지난 시간에 이은 퀴즈

1. 상수와 변수의 차이
2. 변수를 사용하는 2가지 방법
3. 상수를 사용하는 1가지 방법
4. 제어문 혹은 조건문 이란 ?
5. 자바스크립트의 제어문의 종류 3가지를 말해보시오
6. If문을 사용해서 a < b 라면 a는 b보다 작습니다를 출력해보시오
7. 6번에 이어서 else를 사용해서 a >b보다 작다면 a는 b보다 큽니다. 를 출력하시오
8. 7번에 이어서 else if를 사용해서 a == b 라면 a는 b와 같습니다를 출력하시오
9. 논리연산자 종류는 무엇이 있을까요 ?
10. 동등연산자 == 과 일치연산자 ===을 정의해보시고, 차이는 무엇인가요 ?
11. 크거나 같다를 어떻게 표현 할 수 있을까요 ?
12. 부정연산자는 무엇이며 a가 b보다 같지 않다를 표현 해보시오 .
13. AND && 연산자는 무엇이며 , or연산자 || 는 무엇인지 표현 하시오
14. 삼항 연산자를 사용하여, a가 b보다 크거나 같지 않다면 a는 b보다 크거나 같지 않습니다, 반대의 경우라면 a는 b보다 크거나 같습니다를 출력하시오

지난 시간에는 프로그래밍을 조건에 따라 분기하는 제어문에 대해서 학습해보았습니다. 이번 시간에는 반복문과 배열이라는 데이터 타입을 익혀보도록 하겠습니다. 우리가 컴퓨터를 주로 사용하는 이유가 무엇일까요 ? 컴퓨터를 사용하는 이유는 아마도 우리가 하기 싫을 일들을 반복적으로 고속으로 처리해주기 위함 일 겁니다. 지금 까지 우리가 배웠던 console.log를 1번 작성 해보실까요 ? 이번에는 10번을 작성하는데, 1부터 10까지 출력해보도록 할까요 ? 자 어떠셨나요 이처럼 적은 양의 프로그램을 만들 때는 쉽지만, 조금만 변경되고 반복이 가중될수록 힘들고 지겹습니다. 그래서 프로그래밍 언어에서는 이처럼 복잡하고, 힘든 일을 컴퓨터가 반복할 수 있게 끔 해주는 반복문이라는 것이 있습니다. 자바스크립트를 포함하여, 대표적인 반복문으로는 for문과 while문이 있습니다. For문의 생김새를 보며 익혀보도록 할까요 ? for문을 사용하기 위해서는 for문을 선언 해주셔야 합니다. 그리고 for문의 오른쪽에는 () 조건에 쓰일 변수와 반복이 종료되는 조건, 반복 될 때 마다 실행될 문장입니다. 그리고 블락을 열어주어 반복실행하게 할 프로그램을 작성 해주시면 됩니다.

for(조건에 쓰일 변수선언; 반복이 끝날 조건; 반복이 될 때 마다 실행할 코드){

반복이 될 프로그램

}

그렇다면 이번에는 for문을 예제로써 익혀보도록 할까요? Console.log(안녕하세요) 를 열번 실행시키도록 하겠습니다. For(let I = 0; I < 10; i++){

Console.log(안녕하세요)

}

For문장 옆에는 반복에 필요한 것들을 작성해보았는데요. 한번 읽어 보도록 하겠습니다. I라는 변수를 선언하고 i변수가 10보다 작다면 이 반복문을 작동 시키도록 하겠습니다. 그리고 반복이 종료될 때 마다 i변수를 1씩 증가 시켜주세요. I옆에 있는 ++은 후위증감자라는 놈으로 왼쪽에 있는 변수를 1씩 증가시키는 증감자입니다. 반대로 -–라는 녀석은 후위감소자라는 놈으로 왼쪽에 있는 변수를 1씩 감소 시키는 변수입니다. 자세한 내용은 for문의 말미에 말씀드리도록 하겠습니다.

그래서 해당 프로그램에서 i로 변경하고 작동시키면 console.log(i)가 10번 출력되고 종료되는 것을 볼 수 있습니다. 이번에는 반복문안에 조건문을 작성하여, 2의 배수만 출력하는 프로그램을 작성하도록 하겠습니다.

