트에서 함수를 표현하는 고전적인 방법은 function이라는 예약어를 사용하는 것 입니다.

Function sum(num\_1, num\_2){

Console.log(num\_1);

Console.log(num\_2);

Console.log(`${num\_1} + ${num\_2} = ${num\_1 + num+2}`);

}

이렇게 함수를 작성하고 아래에서 sum(10,30); 을 작성하면 띠용 하고 우리가 작성한 함수의 내용이 나옵니다.

함수를 호출해보도록 하겠습니다. 프로그래밍에서는 함수를 사용한다고 하기 보다는 호출한다고 하는 성격이 강하니 이제부터는 사용이라는 단어 대신 호출이라는 단어를 사용하도록 하겠습니다.

우리가 만든 함수에 sum(1,2); 형태를 취해 노드js로 확인해보니 1,2 3이 출력되는 것을 볼 수 있습니다.

여기까지 한 후 함수의 모습을 분석해보록 하겠습니다.

함수를 정의하기 위해서는 for, while,if처럼 function을 선언하고 펑션 옆에는 함수 명을 작성합니다. 그 옆에있는 부분은 매개변수라고 하는 파라메터 부분입니다. 이 파라메터는 함수의 정의 부분에 나열되어 있는 변수로써 호출했을 때 해당 값이 어떤 코드에 쓰일 지 지정하는 변수 입니다. 이 파라메터는 호출자에 있는 전달인자를 보시면 알기 쉽습니다. 호출자()를 보면 우리가 1과 2를 전달 해주었습니다. 이 값들은 전달인자 argument라고 하는 것들인데, 이 쉼표로 구분되어 있는 전달인자들이 파라메터의 값들과 1:1매핑되어 함수의 정의 부분에서 지정되어 있는 파라메터의 움직임과 동일하게 움직입니다.

그래서 sum함수를 살펴보면 1이 num\_1, 2가 num\_2로 매핑되는 모습을 볼 수 있습니다 .

다시 정의 부분으로 돌아가서 파라메터 정의 부분이 끝나면 함수 호출되면 실행 할 코드를 블럭으로 감싸 작성합니다.

그렇다면 이번에 함수를 사용해서 전달인자로 배열을 받고, 배열의 길이만큼 출력하는 함수를 같이 만들어보도록 하죠

Function showAll(array){

For(let I=0; I <= array.length-1; i++){

Console.log(array[i]);

}

}

showAll(array);

어때요 참 쉽죠 ?

이번에는 두 값을 전달 받아, 두 값 중 더 큰 값을 출력하는 함수를 만들어 보도록 할까요 ?

(각자 한번 만들어 보세요.)

잘하셨습니다.

이번에는 output을 만들어 보도록 하겠습니다. Output을 만든다는 의미는 이 함수의 결과가 값이 된다는 의미로 added = sum(1,2); 꼴의 형태가 된다는 의미입니다. sum라는 함수를 첫번째 인자로 전달받은 값과 두번째 인자를 출력하고, 두개를 더한 값을 이 함수에서 반환하도록 하겠습니다. 이 반환 즉 output을 표현 해주는 예약어는 return 입니다.

Function sum(num1,num2){

Console.log(num1)

Console.log(num2)

Return num1 + num2;

}

이 sum라는 함수에 return을 주고 num1 = sum(20,30); 하여 console.log(num1)을 해보면 50이 출력되는 결과를 볼 수 있습니다. 그렇다면 console.log안에서 저 sum함수를 호출하면 어떻게 될까요 ? 네 그렇습니다. …

이번에는 지난번에 사용해보았던 랜덤한 값을 이용해서 input이 없고 반환값으로 랜덤한 길이의 배열을 반환하는 함수를 한번 만들어보도록 하겠습니다.

Function randomePush(){

Const random = Math.floor(math.random() \* 100);

Const array = [];

For(let I = 0; I < random; I++){

Const value = Math.floor(math.random() \* 10);

Array[i] = value;

}

Return array;

}

이 함수를 이용하여 배열 2개를 만들어보시오.

그리고, 이 배열들을 전달 인자로 받아 이 배열의 길이만큼 값을 출력하는 함수를 만들어보시오.

2번을 확장하여 , 배열의 값을 인자로 받아 짝수인지 홀수인지 체크하고 값의 짝홀유무를 출력하는 함수를 만들고 2번 함수 반복문에서 호출하시오

고생하셨습니다. 여기까지 왔다면 전역변수와 지역변수라는 개념을 알아보고 함수를 더욱 활용해 보도록 하겠습니다. 전역변수와 지역변수의 차이는 함수 내부에서 변수가 생성이 되었는가 아니면 함수의 블록 바깥에서 생성이 되었는가의 차이입니다.

먼저 전역변수는 함수 바깥에서 선언된 변수를 말합니다. 함수 내부에서는 이 변수의 값을 참조 할 수 있습니다. 그래서 a라는 전역변수의 값을 참조하여 출력하는 함수를 한번 만들어 보도록 하죠.

let a = 10;

function outA(){

console.log(a);

a = 20;

console.log(a);

}

outA();

console.log(a);

이렇듯 함수 블락 외부에 있는 a값에 접근 할 수도 a변수를 참조 할 수도 있습니다.

이;번에는 지역변수를 알아보도록 하겠습니다. 지역변수는 함수내부에서 생성된 함수 입니다. 이 함수는 전역 변수와 다르게 함수 바깥에서 이 변수를 참조 할 수 없습니다. 그 이유는 이 변수는 함수가 호출 될 때만 생성되는 변수 이기 때문입니다.

지금 까지는 input은 있지만 아웃풋이 없는 형태 ,인풋도 있고 아웃풋도 있는 형태의 함수를 만들어 보았습니다. 이번에는 인풋도 아웃풋도 리셋 버튼을 누르면 모든 인풋창에 있는 값을 다시 돌리는 함수를 간단하게 짜보도록 하겠습니다 .

<!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>

<input id="id"/>

<input id="password"/>

<button onclick=resetInputBox()>리셋</button>

</body>

<script>

let inputBoxId = document.getElementById("id");

let inputBoxPassword = document.getElementById("password");

function resetInputBox(){

inputBoxId.value=""

inputBoxPassword.value = "";

}

</script>

</html>

우리는 inputboxid변수와 inputBoxpassword변수에 각각 id와 password input html 엘리먼트에 관한 정보를 넣어줬고 resetInputBox라는 함수는 규정된 파라메터가 없습니다. 그런데, 함수 바깥에 있는 변수의 내용을 가져와 해당 변수의 내용을 변경 하고 있습니다. 이렇게 반환값과 어규먼트가 없는 함수들은 대체적으로 전역변수를 참조하여, 변수의 내용을 정해진 값들로 변경하는 역할을 합니다.

이번에는 함수를 만들어 스타일을 변경해보자 .