<자바스크립트 정리\_노마드코더>

2022-03-29

1. HTML의 Element들을 JavaScript를 통해 변경함
   1. 자바스크립트는 HTML과 연결되어 있다.
   2. 자바스크립트는 Html을 읽어옴
   3. 따라서 자바스크립트로 html을 읽어올 뿐만 아니라 변경할 수 있다.
2. Html 의 가장 기본이되는 object은 document이다.
   1. Document == Web site 를 의미한다.
   2. Document가 모든것의 시작점이다.
   3. 따라서 자바스크립트로 Element들을 가져올 때 document.getElementById 의 형식으로 사용된다.
3. 변수 종류
   1. Let : 변할 수 있는 값
   2. Const : 변할 수 없는 값
   3. Var : 가장 초창기에 만들어진것으로, 변수의 속성을 파악할 수 없어서 사용 안함
4. Undefined VS null
   1. Undefined은 값이 아예 는 경우를 뜻한다.
   2. Null은 값이 비어있는 상태를 뜻하며, 절대 자연적으로 발생하지 않고 프로그래밍에 의해 할당된다.
5. Object VS Array
   1. 배열과 object의 차이는 괄호에 있다.
      1. 배열 [ ]
      2. Object { }
   2. 배열의 값이 const이면 해당 값을 변경하지 못한다
      1. A.push() 로 배열에 값을 추가할 수 있다.
   3. 배열과 달리 object는 const형으로 되어있어도 object자체를 바꾸지 않는다면 안에있는 값을 변경할 수 있다.
      1. Const Ob = { name : “wony” , age : 19, fat : true }
      2. Ob.name = “wony2” => 이렇게 const이더라도 값을 변경할 수 있다.
      3. Name은 Ob의 property
6. Console.log( ) = 내용출력 console.dir( ) = 해당 정보를 출력
7. Document.querySelector(“.hello h1”)
   1. 클래스 이름이 hello인 태그의 h1을 가져온다.
   2. 만약 hello라는 클래스가 많을경우 첫번째 h1만 가져온다. => 다 가져오고 싶은경우
   3. const hello = document.querySelectorAll(“.hello h1”) 의 형식을 사용한다
   4. hello.addEventListener(“click”, function()) 의 형식으로 event를 추가할 수 있다
   5. hello.onclick = functionName; == hello.addEventListener(“click”, functionName);
   6. transition : color 0.5s ease-in-out => 0.5초로 느리게 색을 변경할 수 있다.
   7. addEventListener를 사용할 때 functionName부분에 () 를 사용하지 않아야 하는 이유는 ()를 사용하게 되면 해당 함수는 한번 실행되고 끝난다.
      1. addEventListner 안의 함수는 브라우저가 실행하는 함수이기 때문에 () 를 사용하면 안된다.
8. H1.class vs h1.classList
   1. Class는 이전에 사용된 class의 정보를 기억하지 못함
   2. classList는 class들의 목록으로 작업할 수 있게 해줌
      1. h1.classList.add(“clicked”) => h1.classList.remove(“clicked”) 이렇게 사용하면 해당 class이름에서 clicked를 추가하거나 제거하면서 사용가능
      2. h1.classList.toggle(“clicked”) => toggle을 사용하면 위의 add와 remove의 기능을 한번에 실행할 수 있다. -> clicked가 className으로 있는지 확인 후 있으면 제거, 없으면 토글
9. form 태그 안의 submit 이나 button을 클릭하면 해당 url은 자동으로 새로고침 된다.
10. Event.preventDefault()
    1. 해당 함수는 이벤트의 기본 기능을 막는 역할을 한다.
    2. 예를들어 form의 submit의 새로고침을 막을수도 있고, 링크태크의 이동을 막을수도 있다.
11. String 합치기 “ “ 와 ` ` 의 차이 => ‘ ‘ (작은 따옴표) 가 아니고 ` ` (백틱) 이다.
    1. “Hello “ + userName;
    2. `Hello ${userName} ` => 작은따옴표가 아닌 백틱( ~ 와 같은곳에 있는 문자 ) 을 사용해야 동작함
12. 브라우저 자체 저장소 localStorage
    1. localStorage.getItem(value), localStorage.setItem(key, value) 으로 값을 저장, 가져올수 있다.
    2. localStorage에 저장되는 값은 무조건 텍스트이다. 따라서 텍스트만 저장된다.
    3. 배열이나 object를 String으로 바꾸기 위해서 JSON.stringify(array) 의 형식을 이용한다.
    4. String형식을 다시 Array나 object 형으로 만들기 위해서는 JSON.parse( ) 를 이용한다.
13. Clock
    1. Interval : 매번 일어나는 이벤트 => setInterval( funcName, 5000) => 5초간격으로 함수 호출
    2. Timeout : 설정시간 뒤에 이벤트 발생 => setTimeout( funcName, 5000) => 5초 뒤 실행
14. “1”.padStart ( 2, “0”) : string의 시작부분에 첫번째 인자로 할당된 길이와 string의 길이를 비교해서 부족하면 0을 채운다
    1. 만약 string이 12면 2자리이기 때문에 아무일도 발생하지 않는다.
    2. Padend()도 사용할 수있다.
15. Math
    1. Round() : 소수자리를 반올림함
    2. Ceil() : 소수자리를 무조건 올림함
    3. Floor() : 소수자리를 무조건 버림함
16. Function a() => 함수의 원소로 event와 item을 기본적으로 사용할 수있다.
    1. Event와 item은 매개변수일뿐 다른 이름으로 사용해도 같은 결과가 나온다.
    2. Event 같은 경우는 event.target의 형태로 쓰이며 이벤트와 관련되어있다.
    3. Item은 해당 태그와 관련된 속성으로,
17. Obj.forEach(funName) => 해당 오브젝트의 수만큼 함수를 반복
    1. Obj.forEach(element => { funName( element ) } ; => element 에 obj[0] 부터 순차적으로 들어감
18. Filter => 해당 배열이나 object에서 조건을 실행한 후 true면 포함하고, false면 값을 빼버린다.
    1. 따라서 filter를 이용하면 해당 조건에 해당하는 값만 빼고 새로운 배열 또는 object를 생성한다.
19. 사용자의 경도와 위도 로 위치를 알 수 있다.
    1. Navigator.geolocation.getCurrentPosition( funName , error);
       1. Coords.latitude 와 coords.longitude 가 위도와 경도임
20. 우클릭 방지 기능
    1. A.addEventListener(“contextmenu”, funcName(() => event.preventDefault()))