HTML은 정적 정보라면, JAVASCRIPT는 동적인 정보 – 사용자와 상호작용을 하는 기능

Html 코드로 자바스크립트 시작이 된다는걸 알려줘야함.

<script>…</script>

<이벤트>

1. 버튼 만들기

<input type=”button” value=”xx” onclick= “자바스크립언어”>

알림창 onclick=”alert(‘hi’)

2. 텍스트 창

<input type = “text” onchange=”alert(‘changed’)”>

Html 코드로 자바스크립트 시작이 된다는걸 알려줘야함.

<script>…</script>

변수와 대입연산자

var xxx = abc

변수 “+xxx+”

검색 (자바스크립 선택 tag : css 선택자)

* Javascript select tag by css selector ((자바스크립 선택 tag : css 선택자))
* 자바스크립트로 스타일 속성에 넣는방법 javascript element style
* Javascript style background color

블리언와 비교연산자

블리언은 true or false 두 데이터로 존재.

비교연산자는 ‘===’

<h1>Comparison operators & Boolean</h1>

<h2> === </h2>

<h3>1===1</h3>

<script>

document.write(1===1);

</script>

<h3>1===2</h3>

<script>

document.write(1===2);

</script>

<h3>1<2</h3> - html 에서 <는 문법이기 때문에 1&lt;2 라고 표기

<script>

document.write(1===2);

</script>

<h3>1&lt;1</h3>

<script>

document.write(1<1);

</script>

<h3>1&gt;1</h3>

<script>

document.write(1>1);

</script>

조건문

If () … else ()

<h1>Conditional statements</h1>

<h2>Program</h2>

<script>

document.write("1<br>");

document.write("2<br>");

document.write("3<br>");

document.write("4<br>");

</script>

<h2>IF-true</h2>

<script>

document.write("1<br>");

if(true){

document.write("2<br>");

} else {

document.write("3<br>");

}

document.write("4<br>");

</script>

<h2>IF-false</h2>

<script>

document.write("1<br>");

if(false){

document.write("2<br>");

} else {

document.write("3<br>");

}

document.write("4<br>");

</script>

조건문 활용

Javascript element get value 검색

<input id="night\_day" type="button" value="night" onclick="

if(document.querySelector('#night\_day').value === 'night') {

document.querySelector('body').style.backgroundColor='black';

document.querySelector('body').style.color='white';

document.querySelector('#night\_day').value = 'day';

} else {

document.querySelector('body').style.backgroundColor='white';

document.querySelector('body').style.color='black';

document.querySelector('#night\_day').value = 'night';

}">

리팩토링 : 공장으로 다시 보내서 보수한다 느낌

<input id="night\_day" type="button" value="night" onclick="

if(document.querySelector('#night\_day').value === 'night') {

document.querySelector('body').style.backgroundColor='black';

document.querySelector('body').style.color='white';

document.querySelector('#night\_day').value = 'day';

} else {

document.querySelector('body').style.backgroundColor='white';

document.querySelector('body').style.color='black';

document.querySelector('#night\_day').value = 'night';

}">

코드에서 id값을 삭제하고 this.value로 사용해도된다.

<input type="button" value="night" onclick="

if(this.value === 'night') {

document.querySelector('body').style.backgroundColor='black';

document.querySelector('body').style.color='white';

this.value = 'day';

} else {

document.querySelector('body').style.backgroundColor='white';

document.querySelector('body').style.color='black';

this.value = 'night';

}">

더 단축하는 방법

document.querySelector('body') 중복

var target = document.querySelector('body');

<input type="button" value="night" onclick="

if(this.value === 'night') {

target.style.backgroundColor='black';

target.style.color='white';

this.value = 'day';

} else {

target.style.backgroundColor='white';

target.style.color='black';

this.value = 'night';

}">

으로 변경가능

배열 (반복문에서 제대로 알수있음)

- 연관된 데이터들을 수납하는 용도?

문자는 “ “ 로 하는데 배열은 [ ] 로 함. 대괄호 안에 값을 씀.

var coworkers = [“egoing”,”leezche”]; 대괄호 안에 값을 인덱스라고 부름

인덱스 0번은 egoing이고 1번은 leezche 이다.

<h1>Array</h1>

<h2>Syntax</h2>

<script>

var coworkers = ["egoing", "leezche"];

</script>

<h2>get</h2>

<script>

document.write(coworkers[0]);

document.write(coworkers[1]);

</script>

<h2>add</h2>

<script>

coworkers.push('duru');

coworkers.push('taeho');

</script>

<h2>count</h2>

<script>

document.write(coworkers.length);

</script>

반복문

While () { } while이 false가 될 때까지 실행.

<h1>Loop</h1>

<ul>

<script>

document.write('<li>1</li>');

var i = 0;

while(i < 3) {

document.write('<li>2</li>');

document.write('<li>3</li>');

i = i+1;

}

document.write('<li>4</li>');

배열과 반복문

<h1>Loop & Array</h1>

<script>

var coworkers = ['egoing','leezche','duru','taeho'];

</script>

<h2>Co Wworkers</h2>

<ul>

<script>

var i = 0;

while(i<coworkers.length) {

document.write('<li><a href="https://www.naver.com/"'+coworkers[i]+'>'+coworkers[i]+'</a></li>');

i = i+1;

}

개발자도구

Console.log() 콘솔창에서 출력

var alist = document.querySelectorAll('a');

var i = 0;

while (i<alist.length) {  
 console.log(alist[i]); - 코딩할때 삭제

alist[i].style.color='blue'

i = i+1;

}

함수 - 수납상자

Function

중복이 연속적이지 않는다면, 함수를 사용해야한다.

    <h1>Function</h1>

    <h2>Basic</h2>

    <ul>

        <script>

            function two(){

            document.write("<li>2-1</li>");

            document.write("<li>2-2</li>");

            }

            document.write("<li>1</li>")

            two();

            document.write("<li>3</li>")

            two();

        </script>

Function을 쓸 때, this로 되어있는 친구들은 self 로 변경 후 매개제체 (this)라고 입력해줘야한다.

객체 (엄청 어렵다..?)

* 정리정돈의 수단 폴더로 봐도 무관하다..
* document.querySelector('body').style.color= color;

document ; 객체

querySelector 메소드. 라고 부름

객체쓰기와 읽기

    <h1>Object</h1>

    <h2>Create</h2>

    <script>

        var coworkers = {

            "programmer":"egoing",

            "designer":"leezche"

        };

        document.write("programmer : "+coworkers.programmer+"<br>");

        document.write("designer : "+coworkers.designer+"<br>");

        coworkers.bookkeeper ="duru";

        document.write("bookkeeper : "+coworkers.bookkeeper+"<br>")

        coworkers["data scientist"] = "taeho";

        document.write("data scientist : "+coworkers["data scientist"]+"<br>")

    </script>

Data scientist는 띄어쓰기가 필요로 한데 [“~~~”] 로 데이터를 입력하여도 똑같이 입력이 가능함.