For(let I = 0; I < 10; i++){

If(I % 2 === 0)

Console.log(I)

}

% 나머지 연산자는 왼쪽에 있는 수를 오른쪽으로 나눠주어 남는 나머지를 계산 하는 연산자 입니다. 그래서 10 %2 를 했을 경우 0으로 딱 떨어지기 때문에 짝수만 출력되게 됩니다.

이번에는 i를 10부터 0까지로 내려가며 출력하는데, 홀수인 경우에만 I 값을 출력하는 프로그램을 짜보도록 하겠습니다. 이번에는 후위 감소자를 쓰면 될 것 같네요 ㅎㅎ.

잘하셨습니다. ! 반복문을 다룰 때 가장 중요한 것은 반복문이 종료할 조건입니다. 어떤 프로그램의 반복문들은 반복을 종료하는 조건이 매우 복잡하게 엮여있는데, 이때 프로그램을 잘 못 짜서 반복문의 조건이 무한히 true를 반환한다면 이 프로그램은 무한히 프로그램을 반복하다, 메모리를 다 잡아먹어 종료될 것 입니다.

예를 들어 for(let I = 0; I < 10; i--;){

Console.log(err)

}

Console.log(hi)

라는 잘못된 반복문을 실행하게 되면 무한히 err를 출력하고 다른 프로그램은 작동하지 못합니다. 이를 이터널 루프라고 합니다.

조금 더 나아가서 for문의 반복문에 필요한 문장들을 커스텀 할 수 도 있습니다. 만약 변수를 선언하지 않고, 외부에 있는 변수를 변경시키고, 사용하고 싶다면 i를 선언했던 부분을 지우고;로 선언하지 않고 사용할 수 있습니다. 그리고 i—와 같은 부분도 필요 없다면 과감히 삭제 시킬 수 있습니다. 그렇지만 이 경우에는 이터널 루프에 빠지게 되니 주의가 더욱 필요할 겁니다.

이제 아까 설명하던 증감자, 감소자 연산자의 사용법에 대해서 간략히 설명 드리도록 하겠습니다.

증감자 ++ 감소자 – 는 연산에 필요한 변수의 위치에 따라 후위와 전위로 나뉘게 됩니다. 이 둘의 차이는 해당 코드를 만났을때 즉시 실행되나, 다음 코드로 넘어 간 후 실행되는가 입니다.

Let dummy = 0 ;

Console.log(dummy++)을 했을 경우 0이 출력되지만 다음 console.log(dummy)로 더미를 체크해보면 dummy변수가 증가했다는 것을 알 수 있습니다. 전위 증감자인 ++은 해당 코드를 만날 경우 즉시 1을 올려줍니다. Console.log(++dummy) . 같은 원리로 감소자 또한 전위와 후위에 맞게 1씩 감소됩니다. 그러나 1씩 증가하거나 감소하는 것이 싫을 수도 있습니다. 그럴 때는 변수 = 변수 + 증가할 숫자 혹은 변수 += 증가할 숫자를 작성해주면 오른쪽에 들어가있는 숫자 만큼 변수가 재할당되게 됩니다.

다음은 while문 입니다. While문은 for문과 조금 다르지만 반복문이라는 것은 동일합니다.

다만 for문과 달리 조건을 위한 선언식과 반복후 실행할 문장이 없다는 점이 가장 큰 차이 점 입니다. 그래서. While문을 쓸 때는 반복문을 빠져나가기 위해서 조건을 위해 값을 변경하는 행위가 내부에서 반복실행되는 코드에서 실행됩니다.

While문의 사용은 매우 간단합니다. While을 적고 소괄호를 연 다음, 조건을 명시하며 반복실행을 위한 블락을 열어줍니다. Dummy변수를 외부에 작성하고, 0부터 10까지 출력하는 코드를 한번 실행 해보도록 할까요 ?

Let dummy = 0;

While(dummy < 10){

Console.log(dummy);

Dummy++;

}

아주 잘하셨습니다. 이번에는 for문과 while문을 써서 각각 Index변수가 0부터 100까지 출력하는데, 조건문을 사용해 3의 배수와 5의 배수일 경우 출력하지 않도록 해주세요.

