지난 시간에 이은 퀴즈

1. 상수와 변수의 차이
2. 제어문이란 ?
3. 자바스크립트의 제어문의 종류 3가지를 말해보시오
4. (a = 20; b = 30 ; c = 40) a가 b보다 큰 경우에 a는 b보다 큽니다를 출력해주시고 아니라면 b = a보다 큽니다를 출력해주세요.
5. 4번에 이어서 b가 a보다 큰 경우에 한하여 , c가 b보다 크다면 c는 b보다 큽니다를 출력해주시고 아니라면 b는 c보다도 큽니다를 출력해주세요
6. 논리 연산자 and 연산자를 사용하여 , b가 a보다 크고 c보다 작은 경우 b는 c와 a의 median값입니다. 를 출력해주세요.
7. 자바스크립트 반복문의 종류는 무엇이 있나요 ?
8. While문과 for문의 차이는 어떻게 되나요 ?
9. For문을 사용하여, 0 부터 10까지 출력해주세요
10. For문을 사용하여 50부터 0까지 출력하되 2의 배수라면 I는 짝수입니다를 출력해주세요
11. While문을 사용하여 25부터 100까지 출력하되 홀수 인 경우 ${i}값은 홀 수 입니다를 출력해주세요
12. 전위증감자와 후위증감자는 무엇이 다른가요 ?
13. 반복문에서 break를 만나면 어떻게 되나요 ?
14. 반복문에서 continue를 만나면 어떻게 되나요 ?

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 코드 라인이 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 이름 옆에 대괄호를 열고, 인덱스를 작성해줍니다. 그렇지만 이때 주의해야 할 것이 있습니다. 바로 배열의 첫번째 방은 1번이 아니라 0번입니다. 이점 꼭 명심해주시고 배열을 사용해 주시길 바라겠습니다.

배열을 선언해보고, 값을 한번 넣어 보도록 하겠습니다.

Cosnt array = [];

배열을 값이 새롭게 할당되는 것이 아닌 해당 데이터타입에 추가되는 것이니, 상수 형태로 사용해도 무방하고, 새롭게 배열을 재할당 될 일이 없다면 상수 형태로 해주는 것이 좋습니다 .

해당 array를 선언하고 나서 console.log()로 array를 확인해보니 대괄호 안에 아무 것도 없는 형태를 볼 수 있습니다. 여기서

array[0] = 1;

array[1] = 2;

를 하여 각 배열의 방안에 값을 넣어주고 내부를 확인 해보면 대괄호 안에 값이 들어있는 것을 알 수 있습니다. 이때 2번방과 3번방에 숫자가 아닌 스트링 형태를 한번 넣어볼까요 ? 0번방과 1번방을 직접 한번 보기도 해볼까요 ? console.log(array[0]) 이렇게 확인 하니 배열의 0번방에 직접 접근하여 값을 가져올 수 있습니다.

Array[2] = “it’s two”;

Array[3] = “it’s three”;

이렇게 넣고 console.log(Array)로 배열안을 확인 해보면 다른 타입임에도 같은 배열안에 들어 가 있음을 확인 할 수 있습니다.

이번에는 조금 특이하게 array[10]다가 30이라는 숫자를 넣으면 어떻게 될까요? 네 , 우리가 따로 값을 할당하지 않은 배열의 공간에는 null로 가득 차게 됩니다. 만약 우리가 배열의 끝보다 더 넘는 공간을 확인해보면 다음과 같이 undefined라고 나오며 아무것도 안나오는 것을 볼 수 있습니다.

그렇다면 이 배열과 반복문을 활용하여 배열을 0번 방부터 100번 방까지 0부터 100까지의 숫자를 넣는 반복문을 만들어 볼까요 ??

For(let I = 0 ; I < 101; i++){

Array[i] = I;

}

이렇게 배열 index를 표시하는 곳에 숫자 타입의 형태를 넣게 되면 해당 빈방에 접근 할 수 있습니다. 그렇다면 이번에는 배열의 0번째 방부터 101번째 방까지 방 번호와 방안에 들어 있는 숫자를 출력해볼까요 ?

잘하셨습니다.

하지만 우리가 만든 배열이 아닌 남이 보내준 배열이라면 어떻게 내부에 있는 내용을 끝까지 출력 할 수 있을까요? 이런 경우에는 배열의 끝을 알 수 없기 때문에 언제까지 배열의 내용을 확인 할 수 있는지 모릅니다. 그렇지만 자바스크립트에는 친절하게도 이런 배열의 속성중에 length라는 녀석이 있습니다. 이 length라는 녀석은 배열의 끝 방까지의 몇개의 item이 배열안에 존재하는 지 알려줍니다. 그래서 우리가 백개의 아이템을 넣은 배열에.length를 붙이고 확인하면 내부에 몇개의 아이템이 존재하는지 알려줍니다. 조심해야하는 점은 length는 배열 내부에 있는 방들의 개수를 새기때문에 0부터 100까지 넣은 배열의 경우 101이 출력됩니다. 그렇다면 이를 활용해서 새로운 배열을 만든 후 해당 배열에 0부터 랜덤 한 갯수까지 랜덤한 숫자를 넣고 length프로퍼티를 이용하여 해당 배열안에 있는 방번호와 item을 출력하는 코드를 한번 각자 만들어 보세요.

const random = Math.floor(Math.random() \* 100);

*//0부터 99까지 임의의 난수*

const array = [];

for(let i = 0; i< random; i++){

array[i] = Math.floor(Math.random() \* 10);;

}

for(let i = 0; i < array.length -1; i++)

{

console.log(`array ${i}번째 방 : ${array[i]}`);

}

잘하셨습니다. 오늘 수업의 마지막으로 배열을 조금 더 편리하게 해주는 배열의 기능들에 대해서 한번 알아봅시다.

Push라는 기능과 pop이라는 기능은 아마도 자바스크립트 배열에서 가장 많이 쓰는 기능이 아닌가 싶습니다. Push라는 녀석은 배열의 끝에 아이템을 자동으로 추가해주는 기능이고, pop이라는 녀석은 배열에서 마지막에 있는 녀석을 제거해줌과 동시에 마지막 녀석을 반환 해주는 녀석입니다.

그래서 사용 예시를 들어보면 다음과 같습니다.

Const array= [10,20,30];

array.push(40);//숫자 40추가

console.log(array)

array.pop();

console.log(array)

const lastOne = array.push();

console.log(lastOne)

console.log(array)

다음은 slice입니다. Slice는 배열의 특정 구간부터 특정 구간까지 배열을 잘라서 반환하는 기능으로 사용은 다음과 같습니다.

var array = []; //선언

var array = [1,2,3,4,5,6,7,8,9,10] // 선언과 동시에 할당

var b = array.slice(1,4); // b배열은 [2,3,4,5]

배열을 위한 for문은 아니지만 배열에 유용하게 쓸 수 있는 for문의 아종격인 for in문이 있습니다.

var array = [10,11,12];

for (property in array){

    console.log("name: " + property + "; value: " + array[property]);

}

For in 문은 in을 사이에 두고 왼쪽에는 오른쪽 배열의의 index값이 할당되어 목적에 따라 해당 배열의 끝까지 index를 사용 할 수 있는 장점이 있습니다.