다음은 0부터 30까지 2씩 조건문을 쓰지 않고 출력하는 반복문을 for과 while문을 만들어주세요 .

잘하셨습니다. 이제 반복문을 조금 더 특별하게 해줄 분기문 2개를 알아 보도록 하겠습니다.

만약 우리가 반복문에서 0부터 100까지 출력하다가 20을 만나면 이 반복문을 해제하고 싶을 수도 있습니다. 물론 조건에서 21까지 제한을 걸어두고 만들 수도 있지만, 우리가 만들 프로그램은 0부터 100까지 변수를 증가시키다 20일 경우 종료하는 프로그램입니다. 이 때는 break라는 분기문을 작성합니다. 만약 반복문이 실행도중 break문을 만나면 아직 반복되는 조건이 충족되었음에도 , 반복문이 종료되는 것을 볼 수 있습니다.

다음 continue는 해당 키워드를 만나면 이 반복문은 해당 실행되는 코드는 중단 되며 다음 반복으로 넘어가게 됩니다.

배열은 우리가 코딩을 하기 매우 편하게 만들어주고 자주 쓰이는 데이터 타입입니다. 배열이라는 말을 들었을 때 어떤 것이 생각날까요 ? 역시 다들 직선적인 구조로써 무언가 순서대로 정렬되어 있는 것이라는 생각이 들죠 ? 맞습니다. 프로그래밍에서 배열은 비슷한 것들 끼리 순서대로 저장되어 있는 자료구조 입니다. 그럼 이런 생각이 들 수도 있겠죠. 굳이 배열을 안 적어도 사용할 수 있는 것 아닌가요 ? 그렇습니다만 우리의 코드라인이 엄청나게 많은 변수들로 차지하고 있는 것은 매우 부적절한 행위이며, 파일의 크기를 늘릴 뿐입니다.

배열에서 중요한 점은 배열은 비슷한 것들 끼리 순서대로 저장된다는 것 입니다. 우리가 기억하고 있는 데이터 타입은 무엇이 있나요 ? 여러가지가 있죠 ? 다른 언어에서 배열은 같은 데이터 형식을 가지고 있는 데이터들을 순서대로 저장합니다. 그러나, 자바스크립트에서 배열은 다른 형식이라도 상관 없이 같이 저장 할 수 있으며, 배우지는 않았지만 함수라는 녀석도 저장이 가능하고 , 같은 배열도 저장이 가능합니다. 이렇게 자유로운 자바스크립트에서 프로그래밍적 배열의 의미를 사용하기 위해서는 유저들간 어떠한 기준점이 필요한데, 그것은 바로 비슷하고 같은 역할을 하지만 다른 데이터를 가지고 있는 것들을 한 배열에 저장 하자 입니다.

배열을 표현 하는 방법은 대괄호를 이용하는 것입니다. 그래서 배열을 선언할 때는 let array = [];

와 같이 표현하고, 할당을 하며 선언을 할 때는 let array = [10,2,3,4,5,6,7,1,2] 와 같이 한다. 현재 만들어진 배열을 그림으로 표현하면 다음과 같다. 작은 방들이 다닥다닥 모여, 그 방에는 우리가 할당하며, 작성했던 숫자들이 한방씩 차지 하고 있습니다. 이 방에 들어가기 위해서는 우리는 그 방에 들어가기 위한 열쇠가 필요한데, 이를 인덱스라고 합니다. 인덱스는 다들 어디서 들어보셨죠 ? 바로 인덱스는 목차라는 의미로써 책의 앞부분에서 내가 찾고자 하는 부분을 쉽게 찾을 수 있도록 순서대로 작성해놓은 것과 동일한 역할을 합니다. 그래서 우리가 이 배열에서 원하는 item을 사용하려면 배열의 이름옆에 대괄호를 열고, 인덱스를 작성해줍니다. 그렇지만 이때 주의해야 할것이 있습니다. 바로 배열의 첫번째 방은 1번이 아니라 0번입니다. 이점 꼭 명심해주시고 프로그램에 임해 주시길 바라겠습니다.

https://hokeydokey.tistory.com/12